Jurnal
Teknologi

Full Paper

A MALAY TEXT CORPUS ANALYSIS FOR SENTENCE COMPRESSION USING PATTERN-GROWTH METHOD

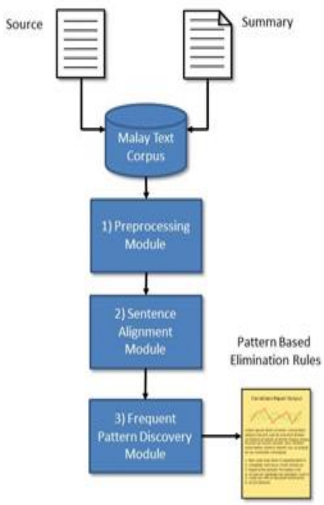
Article history
Received
16 February 2016
Received in revised form
4 April 2016
Accepted
15 July 2016

Suraya Alias[a]*, Siti Khaotijah Mohammad[b], Gan Keng Hoon[b], Tan Tien Ping[b]

[a]Faculty of Computing and Informatics, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia
[b]School of Computer Sciences, Universiti Sains Malaysia, 11800 USM Pulau Pinang, Malaysia

*Corresponding author
suealias@ums.edu.my

Graphical abstract


Source
Summary
Malay Text Corpus
1) Preprocessing Module
2) Sentence Alignment Module
3) Frequent Pattern Discovery Module
Pattern Based Elimination Rules

Abstract

A text summary extracts serves as a condensed representation of a written input source where important and salient information is kept. However, the condensed representation itself suffer in lack of semantic and coherence if the summary was produced in *verbatim* using the input itself. Sentence Compression is a technique where unimportant details from a sentence are eliminated by preserving the sentence's grammar pattern. In this study, we conducted an analysis on our developed Malay Text Corpus to discover the rules and pattern on how human summarizer compresses and eliminates unimportant constituent to construct a summary. A Pattern-Growth based model named Frequent Eliminated Pattern (FASPe) is introduced to represent the text using a set of sequence adjacent words that is frequently being eliminated across the document collection. From the rules obtained, some heuristic knowledge in Sentence Compression is presented with confidence value as high as 85% – that can be used for further reference in the area of Text Summarization for Malay language.

Keywords: Sentence Compression, Pattern-Growth, Text Summarization, Malay

Abstrak

Satu ekstrak ringkasan teks berfungsi sebagai satu perwakilan padat kepada sumber input bertulis, di mana maklumat yang berkaitan dan penting akan disimpan. Walau bagaimanapun, perwakilan padat itu sendiri mempunyai kekurangan dari segi semantik dan kepaduan jika ringkasan itu dihasilkan dengan hanya menyalin kesemua ekstrak input itu sendiri. Pemampatan Ayat ialah satu teknik di mana butir-butir yang tidak penting dari sesebuah ayat dibuang dengan mengekalkan tatabahasa ayat tersebut. Dalam kajian ini, kami telah menjalankan analisis ke atas Korpus Teks Bahasa Melayu untuk mencari kaedah-kaedah dan corak bagaimana manusia memampatkan dan menghapuskan unsur tidak penting untuk membina suatu ringkasan. Satu model berdasarkan Pola-Berkembang dinamakan Pola Penyingkiran Kerap (FASPe) diperkenalkan untuk mewakili teks dengan menggunakan satu set perkataan bersebelahan yang sering dihapuskan di seluruh koleksi dokumen itu. Daripada peraturan yang diperoleh, beberapa pengetahuan heuristik mengenai Pemampatan Ayat dibentangkan dengan nilai keyakinan setinggi 85% yang boleh digunakan untuk rujukan lanjut dalam bidang Peringkasan Teks untuk bahasa Melayu

Kata Kunci: Pemampatan Ayat, Pola-Berkembang, Peringkasan Teks, Bahasa Melayu

1.0 INTRODUCTION

With the vast amount of information made available online; users are overwhelmed in digesting information that is deemed important to them. Therefore, a summary can assist in providing users with the insights of related articles without having to go through a time consuming of extensive reading. The task of a summarizer system is to produce a condensed representation from the source by preserving the salient context, as described by [1, 2]. However, if an extract summary being produced *verbatim* (copying the whole extract) from its source; the sentence may contain inessential information along with salient information that may effect on the overall coherence in the summary generation [3-5]. This special case in Text Summarization is known as Sentence Compression (SC); where given a sentence, the task is to produce a compact and informational content that is also grammatically correct [6, 7].

