Comparison between BIDE, PrefixSpan, and TRuleGrowth for Mining of Indonesian Text

Dian Sa’adillah Maylawati1,2, Mohamad Irfan1,3 and Wildan Budiawan Zulfikar1,4
1Departement of Informatics, State Islamic University of Bandung, Indonesia
2Email: diansm@uinsgd.ac.id
3Email: irfan.bahaf@uinsgd.ac.id
4Email: wildan.budiawan.z@uinsgd.ac.id

Abstract. Mining process for Indonesian language still be an interesting research. Multiple of
words representation was claimed can keep the meaning of text better than bag of words. In
this paper, we compare several sequential pattern algorithm, among others BIDE (BI-
Directional Extention), PrefixSpan, and TRuleGrowth. All of those algorithm produce frequent
word sequence to keep the meaning of text. However, the experiment result, with 14.006 of
Indonesian tweet from Twitter, shows that BIDE can produce more efficient frequent word
sequence than PrefixSpan and TRuleGrowth without missing the meaning of text. Then, the
average of time process of PrefixSpan is faster than BIDE and TRuleGrowth. In the other hand,
PrefixSpan and TRuleGrowth is more efficient in using memory than BIDE.

1. Introduction
Indonesia is one of the biggest social media user, such as Facebook, Twitter, Instagram, Path, and so
on. Mining Indonesian text from social media is interesting things. To get the meaning of text, we
have to keep the meaning well. Moreover, Indonesian language has a unique grammer, such as many
form to use the affixes that is combined in words. And also, in social media, many Indonesian slang
words or it is called “bahasa alay” which should be more attention so that the meaning is manitained.

Compared with bag of words, multiple of words is considered as good text representation [6][11]. As
we know, the meaning of text is stored in complete sentence, minimal own the subject and predicate.
Frequent word sequence [3][5][13] and Set of frequent word sequence [13] are text features based on
sequential pattern which is proved can maintain the meaning of text well. Because of sequential
pattern semantically can maintain the meaning of text by maintaining the relationship between words,
phrases, clauses and sentences [1][6][13]. Sequential pattern mining which is proposed by Agrawal
and Srikant [2][4] is one of the most popular technique to find relationship between event in large
database [8][9][15]. Sequential pattern mining will discover the frequent events or elements or
subsequence in sequence database [2][4][8][9][10][14][16]. Sequential pattern is called frequent when
it meets the minimum support or threshold that is given. Sequential pattern for text has a words as an
items, and it is called sequence of word. Frequent word sequence and set of frequent word sequence is
sequence of word [13].

Many previous study about efficient algorithm of sequential mining, among others BIDE (BI-
Directional Extention) [16], PrefixSpan [14], and TRuleGrowth [8]. In this paper, we compare BIDE,
PrefixSpan, and TRuleGrowth algorithm to discover sequence of word from Indonesian text. The form
of sequence of words from BIDE, PrefixSpan, and TRuleGrowth is frequent word sequence. The
different between frequent word sequence and set of frequent word sequence is frequent word sequence does not pay attention to sequence of sentence [13]. Otherwise, set of frequent word sequence pay attention to sequence of sentence [13].

In this paper, we compare BIDE, PrefixSpan, and TRuleGrowth algorithm because the algorithms have similarity to growth the pattern but different in use approach to process and efficiency. We compare about the efficiency and the ability of those algorithm to discovered frequent word sequence and to keep the meaning of Indonesian text.

2. Text preprocessing for Indonesian text

Text is an unstructure data that need special treatment prior to the mining process. Preprocessing is an important process that can affect the next process to get the best result, especially for text. Bad text preprocessing will exacerbate the meaning of the text custody.

For example, the meaning of text is contained in verb, if there is no stemming process for verb with affixes, the meaning of text will be lost. Because of many slang words in Indonesian text from social media, there is several process that is customized for Indonesian text case [11], among others:

- All of non letter character or regular expression will be deleted from text. Because non letter character such as punctuation must be in every text and it will make frequent. Then, the words will be changed to the same lowercase form.
- The next process is going to check natural language in text. Especially in social media, most of the text contains natural language, among others abbreviation words and slang. Abbreviation word will be changed to be the original form. We used abbreviation dictionary which save abbreviation and the original form, including abbreviation for slang words. For example, bc to be baca, sk to be suka, cnt to be cinta, cnta to be cinta, hdup to be hidup, and so on.
- Stop words removal will be done after all of words have changed to the original form. All of Indonesian slang words is also removed, such as gue, gw, loe, ini, itu, and so on.
- After stop words removal, the next important process is stemming. The stemming process is crucial, because verb contain the meaning of text. We use Porter Algorithm for stemming process which has been customized for Indonesian language and Indonesian slang. For example nyuci tu be cuci, memasak to be masak, ngerjain to be kerja, belajar to be ajar, bayarin to be bayar, and so on.
- After cleaning process, from remove regular expression until create text data sequence, next BIDE, PrefixSpan, and RuleGrowth algorithm will discovered the sequence of words from sequence database of the text. For example the ready text after text preprocessing described in Figure 1 with Indonesian text.
3. Discovering Frequent Word Sequence with PrefixSpan
PrefixSpan or Prefix-projected Sequential Pattern Mining is a sequential pattern mining algorithm that adopts a divide-and-conquer and pattern growth principle for efficient mining of sequential pattern in large sequence database [10][14]. PrefixSpan offers ordered growth and reduce projected database, better than a priori-based algorithm, such as GSP, FreeSpan, and SPADE. In the a priori-based algorithm, a huge set of candidate sequence could be generated in a large sequence database, do multiple scans of database in mining, and generate a combinatorially explosive number of candidate when mining long sequential pattern [2][4][14].

