Clustering Documents with Maximal Substrings

Tomonari Masada[1], Atsuhiro Takasu[2], Yuichiro Shibata[1], and Kiyoshi Oguri[1]

[1] Nagasaki University, 1-14 Bunkyo-machi, Nagasaki-shi, Nagasaki, Japan,
{masada,shibata,oguri}@nagasaki-u.ac.jp
[2] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan,
takasu@nii.ac.jp

Abstract. This paper provides experimental results showing that we can use maximal substrings as elementary building blocks of documents in place of the words extracted by a current state-of-the-art supervised word extraction. Maximal substrings are defined as the substrings each giving a smaller number of occurrences even by appending only one character to its head or tail. The main feature of maximal substrings is that they can be extracted quite efficiently in an unsupervised manner. We extract maximal substrings from a document set and represent each document as a bag of maximal substrings. We also obtain a bag of words representation by using a state-of-the-art supervised word extraction over the same document set. We then apply the same document clustering method to both representations and obtain two clustering results for a comparison of their quality. We adopt a Bayesian document clustering based on Dirichlet compound multinomials for avoiding overfitting. Our experiment shows that the clustering quality achieved with maximal substrings is acceptable enough to use them in place of the words extracted by a supervised word extraction.

Key words: maximal substring, unsupervised method, document clustering, suffix array, Bayesian modeling

1 Introduction

Recently, researchers propose a wide variety of large scale data mining methods, where documents originating from SNS environments or DNA/RNA sequences provided by next generation sequencing are a typical target of their proposals. Many of those methods adopt an *unsupervised* learning, because it is often difficult to prepare a sufficient amount of training data for a *supervised* learning. This paper focuses on text mining, where we have various useful unsupervised methods, e.g. document clustering [15], topic extraction [2], topical trend analysis [23], etc. However, most of such unsupervised methods assume that each document is already represented as a *bag of words*, i.e., as a set of the numbers of occurrences of words. Therefore, we should first extract elementary building blocks that can be called words from documents.

With respect to English, French, German, etc, we can easily obtain such building blocks, because each character sequence separated by white spaces can

be regarded as a word. While we may further conduct a stemming to obtain a canonical form of the words, this causes no serious burden.

However, with respect to Japanese, Chinese, Korean, etc, it is far from a trivial task to extract such elementary building blocks from documents. Japanese and Chinese sentences contain no white spaces and thus give no word boundaries. While Korean sentences contain many white spaces, most of the character sequences separated by white spaces consist of two or more words [4].

Therefore, various word extraction methods have been proposed. However, many of such methods are a *supervised* one. They require a hand-maintained dictionary that should be constantly updated or are based on a mathematical model of character sequences that should be trained with a sufficient amount of training data, where supervised signals (e.g. 0/1 labels giving word boundaries, categorical labels giving grammatical roles, etc) are assigned by human annotators. Therefore, any mining method sitting on a supervised word extraction will show difficulty in scaling up to larger data sets even when the mining method itself is an unsupervised one.

This paper provides experimental results showing that we can use *maximal substrings* [17] as elementary building blocks of documents in place of the words extracted by a current state-of-the-art supervised word extraction. The most important feature of maximal substrings is that they can be extracted in an *unsupervised* manner. Therefore, we need no training data aside target data. Further, maximal substrings can be extracted quite efficiently.

From here, when we use the words extracted by a supervised method to represent each document as a bag of words, we simply call this *bag of words representation* and distinguish it from *bag of maximal substrings representation*.

For evaluating the effectiveness of maximal substrings, we compare bag of maximal substrings representation with bag of words representation in *document clustering*, where the latter representation is obtained by using a state-of-the-art supervised word extraction. We compare these two types of representation based on the quality of document clustering. We run the same Bayesian clustering algorithm on the same document set and obtain two different clustering results depending on whether we use bag of maximal substrings representation or bag of words representation. We then compare the clustering quality in F-scores and clarify the effectiveness of maximal substrings.

As far as we know, this paper firstly gives a quantitative comparison between bag of maximal substrings representation and bag of words representation in document clustering. While Chumwatana et al. [5, 6] conduct a similar experiment with respect to Thai documents, the authors fail to make evaluation reliable, because the data set only contains tens of documents. Further, they do not compare bag of maximal substrings representation with bag of words representation.

Our comparison was conducted on a set of tens of thousands of Korean and Chinese newswire articles. To compare with maximal substrings, we extracted words by applying a dictionary-based morphological analyzer [8] to Korean documents and by applying a word segmenter, implemented by us based on linear conditional random fields (CRF) [19], to Chinese documents. Both are a super-

vised word extraction. The former requires a hand-maintained dictionary, and the latter requires a sufficient amount of human annotated training data.

Our experiment will provide the following observations:

- Both for Korean and Chinese documents, maximal substrings are as effective as the words extracted by a current state-of-the-art supervised method as long as we remove high and low frequency maximal substrings with a little care.
- Document clustering requires longer time and larger memory when we use maximal substrings, because the number of maximal substrings is larger than that of the words extracted by a supervised method. This is the cost we should pay in performing various text mining tasks with maximal substrings.

The rest of the paper is organized as follows. Section 2 reviews the works related to the extraction of elementary building blocks of documents. Section 3 describes the details of maximal substrings and of document clustering used in our evaluation experiment. Section 4 includes the procedure and the results of our experiment. Section 5 concludes the paper with discussions and future work.