A human summarizer has the basic linguistic knowledge in filtering, selecting and eliminating terms or phrases that is less important while preserving the context to be added in the summary. On the other hand, a summarizer system needs to learn this type of linguistic knowledge or "*text elimination rules*" where it can be discovered by analyzing the corpus of compressed summary being produced by human summarizer. Take for example a source sentence and a compressed summary sentence composed by the human summarizer;

Source Sentence

Mengulas lanjut, beliau berkata, waris akan sentiasa dimaklumkan mengenai perkembangan terkini kes tersebut.

Compressed Summary Sentence

Waris akan sentiasa dimaklumkan mengenai perkembangan terkini kes tersebut.

The phrase "Mengulas lanjut" and "beliau berkata" has been eliminated or dropped from the source sentence. However, the compressed summary sentence is still informative and the language grammar pattern is still preserved.

Literature works in Text Summarization focusing in Sentence Compression techniques has been an interest to researchers as a way to improvise the quality and coherence of the summary being produced. Some known techniques such as Rule Based [6, 8-10], Statistics [11-13], Machine Learning [14, 15], and also Integer Linear Programming (ILP)[16, 17] has been explored previously. Some recent works in this area also include Graph related optimization by [18-20] that has be applied in the Multi-Sentence Compression area.

Hence, we would like to extend the interest in this area by using a Patten Based approach in analyzing the Malay Text Corpus. The Pattern Based approach has an appealing feature that it can correlate between words in finding frequent non-consecutive pattern that can provide a natural way of text representation for sets of documents collection [21-23].

In this study, a Pattern-Growth based model named Frequent Eliminated Pattern (FASPe) is introduced to represent the text using a set of sequence adjacent words or phrases that is frequently being eliminated by the human summarizer while performing the summarization task. The core of the Pattern-Growth method is based on "*divide-and-conquer*" method; where the dataset is divided into smaller sets based on current discovered frequent patterns. The subsequence pattern is then conquered based on the local frequent pattern; hence benefited in smaller search space in data structure with reduced candidate generation cost.

The total of 300 summaries with 2,058 sentence pairs (Source and Summary texts) are aligned and used in this Malay Text Corpus analysis. The discovered FASPe that is frequently being eliminated by experts is then converted to a set of text elimination rules for further morphological analysis. The confidence value derived from the rules can be used as a linguistic knowledge to better understand and later assist the Sentence Compression module in area of Text Summarization for Malay language.

2.0 LITERATURE REVIEW

This section reviews some existing work in Sentence Compression or also known as Sentence Reduction and elimination process. Next, the basis of our proposed Pattern Based approach in analyzing human's elimination pattern is presented.

Linguistic Rule Based Approach relies on training corpus of human summaries where heuristic linguistic rules or knowledge can be built upon understanding how human writes, removes and construct certain phrases in a summary. Prior works by Jing and McKeown [9] has shed lights on analyzing the "*cut-and-paste*" methods in human when composing a summary. Their decomposition work imitates how human reduce and combine sentences using the Hidden Markov Model using the training corpus of 300 news article with 1,642 sentences. The result shows that 78% from the total sentences are composed by humans using this method, and since there is no application of Part-of-Speech (POS) methods during preprocessing; the algorithm is considered as a straightforward and also language independent.

Extending the decomposition work from Jing and McKeown [9], Jing [6] developed an Automatic Sentence Reduction system. They added knowledge resources such as syntactic knowledge from WordNet, lexicon database, contextual and statistics information extracted from human summaries to assist the decision to remove a certain phrase from a sentence.

The removal decision of a certain constituent in a sentence derived from the Reduction Rules of;

1. If the phrase is not required grammatically (by referring to the syntactic parse tree),
2. The phrase is not important (based on the phrase importance score); and
3. It is based on removal probability value from humans practice.

However, the sentence parse tree in Jing [6] has a very high dependency on those additional resources in which large training corpus are needed with expensive resources. For a language with limited open source resources in POS such as Malay, a shallower yet effective approach is needed to understand the discovered pattern and linguistic rules from the training corpus.

Shallow Parsing or chunking is a method that aims to identify syntactic constituents such as noun or verb phrases (NP or VP) from a sentence. In Conroy *et al.* [8], their works has shown that by applying only shallow parsing and generating on the fly a list of "function" words that contains prepositions, conjunctions and determiners, etc. (which is applied on the lists of words identified to be trimmed such as adverbs, punctuation and gerunds) has yield better results in the ROUGE scores, an evaluation toolkit by Lin [24]; rather than highly being dependent on the POS tagger itself.

