Automatic Text Summarization Using Latent Drichlet Allocation (LDA) for Document Clustering

Erwin Yudi Hidayat[a,1,*], Fahri Firdausillah[a,2], Khafiizh Hastuti[a,3], Ika Novita Dewi[a,4], Azhari[b,5]

[a] *Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang*
[b] *Computer Science and Electronics Department, Universitas Gajah Mada, Yogyakarta*
[1] *erwin@dsn.dinus.ac.id*; [2] fahri@dsn.dinus.ac.id; [3] afis@dsn.dinus.ac.id, [4] ikadewi@dsn.dinus.ac.id,*
[5] *arisn.softcomp@gmail.com*

ARTICLE INFO

Article history:
Received November 6, 2015
Revised November 25, 2015
Accepted November 30, 2015

Keywords:
LDA
Text Summarization
Clustering
K-means

ABSTRACT

In this paper, we present Latent Drichlet Allocation in automatic text summarization to improve accuracy in document clustering. The experiments involving 398 data set from public blog article obtained by using python scrapy crawler and scraper. Several steps of clustering in this research are preprocessing, automatic document compression using feature method, automatic document compression using LDA, word weighting and clustering algorithm The results show that automatic document summarization with LDA reaches 72% in LDA 40%, compared to traditional k-means method which only reaches 66%.

I. Introduction

Documents summarization process is a process to perform reduction of the volume of documents to be more concise, by taking the core documents and remove terms that considered unimportant without reduce the meaning of it. There are two types for creating summarization of a document, called abstract and extract. Abstract generate an interpretation of the original text, where a sentence would be transformed into shorter sentences [1]. Extraction is a summary of text obtained by restate passages that are considered as main topics in simplified form [2] [3]. This research will use features of summary extracts as a model of automatic document summarization.

The implementation of summarization techniques for documents clustering has a significant impact. This is due to the process of documents clustering usually constrained by the amount of the volume of documents. This problem is caused by large volumes of documents are identical with the size of document term matrix, whereas not all terms are relevant and sometimes term-redundant causes the process of clustering is not optimal [4].

In the model of automatic document summarization, Feature Based and Latent Dirichlet Allocation algorithm can be used for the sentence reduction process [5]. Previous studies show that Feature Based algorithms in the process of automatic document reduction as a feature for generating document clustering perform better in accuracy compared to standard feature reduction techniques [6] [7].

Document clustering is the process of document dataset grouping that refers to the similarity of document data patterns into a cluster. Meanwhile those document without similarity will be grouped into another clusters. [7]. K-means is one of the well-known cluster algorithm and frequently used to resolve clustering problem by grouping a certain number of *k* cluster, where the number *k* has been defined previously [8].

II. Literature Review

A. Web Crawler

A web crawler is one of the main components of the web search engines. The growth of web crawler is increasing in the same way as the web is growing. A list of URLs is available with the web

DOI: http://dx.doi.org/10.12928/ijain.v1i3.p132-139

crawler and each URL is called a seed. Each URL is visited by the web crawler. It identifies the different hyperlinks in the page and adds them to the list of URLs to visit. This list is termed as crawl frontier. Using a set of rules and policies the URLs in the frontier are visited individually. Different pages from the internet are downloaded by the parser and the generator and stored in the database system of the search engine. The URLs are then placed in the queue and later scheduled by the scheduler and can be accessed one by one by the search engine one by one whenever required. The links and related files which are being searched can be made available whenever required at later time according to the requirements. With the help of suitable algorithms web crawlers find the relevant links for the search engines and use them further. Databases are very big machines like DB2, used to store large amount of data

B. *Text Mining*

Text Mining can be visualized as consisting of two phases: (i) Text refining and (ii) Knowledge distillation. Text refining phase transforms the free form text documents into a chosen intermediate form. Knowledge distillation infers patterns or knowledge from intermediate form. The Intermediate Form (IF) can be semi structured such as the conceptual graph representation or structured such as relational data representation. Intermediate form can be document based wherein each entity represents a document or concept based wherein each entity represents an object or concept of interests in specific domain. Mining a document based IF derives patterns and relationships across documents. Document clustering/visualization and categorization are examples of mining from a document based IF.