This paper improves our preceding paper [11] with respect to the following three aspects. First, we added two data sets to make our experiment more reliable. Second, we provided an additional method for reducing the variety of maximal substrings extracted from Chinese documents. While we did not obtain any good results in [11] for Chinese documents, this new reduction method made maximal substrings as effective as the words extracted by a CRF-based supervised method. Third, we conducted an MCMC sampling aside the EM algorithm in [11] for document clustering and made our experiment more comprehensive.

2 Previous Works

Most text mining methods require *word extraction*, i.e., extraction of elementary building blocks that can be called words, as a preprocessing of documents. For English, French, German, etc, we have words as the character sequences separated by white spaces. Therefore, we at most need to apply stemming for obtaining a canonical form of the words. In contrast, for Japanese, Chinese, Korean, etc, word extraction is never a trivial task.

Word extraction can be conducted, for example, by analyzing language-specific word sequence structures with a hand-maintained dictionary [8], or by labeling character sequences with an elaborated probabilistic model whose parameters are in advance optimized with respect to a human annotated training data set [21]. However, recent research trends point to increasing need for large scale text mining. Therefore, an intensive use of such *supervised* methods becomes less realistic, because it becomes more difficult to prepare a hand-maintained data set of size and quality sufficient to serve as a dictionary or as a training data set for exploring very large scale *unknown* data.

Actually, we already have important results for *unsupervised* word extraction.

Poon et al. [18] propose an unsupervised word segmentation by using log-linear models, which are often adopted for supervised word segmentation, in

an unsupervised learning framework. However, when computing the expected count, which is required in learning process, the authors exhaustively enumerate all segmentation patterns. Consequently, this approach is only applicable to the languages whose sentences are given as a set of short character sequences separated by white spaces (e.g. Arabic and Hebrew), because the total number of segmentation patterns is not so large for each of such short character sequences. In other words, this approach may be extremely inefficient for the languages whose sentences contain no white spaces (e.g. Chinese and Japanese).

Mochihashi et al. [13] provide a sophisticated Bayesian probabilistic model for segmenting given sentences into words in a totally unsupervised manner. The authors improve the generative model of Teh [20] and utilize it for modeling both character n-grams and word n-grams. The proposed model can cope with the data containing so-called out-of-vocabulary words, because the generative model of character n-grams serves as a new word generator for that of word n-grams. However, highly complicated sampling procedure, including MCMC for the nested n-gram models and segmentation sampling by an extended forward-backward algorithm, may encounter an efficiency problem when we try to implement this method by ourselves, though the proposed model is well designed enough to prevent any exhaustive enumeration of segmentation candidates.

Okanohara et al. [17] propose an unsupervised method from a completely different angle. The authors extract *maximal substrings*, i.e., the substrings each giving a smaller number of occurrences even by appending only one character to its head or tail, as elementary building blocks of documents. The extraction can be efficiently implemented and conducted as is shown in the works related to suffix array or Burrows-Wheeler transform [7, 1, 14, 16]. While Zhang et al. [25] also provide a method for extracting a special set of substrings, this is not the set of maximal substrings. Further, their method has many control parameters and thus is guided not by a principled methodology, but by a heuristic intuition.

In this paper, we adopt maximal substrings as elementary building blocks of documents by following the line of [17] and evaluate the effectiveness of maximal substrings in *document clustering*, because previous works [17, 25] have proved the effectiveness only in document classification.

While we can find several works employing maximal substrings in document clustering, this paper firstly gives a quantitative comparison between bag of maximal substrings representation and bag of words representation as far as we know. Zhang et al. [24] present a Chinese document clustering method using maximal substrings. However, the authors give no quantitative evaluation. Especially, maximal substrings are not compared with the words extracted by some elaborated supervised method. While Li et al. [9] also propose a document clustering based on the maximality of subsequences, the authors focus not on character sequences, but on word sequences. Further, the proposed method utilizes WordNet, i.e., an external knowledge base, for reducing the variety of maximal subsequences. Therefore, their method is not an unsupervised one.

This paper will show what kind of effectiveness maximal substrings can provide in document clustering. In the evaluation experiment, we prepared a set of

tens of thousands of documents as an input for clustering and made our evaluation reliable. We only appealed to a simple frequency-based reduction of the variety of maximal substrings and used no external knowledge base. Further, we compared the clustering quality achieved with maximal substrings to that achieved with the words extracted by an elaborated supervised method.

3 Clustering Documents with Maximal Substrings

3.1 Maximal Substrings

Maximal substrings are defined as a substring whose number of occurrences is reduced even by appending only one character to its head or tail. We can discuss more formally as follows. Let S denote a string of length $l(S)$ over a lexicographically ordered character set Σ. At the tail of S, a special character \$, called sentinel, is attached, i.e., $S[l(S)] = \$$. The sentinel \$ does not appear in the given original string and is smaller than all other characters in lexicographical order. For a pair of strings S and T over Σ, we define the set of all occurrence positions of T in S as follows:

$$Pos(S,T) \equiv \{i : S[i+j-1] = T[j] \text{ for } j = 1,\ldots,l(T)\ \}\ . \tag{1}$$

We denote the nth smallest element in $Pos(S,T)$ by $pos_n(S,T)$. Further, we define $RPos(S,T)$ as follows:

$$RPos(S,T) \equiv \{(n, pos_n(S,T) - pos_1(S,T)) : n = 1,\ldots,|Pos(S,T)|\}\ . \tag{2}$$

That is, $RPos(S,T)$ is the set of all occurrence positions of T in S *relative to* the first smallest occurrence position. Then, T is a *maximal substring* of S when

- $|RPos(S,T)| > 1$,
- $RPos(S,T) \neq RPos(S,T')$ for any T' such that $l(T') = l(T) + 1$ and $T[j] = T'[j], j = 1,\ldots,l(T)$, and
- $RPos(S,T) \neq RPos(S,T')$ for any T' such that $l(T') = l(T) + 1$ and $T[j] = T'[j+1], j = 1,\ldots,l(T)$.