Meanwhile, Zajic *et al.* [10] applied sets of linguistic motivated rules to the compression process iteratively to each source sentences before moving to the sentence extraction module also known as the "*parse-and-trim*" approach. This approach is suitable to be applied in generating News headlines where it relies heavily on the maximum length of words threshold. Instead, the iterative approach has shown being less practical for a large corpora as reviewed in Nenkova and McKeown [4]. This is because it involves a substantial parsing and compressing process beforehand on well-built and informative grammatical sentence alongside with a non-informative sentence; making it a redundant iterative process.

Another common SC problem definition was given by Knight and Marcu [13]; where it is expressed as a word deletion problem: Given an input source sentence of words x = x1, x2, . . . , xn, the aim is to produce a target compression by removing any subset of these words. Two sentence compression algorithms were introduced by [12, 13]. The first uses a simple Statistical Probabilistic model to compress sentences in a noisy channel environment; while the other is based on Decision Tree model. By constructing parse trees of 1067 sentences from the Ziff-Davis Corpus, the Probabilistic model learns how likely each sentence is compressed using the Naïve Bayes Rules.

An example of a sentence parse tree (*t*) of the string *abcde* and the possible sentence compression tree of *s1* and *s2* to produce the string *abe* are shown in Figure 1. In order to determine which compression tree (*s1* or *s2*) has better probabilities, they computed it against the tree (*t*) and its expansion to see which compression tree is likely being used in the training corpus.

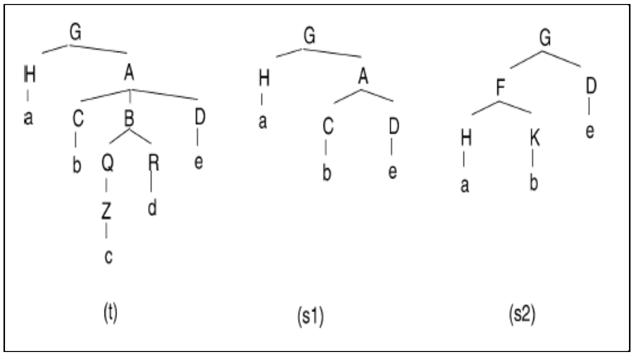


Figure 1 Example of Sentence Parse Tree by [13]

Conversely, a study by Lin [25] using the same noisy channel has reflected that even though the compression algorithm by [12, 13] has performed well grammatically in individual sentence level; it has insignificant effect on the overall Document Summarization system performance. This is due that during the compression process; some important content might have been dropped (deleted) since it is based purely on statistical syntactic approach. Following the drawbacks of this Statistical model, few researchers have made some improvement by applying Machine Learning approach on top of the baseline Statistical model.

For instance, Turner and Charniak [15] has combined statistical with unsupervised compression approach where additional rule and enforcement on the deletion constraint were added. Their result indicates some improvement on the grammatical area from the previous Statistical Model even though without using parallel training data. On top of that, Nguyen et *al.* [14] added semantic information from WordNeT to the Decision Tree Model where it has enhanced the accuracy of the sentence reduction algorithm based on the words importance; however the results on the grammatical and compression shows insignificant difference from the baseline model.

Later, Galley and McKeown [11] experimented on bigger training datasets (25% of the Ziff-Davis Corpus) using a Lexicalized Markov Grammars model. The model is able to cater the deletion rule problem faced in previous work; where their latter proposed head driven Markovization formulation allows them to lexicalize probabilities of the constituent before applying the deletion procedure. Another approach in solving the Sentence Compression problem is where the task is viewed as an optimization problem using Integer Linear Programming (ILP) method Clarke and Lapata [16], where the presence of linguistically motivated constraints has gained better performance over models without constraint. Similar works in ILP is also included in Cohn and Lapata [17], with additional operations such as substitution, reordering, and insertion, where it has also shown coherent output.

An extension from a Single Sentence Compression problem is known as Multi-Sentence Compression (MSC); where the task is to produce a compressed single sentence summary from a cluster of related sentences. Filippova [19] has introduced a straightforward approach based on the shortest paths in word graphs where it only requires a POS tagger and list of stop words. The idea is to eliminate redundant sentence by weighting the edge by i.e. link frequencies to search for the lightest and shortest path based on pre-defined minimum length.