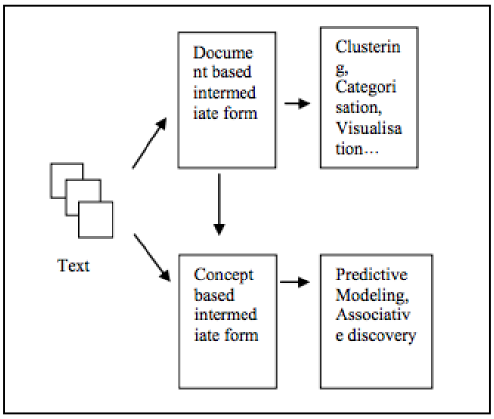
Docume nt based intermed iate form
Clusterin g, Categori sation, Visualisa tion…
Text
Concept based intermed iate form
Predictive Modeling, Associativ e discovery

Fig. 1.General framework for Tex Mining

There are seven types of text mining:

- Search and Information Retrieval (IR): Storage and retrieval of text documents, including search engines and keyword search.
- Document Clustering: Grouping and categorizing terms, snippets, paragraphs or documents using data mining clustering methods.
- Document Classification: Grouping and categorizing snippets, paragraphs, or document using data mining classification methods, based on models trained on labeled examples.
- Web Mining: Data and Text Mining on the Internet with a specific focus on the scale and interconnectedness of the web.
- Information Extraction (IE): Identification and extraction of relevant facts and relationships from unstructured text; the process of making structured data from unstructured and semi-structured text.
- Natural Language Processing (NLP): Low-level language processing and understanding tasks (e.g., tagging part of speech); often used synonymously with computational linguistics
- Concept Extraction: Grouping of words and phrases into semantically similar groups

C. Clustering Method

The k-means algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares. This algorithm requires the number of clusters to be specified. It scales well to large number of samples and has been used across a large range of application areas in many different fields. The k-means algorithm divides a set of *N* sample *X* into *K* disjoint clusters *C*, each described by the mean μ_j of the samples in the cluster. The means are commonly called the cluster *centroids*; note that they are not, in general, points from *X*, although they live in the same space. The k-means algorithm aims to choose centroids that minimize the inertia, or within-cluster sum of squared criterion:

$$\sum_{i=0}^{n} \underset{\mu_{j\in C}}{min}(||x_j - x_i||)^2 \tag{1}$$

Inertia, or the within-cluster sum of squares criterion, can be recognized as a measure of how internally coherent clusters are. It suffers from various drawbacks:

- Inertia makes the assumption that clusters are convex and isotropic, which is not always the case. It responds poorly to elongated clusters, or manifolds with irregular shapes.
- Inertia is not a normalized metric: we just know that lower values are better and zero is optimal. But in very high-dimensional spaces, Euclidean distances tend to become inflated (this is an instance of the so-called "curse of dimensionality"). Running a dimensionality reduction algorithm such as PCA prior to k-means clustering can alleviate this problem and speed up the computations.

K-means is often referred to as Lloyd's algorithm. In basic terms, the algorithm has three steps. The first step chooses the initial centroids, with the most basic method being to choose *k* samples from the dataset *X*. After initialization, K-means consists of looping between the two other steps. The first step assigns each sample to its nearest centroid. The second step creates new centroids by taking the mean value of all of the samples assigned to each previous centroid. The difference between the old and the new centroids are computed and the algorithm repeats these last two steps until this value is less than a threshold. In other words, it repeats until the centroids do not move significantly.

Mean shift clustering aims to discover *blobs* in a smooth density of samples. It is a centroid based algorithm, which works by updating candidates for centroids to be the mean of the points within a given region. These candidates are then filtered in a post-processing stage to eliminate near-duplicates to form the final set of centroids.

Given a candidate centroid x_i for iteration *t*, the candidate is updated according to the following equation:

$$x_i^{t+1} = x_i^t + m(x_i^t) \tag{2}$$

Where $N(x_i)$ is the neighborhood of samples within a given distance around x_i and *m* is the *mean shift* vector that is computed for each centroid that points towards a region of the maximum increase in the density of points. This is computed using the following equation, effectively updating a centroid to be the mean of the samples within its neighborhood:

$$m(x_i) = \frac{\Sigma x_j \in N(x_i)_j K(x_j - x_i) x_j}{\Sigma x_j \in N(x_i)_j K(x_j - x_i)} \tag{3}$$

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centres of the latent Gaussians.

III. Methodology

A. Preprocessing Stages

Preprocessing stages are stages prior to the beginning of the process of clustering. This step is necessary in order to the document crawling outcomes are in proper form and can be used in the next following process.