The last condition corresponds to "left expansion" discussed in [17].

When we extract maximal substrings from a document set, we first concatenate all documents by inserting a special character, which does not appear in the given document set, between the documents. The concatenation order is irrelevant to our discussion. We put a sentinel at the tail of the resulting string and obtain a string S from which we extract maximal substrings. We can efficiently extract all maximal substrings from S in time proportional to $l(S)$ [17].

After the extraction, the maximal substrings containing special characters put between the documents are removed. However, the number of the resulting maximal substrings is in general far larger than the number of the words extracted by a state-of-the-art supervised method from the same document set. Therefore, we further reduce the variety of maximal substrings by removing

the maximal substrings containing white spaces, delimiters (e.g. comma, period, question mark, etc), and other functional characters (e.g. parentheses, hyphen, center dot, etc).

Even after the above reduction, we still have a large number of maximal substrings. Therefore, we propose a simple frequency-based strategy for reducing the variety by using three integer parameters n_L, r_H, and r_h as follows:

1. Remove the maximal substrings whose frequencies are smaller than n_L;
2. Remove the top r_H highest frequency maximal substrings; and
3. Remove the maximal substrings *of length one* among the top r_h highest frequency maximal substrings, where r_h should be larger than r_H.

The third reduction using r_h was not proposed in [11]. However, this additional reduction made maximal substrings as effective as the words extracted by the supervised word segmenter for Chinese documents. Consequently, we could obtain more interesting results than in [11] with respect to Chinese documents.

While we tried various settings for n_L, r_H, and r_h, this paper provides a limited number of settings, because other settings gave no remarkable improvement. Our reduction strategy was also applied to the words extracted by a supervised method, because we could obtain better evaluation results with this reduction.

3.2 Bayesian Document Clustering

Dirichlet compound multinomial (DCM) When we represent documents as a bag of maximal substrings or of words, multinomial distribution [15] is a natural choice for document modeling, because we can identify each document with a frequency histogram of maximal substrings or of words. However, it is often discussed that multinomial distributions are likely to overfit to *sparse* data. Here the term "sparse" means that the number of different maximal substrings or of different words appearing in each document is far less than the total number observable in the entire document set.

Therefore, we use a Bayesian document model called *Dirichlet compound multinomial* (DCM) [10] and avoid overfitting. Let K denote the number of clusters. We prepare K multinomial distributions each modeling a frequency distribution for a different document cluster. Further, a Dirichlet prior distribution is applied to each multinomial distribution. By marginalizing out the multinomial parameters, we obtain a DCM for each document cluster. The parameters of these K DCMs and their mixing proportions are estimated by the EM algorithm described below.

EM algorithm We prepare notations for discussions. We assume that the given document set contains J documents and that W different words (or maximal substrings) can be observed in the document set. Let c_{jw} be the number of occurrences of the wth word (or maximal substring) in the jth document. The sparseness in our case means that $c_{jw} = 0$ holds for most w. Let $\alpha_k = (\alpha_{k1}, \ldots, \alpha_{kW})$ be the hyperparameters of the Dirichlet prior prepared for the kth document

cluster. The probability that the jth document belongs to the kth cluster is denoted by p_{jk}. Note that $\sum_k p_{jk} = 1$. We define $\alpha_k \equiv \sum_w \alpha_{kw}$ and $c_j \equiv \sum_w c_{jw}$. We update the cluster assignment probabilities and the hyperparameters with the EM algorithm described below.

E step: For each j, update p_{jk}, $k = 1, \ldots, K$ by

$$p_{jk} \leftarrow \frac{\sum_j p_{jk}}{\sum_j \sum_k p_{jk}} \cdot \frac{\Gamma(\alpha_k)}{\Gamma(c_j + \alpha_k)} \prod_w \frac{\Gamma(c_{jw} + \alpha_{kw})}{\Gamma(\alpha_{kw})}$$

and then normalize p_{jk} by $p_{jk} \leftarrow p_{jk} / \sum_k p_{jk}$.
M step: For each k, update α_{kw}, $w = 1, \ldots, W$ by

$$\alpha_{kw} \leftarrow \alpha_{kw} \cdot \frac{\sum_j p_{jk} \{\Psi(c_{jw} + \alpha_{kw}) - \Psi(\alpha_{kw})\}}{\sum_j p_{jk} \{\Psi(c_j + \alpha_k) - \Psi(\alpha_k)\}}$$

where $\Gamma(\cdot)$ is gamma function and $\Psi(\cdot)$ is digamma function. The M step is based on Minka's discussion [12]. We ran 200 iterations of the E and M steps.