Meanwhile, Boudin and Morin [18] presented an N-best reranking method based on key phrase extraction on top of Filippova's [19] idea in order to tackle the missing of salient information generated by the prior approach. Their work shows some improvement in the score of sentence informative value of the compressed summary but a slight decreased on grammatical evaluation value.

To summarize, most techniques has experimented on resourceful language with large training corpus such as English and French. Nevertheless, there are some works that shows even with straightforward method and using shallow parsing can assist in finding correlation between how human composes a summary; which has motivated this study to develop a Malay Text Corpus and to analyze the discovered pattern.

2.1 A Pattern Based Approach

The goal of Sequential Pattern Mining (SPM) approach is to discover hidden pattern in a large datasets where it started on transactional database [26]. Steadily this approach has been gaining interest in application to Text Database, where it has been used in the area of Document Classification, Topic Identification and also Document Summarization [27].

The basic problem of SPM can be generally defined as; "*Given a set of input sequences with user specified minimum support threshold value, the problem is to discover the set of patterns (frequent subsequences) that satisfies the given threshold*".

A pattern is basically a set of items or sequences that co-occur in a given dataset. A pattern is said as frequent or Frequent Pattern (FP), if the pattern occurs in the dataset more than the predefined minimum support threshold value. Also, if the Frequent Pattern is order-respected, then it is known as Frequent Sequential Pattern (FSP).

The SPM Algorithm can be generally categorized into 2 methods that is Apriori-based and Pattern Growth method. The concept of an Apriori-based also known as "*generate-and-test*" strategy is to generate a list of candidates before it was tested against the database scan in order to search for frequent items. Some known example of Apriori-based method are Generalized Sequential Pattern (GSP) Algorithm by Srikant and Agrawal [28] and SPADE by Zaki [29] that uses a Vertical representation format.

However, according to [30-32], the Apriori-based approach has three known limitations:

1. It has difficulties in mining problem that involves a large dataset with long frequent pattern,
2. It has potential of higher cost in candidate generation, and
3. It has to scan the database repeatedly in order to test the large candidates

In order to solve the aforementioned limitation in the Apriori-based method has led to the introduction of the Pattern-Growth method that implements the "*divide-and-conquer*" strategy; where algorithms such as PrefixSpan by Pei *et al.* [32] and FreeSpan by Han *et al.* [33] has shown to be more efficient. In the experiment conducted by Han *et al.* [34], the Pattern Growth method outperforms the Apriori-based method about an order of magnitude faster using a dense dataset with long frequent patterns.

The core of a Pattern-Growth method is "*it recursively divides (projects) the sequence database into smaller sets based on the current discovered frequent sequential pattern; and by using local frequent patterns, the sequential patterns are grown (conquered) in each projected database*".

The PrefixSpan Algorithm as illustrated in Figure 2 has three basic steps;

1. Find length-1 Sequential Pattern.
2. Divide search space.
3. Find subsets of sequential pattern.

```
Input: A sequence database S, minimum support threshold min_support.
Output: The complete set of sequential patterns.
Subroutine: PrefixSpan(α, L, S|α).
Parameters:
       α: sequential pattern,
       L: the length of α;
       S|α: the α-projected database, if α ≠<>;
       otherwise;the sequence database S.
Method: Call PrefixSpan(<>,0,S).
Subroutine PrefixSpan(α, L, S|α)
Step 1.
 Scan S|α once, find the set of frequent items b such that:
 b can be assembled to the last element of α to form a sequential pattern; or
 <b> can be appended to α to form a sequential pattern.
Step 2.
 For each frequent item b: append it to α to form a sequential pattern α' and output α';
Step 3.
 For each α': construct α'-projected database S|α' and call PrefixSpan(α', L+1, S|α').
```

Figure 2 The PrefixSpan Algorithm by [32]

There a few modifications that have been made to the prior PrefixSpan algorithm to suit our FASPe method namely;

1. The length-1 Sequential Pattern in our study is a list of frequently eliminated terms derived from aligning the human summary against the original source. This step will be explained further in section 3.
2. The next (*k+1*) sequence or adjacent terms is based on the sequential word order of the sentences in the original source document. This is done in order to preserve the semantic and grammatical flow of the news document.
3. The adjacent terms must also belong to the length-1 Sequential Pattern set; this is to reduce the effort in generating non-frequent candidate sequences and also to maintain small projection of the dataset.

In the next section, we will define the problem of this study and the application of Pattern-Growth approach in discovering Frequent Eliminated Pattern (FASPe) in Malay Text Corpus.