PrefixSpan introduce the concept of prefix and suffix where prefix suppose all the item within an element are listed alphabetically. While, sequence s’ is called the suffix of sequence s with regards to prefix or consecutive after prefix. Sequential pattern would be extracted locally from projected database through process of finding frequent items that appended to the prefix [14][15]. The algorithm of PrefixSpan is described below in paper [14][15]. The item of sequential pattern for text mining is the word. For example, from Figure 1, the text document collection will be a sequence database and if the minimum support for sequential pattern is 2, then the result of sequence of word in Table 1 that present with Indonesian text.

| Sequential Pattern | Freq. | Sequential Pattern | Freq. |
|---------------------|-------|---------------------|-------|
| <pkl>               | 3     | <jual, area, monumen> | 2     |
| <pkl, tertib>      | 3     | <area>              | 2     |
| <pkl, jual>        | 3     | <gasibu>            | 2     |
| <pkl, kawasan>     | 2     | <gasibu, jual>      | 2     |
| <pkl, gasibu, jual, area> | 2 | <tertib, area, monumen> | 2 |
| <pkl, gasibu, tertib, jual> | 2 | <gasibu, jual, area, monumen> | 2 |

4. Discovering Frequent Word Sequence with BIDE
BIDE or BI-Directional Extention is a sequential pattern mining algorithm without candidate maintenance for mining frequent closed sequence [10][16]. Another frequent closed itemset mining algorithms need to maintain the set of already mined closed sequence candidates that cause poor scalability in the number of frequent closed pattern. Because a large number of frequent closed pattern will occupy much memory and lead to large search space for the closure checking of new pattern, specially when the support threshold is low or the pattern become long.

BIDE do not need closure checking to keep track of any single candidate for new pattern [16]. BIDE propose a deep search space pruning method with the BackScan pruning method and ScanSkip optimization techniques. The algorithm of BIDE is described in reference number [16].

A frequent closed sequence is said to be close if there is no more subsequence s’ from sequence s and s’ has the same minimum support threshold as s. Based on the definition, from Table 2, frequent closed sequences that are produced among others <pkl> = 3; <pkl, tertib> = 3; <pkl, jual> = 3; <pkl, gasibu, jual, area, monumen> = 2; <pkl, kawasan, tertib> = 2; <pkl, pkl, boleh, jual, area, monumen> = 2. The number of feature of frequent closed sequence less than classic sequential pattern, but will not eliminate the meaning of text because the meaning of subsequence saved in supersequence.

5. Discovering Frequent Word Sequence with TRuleGrowth
TRuleGrowth is a modified version of RuleGrowth algorithm that accept a sliding-window constraint [8] for mining sequential rules common to several sequences uses a pattern-growth approach for discovering sequential rules [8][9]. Pattern-growth approach can be much more efficient and scalable. Sequence database S is a set of sequences s, S={s1, s2…sn}, and a set of items I={i1, i2,…in}, where each sequence s is an ordered list of itemsets s={X1, X2, … Xn} such that X1, X2, …Xn ⊆ I . A sequential rule X⇒Y is defined as a relationship between two itemsets X, Y ⊆ I such that X∩Y = Ø
and X, Y are not empty [8][9]. If the items of X occur in some itemsets of a sequence, then the items in Y will occur in some itemsets afterward in the same sequence as long as there is no ordering restriction between items in X and between items in Y. And also that items from X and items from Y do not need to appear in a same itemset in a sequence. The algorithm of RuleGrowth and TRuleGrowth is described in reference number [8].