Before entering into the loop of E and M steps, we initialize all α_{jk} to 1, because this makes every Dirichlet distribution a uniform distribution. Further, we initialize p_{jk} not randomly but by the EM algorithm for multinomial mixtures [15]. In the EM for multinomial mixtures, we use a random initialization for p_{jk}. The execution of the EM for multinomial mixtures is repeated 30 times. Each of the 30 executions of the EM for multinomial mixtures gives a different estimation of p_{jk}. Therefore, we choose the estimation giving the largest likelihood as the initial setting of p_{jk} in the EM algorithm for DCM. We conduct this entire procedure three times. Among the three results, we select the one giving the largest likelihood as the final output of our EM algorithm. We then assign each document to the cluster giving the largest value among $p_{j1}, \ldots, p_{jK}$ in this final output. The time complexity of this EM is $O(IKM)$, where I is the number of iterations and M is the number of unique pairs of document and word (or the number of unique pairs of document and maximal substring). Note that M is far smaller than $J \times W$ due to the sparseness discussed above.

MCMC sampling We also employed *Gibbs sampling*, a widely used class of MCMC samplings, for inference. Our Gibbs sampling updates cluster assignments by picking up the documents in a random order. The assignment of the jth document is a random multinomial draw determined by the following probabilties for $k = 1, \ldots, K$:

$$p_{jk} \propto m_k^{\neg j} \cdot \frac{\Gamma(c_k^{\neg j} + \alpha_k)}{\Gamma(c_k^{\neg j} + c_j + \alpha_k)} \prod_w \frac{\Gamma(c_{kw}^{\neg j} + c_{jw} + \alpha_{kw})}{\Gamma(c_{kw}^{\neg j} + \alpha_{kw})} , \tag{3}$$

where m_k is the number of documents assigned to the kth cluster, c_{kw} is the number of occurrences of the wth word (or maximal substring) in the documents assigned to the kth cluster, and c_k is defined as $\sum_w c_{kw}$. The notation "$\neg j$" in Eq. (3) means that we use the corresponding statistics after removing the jth

document. The probabilities $p_{j1}, \ldots, p_{jK}$ for each j should be normalized so that $\sum_k p_{jk} = 1$ is satisfied. Based on these K probabilities, we draw a new cluster assignment of the jth document.

We repeat a series of 50 iterations of this MCMC sampling ten times from different initializations. We then choose the run giving the largest likelihood among the ten runs and continue the chosen run until we reach 300 iterations. We regard the cluster assignments at the 300th iteration of the chosen run as the final output of our MCMC sampling. The time complexity of the MCMC sampling is $O(IKM)$, where I is the number of iterations.

4 Evaluation Experiment

4.1 Document Sets

We used four document sets in our experiment: two sets of Korean newswire articles and two sets of Chinese newswire articles. Each set consists of already categorized articles downloaded from the Web. Our task for evaluation is to guess the categories by clustering documents. No documents belong to more than one categories. Below we describe how we collected each set.

1. The first set is a set of Korean newswire articles downloaded from the Web site of *Seoul Newspaper*[1]. We denote this set as SEOUL1. This set consists of 35,783 articles from the six categories: Economy, Sports, International, Entertainment, Politics, and Culture. We collected this data set so that the numbers of the documents contained in each category is almost the same. Consequently, the ranges of document dates are different for each category. For example, while the dates observed in Entertainment category range from July 2007 to May 2011, those in Politics category range from July 2010 to May 2011. This is because the per day number of articles in Politics category is larger than that in Entertainment category. Table 1 gives the numbers of documents in each category. This table also includes the numbers for the other three document sets.
2. The second one is also a set of Korean newswire articles downloaded from the Web site of Seoul Newspaper. However, we collected the articles from the same range of the dates for all categories. We denote this set as SEOUL2. SEOUL2 consists of 52,730 articles whose dates range from January 2008 to September 2009. Each article belongs to one among the following four categories: Economy, Local Issues, Sports, and Politics.
3. The third one is a set of Chinese newswire articles downloaded from the Web site of *China News*[2]. This set, denoted as CNEWS, consists of 47,171 articles whose dates range from June to December in 2010. Each article belongs to one among the following six categories: Economy, International, Entertainment, Information & Technology, Domestic Issues, and Social Issues.

[1] http://www.seoul.co.kr/
[2] http://www.chinanews.com/

Table 1. Number of documents belonging to each category in the four document sets prepared for our experiment, i.e., SEOUL1, SEOUL2, CNEWS, and XINHUA.

SEOUL1 Korean document set						
Economy	Sports	International	Entertainment	Politics	Culture	total
5,870	5,129	6,309	6,206	6,242	6,027	35,783

SEOUL2 Korean document set				
Economy	Local	Sports	Politics	total
13,058	22,993	6,621	10,058	52,730

CNEWS Chinese document set						
Economy	International	Entertainment	Info&Tech	Domestic	Social	total
11,285	5,515	9,448	10,589	6,955	3,379	47,171

XINHUA Chinese document set			
Economy	International	Politics	total
3,290	10,230	6,607	20,127

4. The fourth one is a set of Chinese newswire articles downloaded from *Xinhua Net*[3]. We denote this data set as XINHUA. This set consists of 20,127 articles whose dates range from May to December in 2009. Each article belongs to one among the following three categories: Economy, International, and Politics. For this set, it was relatively difficult to discriminate between Economy category and International category. Therefore, the evaluation results were not so good even though the number of categories is only three.