In this paper, three stages of preprocessing are used, namely tokenization, stopword, and stemming.

1) Tokenization

Tokenizing stage is the stage cutting input string based on each word composing a sentence. For example,

Input text: “Membuat campuran warna”.

Tokenization result: Membuat
campuran
warna

2) Stopword

In stopword stage, irrelevant words in topic determination of a document will be eliminated, e.g. the word “di”, “pada”, “dari”, “atau”, and some other words in Bahasa Indonesia.

3) Stemming

Stemming is the stage of looking for the word’s root or the primary word of each word resulted from filtering. An example of this stage is as follows:

Filtering result: Membuat
campuran
warna

Stemming result: buat
campur
warna

B. Automatic Text Summarization

Automatic text summarization is a concise form of the document, which aims to eliminate terms that are considered irrelevant or redundant to keep the core meaning of the document. So that even though the related document has a large volume, the users are able to understand the core document meaning quickly and correctly [9] [10].

C. Feature based Method

There are some features based method phase used in this paper, as follows:

- Title feature

$$Score(S_i) = \frac{No.title\ word\ in\ S_i}{No.word\ in\ title} \qquad (4)$$

- Sentence length

$$Score(S_i) = \frac{No.word\ occuring\ in\ S_i}{No.word\ occuring\ in\ longest\ sentence} \qquad (5)$$

- Term weight

$$Score(S_i) = \frac{Sum\ of\ TF{-}IDF\ in\ S_i}{Max(Sum\ of\ TF{-}IDF)} \qquad (6)$$

- Sentence position

$$Score(S_i) = 1 for\ First\ and\ Last\ sentence, 0\ for\ other\ sentences \quad (7)$$

- Sentence to sentence similarity

$$Score(S_i) = \frac{Sum\ of\ Sentence\ Similarity\ in\ S_i}{Max(Sum\ of\ Sentence\ Similarity)} \quad (8)$$

- Proper noun

$$Score(S_i) = \frac{No.proper\ nouns\ in\ S_i}{Length(S_i)} \quad (9)$$

- Thematic word

$$Score(S_i) = \frac{No.thematic\ word\ in\ S_i}{Length(S_i)} \quad (10)$$

- Numerical data

$$Score(S_i) = \frac{No.numerical\ data\ in\ S_i}{Length(S_i)} \quad (11)$$

D. K-Means

In pseudo code, k-means is as follow:

```
Initialize $\mathbf{m}_i$, $i = 1,\ldots,k$, for example, to k random $\mathbf{x}^t$
Repeat
    For all $\mathbf{x}^t$ in X
        $b_i^t \leftarrow 1$ if $\| \mathbf{x}^t - \mathbf{m}_i \| = \min_j \| \mathbf{x}^t - \mathbf{m}_j \|$
        $b_i^t \leftarrow 0$ otherwise
    For all $\mathbf{m}_i$, $i = 1,\ldots,k$
        $\mathbf{m}_i \leftarrow$ sum over t $(b_i^t \mathbf{x}^t)$ / sum over t $(b_i^t)$
Until $\mathbf{m}_i$ converge
```

The vector **m** contains a reference to the sample mean of each cluster. **x** refers to each of our examples, and **b** contains our "estimated [class] labels"

E. Latent Semantic Analysis

Latent Dirichlet Allocation (LDA) is a statistical model that tries to capture the latent topics in a collection of documents. LDA was first introduced by David Blei in 2003 [7]. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. One important assumption about the LDA generative model is that the number of topics is known in advance.

F. Vector Space Model Document Representation

Vector Space Model (VSM) changes document collection into a term-document matrix [9]. In figure 1, *d* refers to document and *w* is the weight or value for each term.

$$A_{mxn} = \begin{array}{c} \begin{matrix} d_1 & d_2 & \cdots & d_n \\ \downarrow & \downarrow & & \downarrow \end{matrix} \\ \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix} \end{array} \begin{matrix} \\ \\ \leftarrow t_1 \\ \leftarrow t_2 \\ \vdots \\ \leftarrow t_m \end{matrix}$$

Fig. 2. Document-term matrix

G. Term Frequency–Inverse Document Frequency (TF-IDF)

As its name, Term Frequency or simply TF shows how often a term appears in a document. Inverse Document Frequency (IDF) is a calculation of division logarithm of documents number with document frequency containing a term. Term Frequency–Inverse Document Frequency is a result by multiplying TF to IDF for a term in document.