TRuleGrowth same as RuleGrowth has two measure for the rule, the first measure is sequential rule support that defined as sup(X ⇒ Y) where denotes the number of sequence from sequence database where all the items of X appear before all items of Y. The second measure is sequential confidence that defined as conf(X ⇒ Y) where denotes the number of sequences that contains the items of X or the ratio of the number of Y to the number of X in all sequence in sequence database. From the example text data sequence in Figure 1 for 50% of minimum support and minimum confident will produce a sequence rule such as in Table 2 that present with Indonesian text.

| Rule | Supp. | Conf. | Rule | Supp. | Conf. |
|------|-------|-------|------|-------|-------|
| <pkl> ⇒ <kawasan> | 0.67 | 0.67 | <tertib> ⇒ <area, monumen> | 0.67 | 0.67 |
| <pkl, gasibu> ⇒ <tertib, jual> | 0.67 | 1 | <jual, area> ⇒ <monumen> | 0.67 | 1 |
| <pkl> ⇒ <tertib> | 1 | 1 | ... | ... | ... |

6. The Experiment and Result

Table 3. An example of sequence rule by PrefixSpan

| No | Label | Number of File | No | Label | Number of File | No | Label | Number of File |
|----|-------|---------------|----|-------|---------------|----|-------|---------------|
| 1  | Agama | 427           | 8  | Fashion | 191           | 15 | Kampanye | 450           |
| 2  | Bencana | 170           | 9  | Film    | 1011          | 16 | Keadilan | 580           |
| 3  | Bisnis | 752           | 10 | Gaul    | 437           | 17 | Keamanan | 68            |
| 4  | Bola   | 1557          | 11 | Hobi    | 211           | 18 | Kehidupan | 432          |
| 5  | Cantik | 1378          | 12 | Hukum   | 398           | 19 | Kejahatan | 342          |
| 6  | Cinta  | 4322          | 13 | Ibadah  | 495           | 20 | Komputer | 362          |
| 7  | Ekonomi | 226           | 14 | Idola   |               |    |         |               |

In accordance with the needs of research, text documents taken from 14.006 of Indonesian tweet of Twitter. Text documents are grouped into 20 directories according to the topic that labeled with Indonesian text (explained in Table 3). We use the text preprocessing library for Indonesian slang [12] and Sequential Pettern Mining Framework (SPMF) Library for BIDE, PrefixSpan, and TRuleGrowth algorithm for this experiment [7].

The experiments were conducted based on several test scenarios as follows:
- Examine the meaning of Indonesian language with sequence of word that produced by BIDE, PrefixSpan, and TRuleGrowth algorithm. And also comparing the accuracy of sequence of word between BIDE, PrefixSpan, and TRuleGrowth algorithm. We test the algorithms with several various of minimum support, among others 10%, 25%, and 50% of minimum support value. And for RuleGrowth, we use 10%, 25%, and 50% for minimum confident and 10 for size of window-sliding. We will get a number of sequence of word features from each algorithms that keep the meaning of text.
- Comparing the time process (in milisecond) and memory usage (in megabyte) between BIDE, PrefixSpan, and TRuleGrowth in discovered the sequence of word.

The evaluation of the results can be summarized as follows:
- PrefixSpan and BIDE algorithm are better in keep the meaning of text than TRuleGrowth algorithm. It is shown in Figure 8, Figure 9, and Figure 10. Surely, the number of sequence of
words or frequent word sequence that are produced depend on contain of text and text preprocessing. But, the experiment shows that with same minimum support, PrefixSpan and BIDE can produce frequent word sequence well. Instead BIDE can produce more efficient the number of frequent word sequence than PrefixSpan without losing the meaning of text. In other hand, TRuleGrowth can not find frequent word sequence that BIDE and PrefixSpan found. It shows in 25% and 50% minimum support, TRuleGrowth can not find frequent of words from the most of categories, only in “Idola” and “Kampanye” with bigger number of frequent word sequence than PrefixSpan. This causes the meaning of text is missing. This result is caused by dependence on minimum confident and window-sliding constraint in TRuleGrowth.

- The average of time process in Figure 2, Figure 3, and Figure 4 for producing frequent word sequence shows that PrefixSpan is the fastest. The average time process of PrefixSpan is 336,883 ms, BIDE is 3391,05 ms, and TRuleGrowth is 4057,37 ms. BIDE have higher time process because BIDE finds frequent closed sequence for efficiency number of feature. While, the average of time process between PrefixSpan is not much different from TRuleGrowth, but TRuleGrowth produces fewer frequent word sequence with longer time process than PrefixSpan and BIDE. Moreover, TRuleGrowth do not produce frequent word sequence mostly.