For each data set, we set the number of clusters K to the number of categories and ran the EM algorithm and the MCMC sampling described in Section 3.2. We regarded article categories as the ground truth for evaluation.

4.2 Extraction and Reduction

For every document in each document set, we obtained two representations, i.e., a bag of maximal substrings representation and a bag of words representation. We obtained the former representation by extracting all maximal substrings from the document set and then counting their numbers of occurrences in each document. The latter was obtained by applying a supervised word extraction sentence by sentence and then count their numbers of occurrences in each document.

We applied KLT morphological analyzer [8] to SEOUL1 and SEOUL2. To CNEWS and XINHUA, we applied a word segmenter implemented based on an L1-regularized linear conditional random fields (CRF) [19]. The parameter optimization in training this Chinese word segmenter is based on a stochastic gradient descent algorithm with exponential decay scheduling [22]. This segmenter achieved the following F-scores for the four data sets of SIGHAN Bakeoff 2005 [21]: 0.943 (AS), 0.941 (HK), 0.929 (PK) and 0.960 (MSR). In our experiment, we used the segmenter trained with MSR data set, because this gave the

[3] http://www.xinhuanet.com/

表	关	其	实	同	们	品	还	面	与	事	内	据	已	天	示	都	最	商	目
区	民	理	主	外	度	但	务	此	子	定	小	合	:	之	加	当	重	报	力
得	司	体	政	法	员	展	名	手	入	《	》	金	%	信	从	水	次	并	增
被	道	期	因	相	万	间	比	（	企	提	所	如	）	情	利	没	应	制	很
数	位	明	点	好	平	调	联	安	心	正	今	费	ｉ	格	计	然	及	ｅ	受
三	量	收	ｎ	化	海	两	保	建	学	消	起	看	至	ｏ	济	影	着	更	些
台	东	持	投	总	果	户	问	意	北	基	门	性	近	管	达	第	视	认	南
里	向	接	件	去	该	交	强	由	、	称	口	无	解	演	系	广	种	式	车
西	ａ	布	购	规	设	题	Ｐ	息	城	特	告	只	；	局	代	亿	文	流	社
回	游	导	任	各	指	统	她	张	营	运	查	站	服	少	处	物	京	销	常
打	女	立	求	程	省	身	样	组	己	做	推	使	路	原	未	山	显	直	团
股	周	续	需	预	专	造	线	标	活	每	集	支	院	江	给	结	华	售	灾
权	二	话	客	？	完	传	让	份	先	么	气	士	际	低	知	府	非	王	众
技	别	首	整	片	超	想	老	警	证	州	放	战	级	Ｃ	几	单	斯	速	项
再	界	带	转	创	络	取	议	供	世	空	约	势	友	领	较	节	步	术	降
头	共	易	改	涨	快	变	案	或	委	风	难	Ｓ	ｓ	办	采	引	始	准	观
ｒ	施	防	那	军	责	决	备	一	质	李	确	升	房	包	感	率	十	Ｄ	选
响	源	则	黄	模	票	救	亚	况	款	德	争	反	光	昨	值	科	货	店	乐
随	型	何	获	ｔ	某	农	注	博	球	米	环	研	尔	深	ｌ	半	ｄ	Ｇ	把
ｈ	才	买	Ａ	论	马	划	言	书	银	形	具	队	构	雨	媒	走	号	监	存

Fig. 1. A part of the maximal substrings of length one removed from CNEWS data set by our new reduction method proposed for Chinese documents.

highest F-score. For Korean language, we could not find any training data comparable with SIGHAN training data in its size and quality. Therefore, we used a dictionary-based morphological analyzer for Korean documents.

The wall clock time required for extracting all maximal substrings was only a few minutes for all data sets on a PC equipped with Intel Core i7 920 CPU. This wall clock time is not widely different from the time required for word extraction by our CRF-based Chinese word segmenter, though the time required for training the segmenter is not included. However, the wall clock time required for extracting all maximal substrings is much less than the time required by the Korean morphological analyzer, because this morphological analyzer achieves its excellence by dictionary lookups. While this morphological analyzer can provide part-of-speech tags, they are not used in the experiment.

Both for maximal substrings and the words extracted by the supervised method, we reduce the varieties based on their frequencies as Table 2 presents. Both in SEOUL1 and SEOUL2, we only removed low frequency maximal substrings by setting n_L to 50 or 100. For example, when $n_L = 50$, we remove all maximal substrings whose frequencies are less than 50. We did not remove any high frequency maximal substrings, because this gave no remarkable improvement. With respect to both SEOUL1 and SEOUL2, we applied the same reduction procedure to the words extracted by the morphological analyzer.