$$TF = log\frac{D}{DF} \tag{12}$$

$$TFIDF_{(t)} = TF * log\frac{D}{DF} \tag{13}$$

H. Similarity Measure

In this study, calculating similarity in documents was carried out by measure the distance the distance between two documents d_i and d_j, using the cosines similarity formula. In the VSM, document is represented as $d = \{w_1, w_2, w_3, w_n\}$, where d is document and w is the weight of each term in document [14]. Similarity measeure is shown as follow:

$$Similarity\ (d_i, d_j) = cosines\ \theta = \frac{\overrightarrow{d_\iota}.\overrightarrow{d_J}}{||d_i|.|d_i||} \tag{14}$$

I. Evaluation Measure

There are several techniques to measure performance quality of document clustering model, such as information matrix, misclassification index, purity, and F-measure. This study utilizes the last mentioned technique. F-measure measurement is based on precision and recall values. The higher the both values, the better accuracy of the document clustering result.

Recall and precision of category *i* in cluster *j* are obtained from:

$$Recall = \frac{n_{ij}}{n_i} \tag{15}$$

$$Precision = \frac{n_{ij}}{n_j} \tag{16}$$

where n_{ij} is document number of category *i* in cluster *j*, n_i is document number of category *i*, and n_j is document number in cluster *j*.

F-measure is defined as follow:

$$F(i,j) = \frac{2*(precision*recall)}{(precision+recall)} \tag{17}$$

The mean of F-measure calculation is conducted by:

$$F = \Sigma_i \frac{n_{ij}}{n_i}\ max_j = 1, \ldots, k\ F(i,j) \tag{18}$$

max *F(i, j)* is the maximum F-measure value of category *i* in cluster *j*.

IV. Implementation and Result

A. Dataset

This research uses dataset which consists of 398 articles in Bahasa Indonesia, obtained from public blog article by using python scrapy crawler and scraper. This dataset then transformed into certain form to acquire relevant attributes, match to input format of the document clustering algorithm.

For these 398 articles, the authors categorize manually into five different section: economy news, market reports, government, finance, and finance information.

B. Performance Evaluation Measure

Evaluation is done by observing the clustering results from testing the proposed method using the LDA algorithm. This study used the F-measure to measure the clustering performance. F-measure is obtained from the measurement of recall and precision. Recall is the ratio of acquired relevant documents by the total number of documents in documents collection. Meanwhile, precision is the ratio of the retrieved relevant documents number with a whole number of retrieved documents. Validation of the results is carried out by comparing evaluation method result of the method.

Table 1 below is a comparison of the results from several tested models and the proposed model. Results show that improvement in accuracy occurs in clustering Bahasa Indonesia documents by using LDA method. The highest average accuracy was obtained using the automatic document summarization with LDA that reaches 72% in LDA 40%. Compared to traditional k-means method which only reaches 66%.

Table 1. Experimental result

Methods	**Result**						
	F-Measure 1	***F-Measure 2***	***F-Measure 3***	***F-Measure 4***	***F-Measure 5***	***Average***	***Percentage***
K-Means	0.6130	0.6240	0.6333	0.6456	0.7100	0.6452	65%
Feature-based 20 %	0.6233	0.6544	0.6500	0.6780	0.6900	0.6591	66%
Feature-based 40 %	0.7186	0.7030	0.6997	0.6870	0.6960	0.7009	70%
Feature-based 60 %	0.6450	0.6678	0.7440	0.7000	0.6613	0.6836	68%
LDA 20 %	0.7013	0.7120	0.6656	0.6877	0.6900	0.6913	69%
LDA 40 %	0.6773	0.7455	0.7500	0.6986	0.7211	0.7185	72%
LDA 60 %	0.8026	0.7044	0.7288	0.6522	0.6497	0.7075	71%

From table 1 above can also be seen that in overall, automatic text summarization performs better than clustering without automatic text summarization. Fig 3 depicts a complete comparison in bar chart.