- For memory usage experiment, we find that PrefixSpan and TRuleGrowth are more efficient than BIDE. Not only from the average of memory usage but also TRuleGrowth and PrefixSpan is not use memory when do not fine the feature or frequent word sequence. It is different than BIDE that still use the memory though do not find the sequence of words. The average of memory usage of PrefixSpan is 3,603 mb, BIDE is 4,587 mb, and TRuleGrowth is 1,733 mb.
7. Conclusion
- There is many sequential pattern algorithm that used for sequential pattern mining. This research do comparison between PrefixSpan, BIDE, and TRuleGrowth algorithm for mining Indonesian text with frequent word sequence feature. The experiment with 14,006 tweets from Twitter shows that PrefixSpan and BIDE are better in keep the meaning of text than TRuleGrowth. Moreover, BIDE produces more efficient sequence of word with frequent closed sequence without missing the meaning of text. It was different than TRuleGrowth that do not produces features mostly.
- The average of time process of PrefixSpan shows that PrefixSpan is fastest. It is because BIDE must find frequent closed sequence for efficient feature and TRuleGrowth must have more time for satisfy minimum confident and window-sliding constraint. However, for the memory usage, PrefixSpan and TRuleGrowth are more efficient than BIDE because PrefixSpan and TRuleGrowth do not use memory when do not find the feature. It is different than BIDE that keep using memory though the feature is not found.
- For discussion and future work, we have to try do the experiment with larger or scalable or big text data for PrefixSpan, BIDE, and TRuleGrowth. Because BIDE and TRuleGrowth is match for scalable data. It does not rule out the possibility for comparing another sequential pattern or frequent pattern algorithm for Indonesian text.

8. References
[1] Adityarani, Kania. 2009. Aplicating CFWS* Algorithm for Document Grouping with Text representation Set of Frequent Word Sequence and Text Representation Qord Frequent itemset. Informatic, Bandung Institute of Technology.
[2] Agrawal, Rakesh, Srikant, Ramakrishnan. 1994. Fast Algorithm for Mining Association Rules. Proceeding of the 20th VLDB Converence. Santiago, Chile.
[3] Agrawal, Rakesh, et al. 1999. Mapping Words, Phrases using Sequential Pattern to Find User Specific Trends in a Text Database – Patent Document. International Business Machines Corporation, NY.
[4] Agrawal, Rakesh, Srikant, Ramakrishnan. 1995. Mining Sequential Patterns. IBM Research Center.
[5] Ahonen, Helena. 1999. Knowledge Discovery in Document by Extracting Frequent Word Sequence. Library Trends. Vol. 48, No. 1, Summer 1999, pp. 160-181.
[6] Doucet, Antoine. 2005. Advance Document Description, a Sequential Approach. Accademic Dissertation, Departemt of Computer Science, Series of Publication A. Report A-2005-2, University of Helsinki.
[7] Fournier-Voger, Philippe. 2008-2016. Sequential Pattern Mining Framework (SPMF) Data Mining Software Tools. Available in: http://www.philippe-fournier-viger.com/spmf
[8] Fournier-Viger, Philippe, Wu, Cheng-Wei. Tseng, Vincent S. Cao, Longbing. Nkambou, Roger. 19th of February 2015. Mining Partially-Ordered Sequential Rules Common to Multiple Sequences. IEEE Transactions on Knowledge and Data Engineering.
[9] Fournier-Viger, Philippe. Nkambou, Roger. Tseng, Vincent Shin-Mu. 2011. RuleGrowth: Mining Sequential Rules Common to Several Sequence by Pattern Growth.. SAC’11 Proceedings of the 2011 ACM Symposium on Applied Computing, pages 956-961.
[10] Han, Jiawei, Kamber, Micheline. 2006. Data Mining: Concepts and Techniques Second Edition. Simon Fraser University. Morgan Kaufmann Publisher.
[11] Hoonlor, Apirak. July 2011. Sequential Patterns and Temporal Patterns for Text Mining. A Thesis Submitted to Graduate Faculty of Rensselaer Polytechnic Institute in Partial Fulfillment of the Requirement for Degree of Doctor of Philosophy. Major Subject: Computer Science, New York.
[12] Maylawati, Dian Sa’dillah. 2011. “Development of Pre-processing Library with Set of Frequent Word Item Set Representation for Text Mining Application. Case Study: Indonesian’s Slang Language”. Master of Informatic Programs. STEI, Bandung Institute of Technology.
[13] Nalistia, Rindhu Astri. 2008. Feature-Based Clustering for documents grouping with the Structured Representation of Text. Informatic, Bandung Institute of Technology.
[14] Pei, Jian. Han, Jiawei, et.al. October, 2004. Mining Sequential Patterns by Pattern Growth: The PrefixSpan Approach. IEEE Transaction on Knowledge and Data Engineering, Vol. 16, No. 10.
[15] Sabrina, Puspita Nurul. Saptawati, G.A. Putri. November, 2015. Multiple MapReduce and Derivative Projected Database: New Approach for Supporting PrefixSpan Scalability. International Conference on Data and Software Engineering (ICoDSE)
[16] Wang, Jianyong. Han, Jiawei. 2004. BIDE: Efficient Mining of Frequent Closed Sequence. ICDE ’04 of 20th International Conference on Data Engineering.