In CNEWS and XINHUA, we removed low frequency maximal substrings by setting n_L to 50 or 100. Further, we removed high frequency ones by setting r_H to 100, which means that we removed the top 100 highest frequency maximal substrings. The same reduction is also employed for reducing the variety of the words extracted by our CRF-based word segmenter. However, only for maximal substrings, we additionally reduced their variety by setting r_h to 1,000. That is, we removed the maximal substrings *of length one* from the top 1,000 highest

Table 2. Specifications of the four data sets used in our experiment.

data set name	J	K	extraction method	n_L	r_H	r_h	W	M
SEOUL1	35,783	6	MaxSubstr	50	-	-	72,544	36,462,658
			MaxSubstr	100	-	-	44,048	34,813,328
			Morph	50	-	-	16,908	6,548,876
			Morph	100	-	-	10,165	6,196,036
SEOUL2	52,730	4	MaxSubstr	50	-	-	72,104	34,562,947
			MaxSubstr	100	-	-	45,360	33,037,750
			Morph	50	-	-	20,068	7,312,519
			Morph	100	-	-	12,411	6,913,269
CNEWS	47,171	6	MaxSubstr	50	100	1,000	220,107	42,187,771
			MaxSubstr	100	100	1,000	103,815	35,406,367
			WordSeg	50	100	-	19,998	7,155,607
			WordSeg	100	100	-	12,990	6,796,572
XINHUA	20,127	3	MaxSubstr	50	100	1,000	52,530	8,324,321
			MaxSubstr	100	100	1,000	24,635	6,775,577
			WordSeg	50	100	-	8,518	2,018,329
			WordSeg	100	100	-	5,444	1,862,819

frequency maximal substrings. We did not use this reduction in [11] and could not obtain any good results for maximal substrings. Figure 1 presents a part of the maximal substrings of length one removed from CNEWS data set by this new reduction method. As Figure 1 shows, many of the maximal substrings of length one have no power to discriminate topics. They may relate to a specific topic as a part of the words of length two or more. While this reduction using r_h led to a drastic improvement for maximal substrings, we could not obtain any remarkable improvements for the words extracted by our segmenter. This may be because the supervised segmenter did not give so many words of length one. Therefore, we employed the reduction using r_h only for maximal substrings.

Table 2 provides the number of different words (or different maximal substrings) W and the number of unique document word pairs (or unique document maximal substring pairs) M for all document sets. The number M is important, because the running time of our clustering algorithm is proportional to this number. Table 2 shows that M is increased roughly by factor of five when we use maximal substrings in place of the words extracted by the supervised method, i.e., KLT morphological analyzer or our CRF-based segmenter. Consequently, the running time of the document clustering is also increased roughly by factor of five. This is the price we should pay when we adopt bag of maximal substrings representation in place of bag of words representation.

4.3 Evaluation Measure

We evaluated the quality of document clustering as follows:

1. We calculate precision and recall for each cluster. Precision is defined as

$$\frac{\#(\text{true positive})}{\#(\text{true positive}) + \#(\text{false positive})} \ ,$$

and recall is defined as

$$\frac{\#(\text{true positive})}{\#(\text{true positive}) + \#(\text{false negative})} \ ;$$

2. We calculate *F-score* as the harmonic mean of precision and recall; and
3. The F-score is micro-averaged over all clusters.

The above micro-averaged F-score is our evaluation measure. From here, we denote this micro-averaged F-score simply as F-score.

We ran document clustering 50 times and obtained 50 F-scores for each setting. Table 2 gives four rows for each document set. That is, we tried four settings for each set. Further, we conducted two types of inference, i.e., the EM algorithm and the MCMC sampling. Consequently, we tried eight settings for each data set and had a set of 50 F-scores for each of these eight settings. The evaluation result was represented by the mean and standard deviation of 50 F-scores for each of the eight settings with respect to each data set.

4.4 Analysis

Figure 2 presents all results with four charts corresponding to four data sets.

The top left chart gives the results for SEOUL1 data set. Each bar accompanied with an error bar shows the mean and standard deviation of 50 F-scores obtained by running the clustering algorithm 50 times for each setting. The four bars in the upper half gives the results obtained when we use maximal substrings as elementary building block of documents, and the four bars in the lower half gives the results obtained when we use the words extracted by the morphological analyzer. For both cases, we tested the two settings, $n_L = 50$ and $n_L = 100$, for n_L and ran the two types of inference, the EM algorithm and the MCMC sampling. Therefore, we have eight settings in total. Also for the other data sets, we have eight settings. With respect to SEOUL1, we achieved the best mean F-score 0.754 when we used maximal substrings, reduced their variety by setting $n_L = 100$, and ran the EM algorithm. The difference from the mean F-scores obtained with the words extracted by the morphological analyzer is statistically significant based on a two-tailed Student's t-test with p value less than 0.01.

The top right chart presents the results for SEOUL2. For this data set, the EM algorithm led to a better result than the MCMC sampling for every setting. We could obtain the best mean F-score 0.887 when we used the words extracted by the morphological analyzer, reduced their variety by setting $n_L = 50$, and ran the EM algorithm for clustering. The difference from the best result obtained with maximal substrings is statistically significant based on a two-tailed Student's t-test with p value less than 0.01. However, the difference is at most 0.020 ($= 0.887 - 0.867$). On the other hand, for SEOUL1, the difference of the best mean F-score obtained with maximal substrings from that obtained with the words given by the morphological analyzer amounts to 0.037 ($= 0.754 - 0.717$).