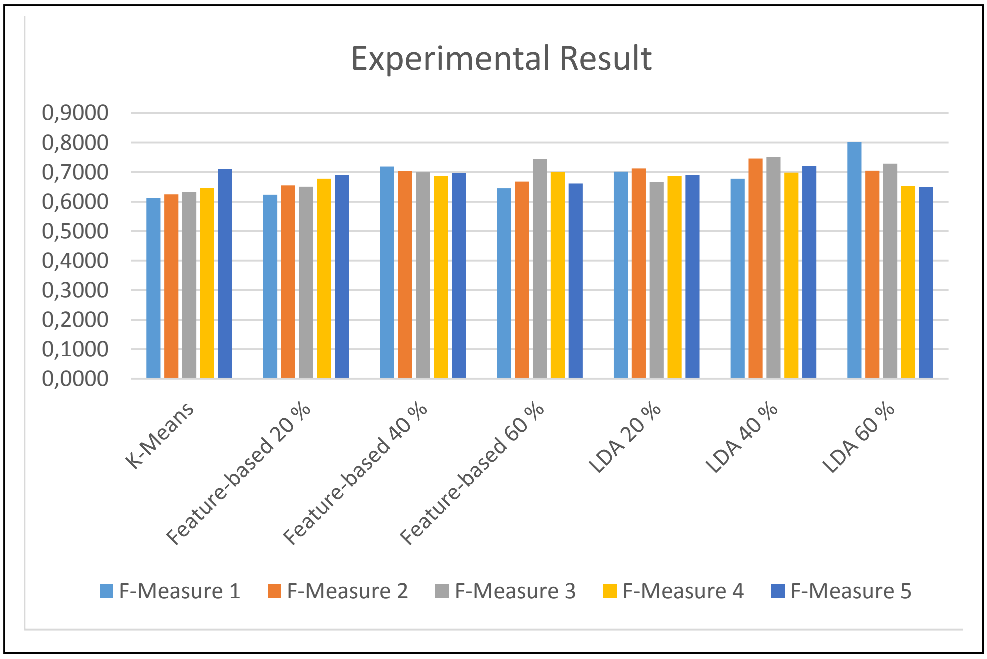

Fig. 3. Comparison of experimental result

V. Conclusion

In this paper, we have presented LDA automatic text summarization for document clustering in Bahasa Indonesia. Our experiments involving 398 data set from public blog article by using python scrapy crawler and scraper. Comparing our summarizer with traditional k-means and feature-based method, the results show that the best average precision text summarization for document clustering produced by the LDA method. Certainly, the experimental result is based LDA could improve the accuracy of document clustering.

References

[1] Changqiu Sun, Xiaolong Wang & Jun Xu, “Study on Feature Selection in Finance Text Categorization,” International Conference on Systems, Man, and Cybernetics Proceedings of the 2009 IEEE

[2] H. Al-mubaid and A.S. Umair, "A new text categorization technique using distributional clustering and learning logic," IEEE Trans. Knowl. Data Eng, vol. 18, 2006, pp. 1156-1165.

[3] Ladda Suanmali, Naomie Salim & M Salem Binwahlan, “Automatic text summarization using feature based fuzzy extraction,” Jurnal teknologi Maklumat jilid 20. Bil 2, 2008.

[4] Luying Liu, Jianchu Kang, Jing Yu & Zhongliang Wang, “A Comparative Study on Unsupervised Feature Selection Methods for Text Clustering,” Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE '05. Proceedings of 2005 IEEE International Conference.

[5] Manika Kar, Sérgio Nunes, Cristina Ribeiro, “Summarization of Changes in Dynamic Text Collections Using Latent Dirichlet Allocation Model,” Journal of Information Processing & Management, vol 51, no. 6, 2015, pp. 809-833

[6] Tao Liu, Shengping Liu, Zheng Chen & Wei-Ying Ma, “An Evaluation on Feature Selection for Text Clustering,” Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

[7] L. Muflikhah & B. Baharudin, "Document Clustering using Concept Space and Cosine Similarity Measurement," International Conference on Computer Technology and Development, Kota Kinabalu: 2009, pp. 58 - 62.

[8] W. Song and S. C. Park, “A Novel Document Clustering Model Based on Latent Semantic Analysis,” pp. 539–542, 2007.

[9] Krysta M. Svore, Lucy V., & Christopher J.C. Burges, “Enhancing Single-document Summarization by Combining RankNet and Third-party Sources,” Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 448–457, Prague, June 2007.

[10] JIANG Xiao-Yu, FAN Xiao-Zhong, Wang Zhi-Fei & Jia Ke-Liang, “Improving the Performance of Text Categorization using Automatic Summarization,” International Conference on Computer Modeling and Simulation IEEE 2009.