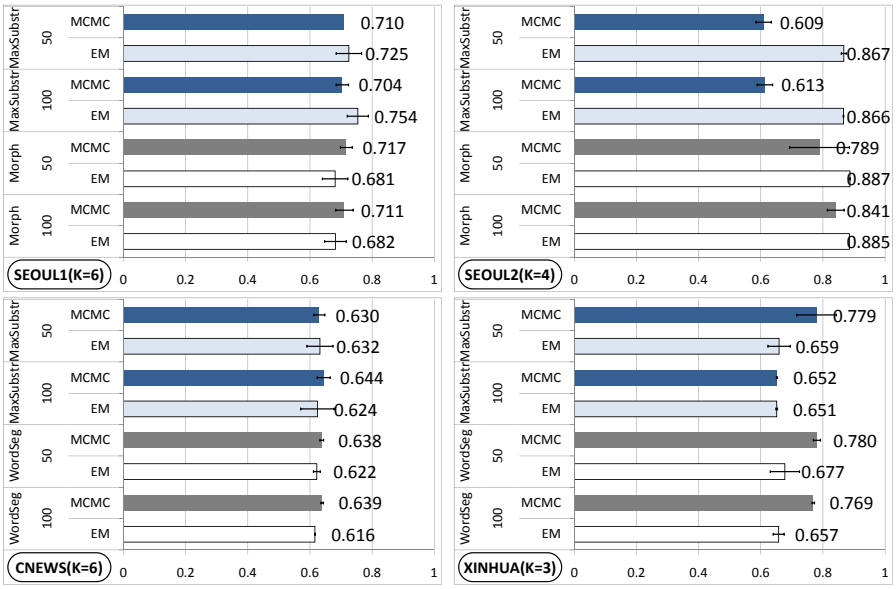

Fig. 2. Comparing between the F-scores achieved with bag of maximal substrings representation and those achieved with bag of words representation.

The bottom left chart shows the results for CNEWS. Recall that we employed an additional reduction of the variety of maximal substrings for CNEWS and XINHUA. To be specific, we remove the maximal substrings of length one from the 1,000 highest frequency maximal substrings. For CNEWS, the chart shows that all mean F-scores are almost the same. In fact, the best mean F-score obtained with maximal substrings and that obtained with the words extracted by our CRF-based word segmenter gave no significant difference based on a one-tailed Student's t-test with p value less than 0.05.

The bottom right chart provides the results for XINHUA. For this data set, the MCMC sampling is likely to give better results than the EM. The best mean F-score obtained with maximal substrings and that obtained with the words extracted by our CRF-based word segmenter gave no significant difference based on a one-tailed Student's t-test with p value less than 0.05. With respect to bag of maximal substrings representation, the mean F-score dropped when we set n_L to 100, though, for CNEWS data set, the two settings of n_L gave no significant differences. Therefore, we need a little care in reducing the variety of maximal substrings for Chinese documents lest we remove too many low frequency ones.

Based on Figure 2, we can draw the following considerations. First, we should try both the EM algorithm and the MCMC sampling in document clustering, because which one performs better depends on the data. For example, we can conduct a rough comparison on a hold out document set. This argument applies both to maximal substrings and the words extracted by a supervised method.

Second, with respect to the comparison between the clustering quality given by bag of maximal substrings representation and that given by bag of words representation, which one performs better again depends on the data. However, the difference is not so large. Therefore, we can use one among the two representations consistently. When we have a hand-maintained dictionary or a human annotated training data set of large enough size for the tasks we envision, we can consistently use a supervised word extraction, because M in Table 2 is far smaller for bag of words representation than for bag of maximal substrings representation and thus can conduct text mining tasks efficiently.

However, out of vocabulary words, i.e., the words not contained in training data sets, may have a serious effect on text mining sitting on a supervised word extraction. This is the very reason why an elaborated word n-gram model was proposed in [13]. Further, text data available from SNS environments are a typical example where we can observe a wide variety of out of vocabulary words, because SNS users are likely to coin new terms, e.g. hard-to-understand abbreviations and homophones derived from widely used words, without hesitation. In such a case, unsupervised word extraction will show an advantage.

5 Conclusions

As text data originating from SNS environments come to show a wider diversity in writing style or vocabularies, unsupervised extraction of elementary building blocks from documents becomes more important as a preprocessing for various text mining techniques than before. This paper provided the results where we compare bag of maximal substrings representation with bag of words representation in a typical text mining task, i.e., in document clustering, because maximal substrings can be efficiently extracted in an unsupervised manner.

Our results showed that bag of maximal substrings representation was as effective as bag of words representation. While the two representations may show a statistically significant difference in their effectiveness, the winner changes from data to data. Further, the difference is not so large to prevent us from adopting one representation consistently. With respect to the running time and memory space of document clustering, bag of maximal substrings representation showed no advantage, because the number of maximal substrings is far larger than that of the words extracted by a supervised method from the same document set. However, when we use a supervised word extraction, we should update a training data set constantly, because it is a fact that many new words are coined day by day especially in SNS environments.

Therefore, it must be an important future work to acquire a more realistic insight with respect to the trade-off between the following two types of cost:

– the execution time and memory space required for a text mining task conducted on a set of documents represented as bags of maximal substrings; and
– the hours and money required for preparing and constantly updating training data sets used in a supervised word extraction.

In addition, a method for further reducing the variety of maximal substrings is required to reduce the running time of mining tasks using maximal substrings.

We also have a plan to conduct experiments where we use maximal substrings as elementary building blocks of DNA/RNA sequences. We would like to propose a multi-topic analysis, e.g. by using latent Dirichlet allocation [2], with maximal substrings and to revise the results reported in [3], where the authors simply use k-mers of fixed length as elementary building blocks of DNA/RNA sequences.

Acknowledgement

This work was done as a joint research with National Institute of Informatics (NII) and was also supported in part by Nagasaki University Strategy for Fostering Young Scientists with funding provided by Special Coordination Funds for Promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology (MEXT).

References

1. Abouelhoda, M., Ohlebusch, E., Kurtz, S.: Optimal Exact String Matching Based on Suffix Arrays. In: Laender, A.H.F., Oliveira, A.L. (eds.) SPIRE 2002. LNCS, vol. 2476, pp. 31–43. Springer (2002)
2. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
3. Chen, X., Hu, X., Shen, X., Rosen, G.: Probabilistic Topic Modeling for Genomic Data Interpretation. In: Park, T., Tsui, S.K.-W., Chen, L., Ng, M.K., Wong, L., Hu, X. (eds.) IEEE International Conference on Bioinformatics and Biomedicine, pp. 18-21. IEEE (2010)
4. Choi, K.-S., Isahara, H., Kanzaki, K., Kim, H., Pak, S.M., Sun, M.: Word Segmentation Standard in Chinese, Japanese and Korean. In: 7th Workshop on Asian Language Resources, pp. 179-186. Association for Computational Linguistics (2009)
5. Chumwatana, T., Wong, K., Xie, H.: An Automatic Indexing Technique for Thai Texts Using Frequent Max Substring. In: Imsombut, A. (ed.) Eighth International Symposium on Natural Language Processing, pp. 67–72. IEEE (2009)
6. Chumwatana, T., Wong, K., Xie, H.: A SOM-Based Document Clustering Using Frequent Max Substrings for Non-Segmented Texts. Journal of Intelligent Learning Systems & Applications 2, 117–125 (2010)
7. Kasai, T., Lee, G., Arimura, H., Arikawa, S., Park, K.: Linear-Time Longest-Common-Prefix Computation in Suffix Arrays and Its Applications. In: Amir, A., Landau, G.M. (eds.) CPM 2001. LNCS, vol. 2089, pp. 181–192. Springer (2001)
8. Gang, S.: Korean Morphological Analyzer KLT Version 2.10b. http://nlp.kookmin.ac.kr/HAM/kor/ (2009)
9. Li, Y., Chung, S.M., Holt, J.D.: Text Document Clustering Based on Frequent Word Meaning Sequences. Data & Knowledge Engineering 64, 381–404 (2008)
10. Madsen, R., Kauchak, D., Elkan, C.: Modeling Word Burstiness Using the Dirichlet Distribution. In: Raedt, L.D., Wrobel, S. (eds.) 22nd International Conference on Machine Learning, pp. 545–552. ACM (2005)

11. Masada, T., Shibata, Y., and Oguri, K.: Documents as a Bag of Maximal Substrings: An Unsupervised Feature Extraction for Document Clustering. In: 13th International Conference on Enterprise Information Systems, pp.5–13. INSTICC (2011)
12. Minka, T.: Estimating a Dirichlet Distribution. http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/ (2000)
13. Mochihashi, D., Yamada, T., Ueda, N.: Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling. In: Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the Fourth International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, pp. 100–108. Association for Computational Linguistics (2009)
14. Navarro, G., Mäkinen, V.: Compressed Full-Text Indexes. ACM Comput. Surv. 39(1) (2007)
15. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text Classification from Labeled and Unlabeled Documents Using EM. Machine Learning 39(2/3), 103–134 (2000)
16. Nong, G., Zhang, S., Chan, W.H.: Two Efficient Algorithms for Linear Time Suffix Array Construction. IEEE Transactions on Computers 99(PrePrints) (2008)
17. Okanohara, D., Tsujii, J.: Text Categorization with All Substring Features. In: Ninth SIAM International Conference on Data Mining, pp. 838–846. Society for Industrial and Applied Mathematics (2009)
18. Poon, H., Cherry, C., Toutanova, K.: Unsupervised Morphological Segmentation with Log-Linear Models. In: Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 209–217. Association for Computational Linguistics (2009)
19. Sutton, C., McCallum, A.: An Introduction to Conditional Random Fields for Relational Learning. In: Getoor, L., Taskar, B. (eds.) Introduction to Statistical Relational Learning, pp. 93–128. The MIT Press (2007)
20. Teh, Y.W.: A Hierarchical Bayesian Language Model Based on Pitman-Yor Processes. In: the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 985–992. Association for Computational Linguistics (2006)
21. Tseng, H., Chang, P., Andrew, G., Jurafsky, D., Manning, C.: A Conditional Random Field Word Segmenter for SIGHAN Bakeoff 2005. In: Fourth SIGHAN Workshop on Chinese Language Processing, pp. 168–171. Association for Computational Linguistics (2005)
22. Tsuruoka, Y., Tsujii, J., Ananiadou, S.: Stochastic Gradient Descent Training for L1-Regularized Log-Linear Models with Cumulative Penalty. In: Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the fourth International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, pp. 477–485. Association for Computational Linguistics (2009)
23. Wang, X., McCallum, A.: Topics over Time: a Non-Markov Continuous-Time Model of Topical Trends. In: 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 424–433. ACM (2006)
24. Zhang, D., Dong, Y.: Semantic, Hierarchical, Online Clustering of Web Search Results. In: Yu, J.X., Lin, X., Lu, H., Zhang, Y. (eds.) APWeb 2004. LNCS, vol. 3007, pp. 69–78. Springer (2004)
25. Zhang, D., Lee, W.: Extracting Key-Substring-Group Features for Text Classification. In: 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 474–483. ACM (2006)