3.0 MALAY TEXT CORPUS ANALYSIS

A corpus can be defined as a collection of natural language data either in text or speech form. Reference to a public Malay corpus until to date is still very hard to find since the development are mainly focused for Academic use, thus the corpus is not being released publicly.

The input to our developed Malay Text corpus are multiple sets of Malay news articles with corresponding human summaries focusing on the Tragedy and Natural Disaster domain in Malaysia. In this section we will first define the problem of the study and following that the Pattern Based Malay Text Corpus Model analysis work is presented.

3.1 Problem Definition

By referring to the use of Sequential Pattern Mining in transactional databases defined by Agrawal and Srikant [26], the problem of finding "*textual elimination patterns*" or FASPe in a sentence can be formulated as:

"*Given an input sequence of words in a sentence-summaries collections database and a user specified minimum support threshold value; the problem of mining text data is to extract the frequent textual elimination patterns from the sentence-summaries text*."

A pattern or textual pattern here is a set of Frequent Adjacent terms or sequences that was eliminated in the manual summaries prepared by the panel of experts or human summarizer. Thus, a sequence of terms is said to be a textual elimination pattern if it was eliminated in a certain number of sentence summaries documents (more than the predefined threshold).

Definition 1:

Let S be a set of summaries; where each summaries d in S consists a list of sentences**.** We represent each $d = \{s_1, s_2, s_3 \ldots s_n\}$ where $s_1, s_2, s_3 \ldots s_n$ is a list of sentences in d.

A sentence $s \in d$ is a sequence of words or terms denoted as $s = \{t_1, t_2, t_3 \ldots t_n\}$, where $t_1, t_2, t_3 \ldots t_n$ is an ordered list of terms that was eliminated from the source article.

An *adjacent k-sequence* indicates a sequence of terms that follows (next-to) from the previous sequence in the sentences accordingly with the length of k.

For example the term <Mengulas> is a 1-sequence while <Mengulas lanjut> is a 2-sequence. When the term "lanjut" is the adjacent term from the word "Mengulas", it also can be written as "Mengulas -> lanjut", with the -> symbol indicating the term "lanjut" follows the term "Mengulas" in the sentence-summaries collection. Also, the adjacent sequence is said to be frequent if it fulfills the given threshold value.

Definition 2:

Given pattern *P* is a sequence of terms, the support of pattern *P* is the frequency of a sequence. There are two types of support value used in this study that is the global support denoted as *gSupp*; and the other is the elimination support denoted as *eSupp*.

The *gSupp* basically counts the frequency of the eliminated term being used against the overall source collection. Meanwhile, the *eSupp* is counted based on number of time the term occurs to be eliminated in the summary sentence when compared with actual aligned source article. If pattern *P* occurs more than once in the same sentences, the *gSupp* is still counted as one.

Given user specified minimum support threshold denotes as min_sup or σ; if the gSupp$(P) \geq \sigma$, then the sequence of *P* is considered as frequent or is a Frequent Pattern (FP).

Given user specified minimum confidence threshold denotes as min_conf or β, we can generate sequential rules that meets the confidence or Conf condition of the discovered FP.

Definition 3:

A *prefix_EliminatedTerms* or α is a set of terms of length-1 Sequential Pattern that is frequently eliminated in the overall sentence-summaries collection, denoted as α in this experiment. Given $\alpha = \{t_1, t_2, t_3 \ldots t_n\}$, where α will be used as a prefix to project the next adjacent sequence of the current document if and only if α is also a frequent locally (exist in the current sentence).

3.2 A Pattern Based Malay Text Corpus Model

Datasets

The Malay Text corpus consists of matching news source articles with summaries that was manually composed by Malay Language Expert. The total of 100 archived news articles were downloaded from the Bernama Library & Infolink Service (BLIS)[1]; a Malaysian news archive website using a keyword query such as "MH370" for the Tragedy dataset domain and "Banjir Kuala Krai" for Natural Disaster domain. These articles are then given to three Malay language experts where they manually prepare an extract summary of 30% length for each news article by selecting important sentences from the source, giving the total of 300 corresponding summaries.

Any modification such as the elimination of unimportant word or phrases from the selected sentence is permitted during the process; with respect of preserving the original source content and also the sentence sequence flow. Our language experts also perform some Shallow POS tagging exercise on the summary dataset which follows on the four main classes in Malay Language as classified by Nik Safiah Karim, Farid M Onn, and Musa [35]; that is Noun (Kata Nama or KN), Verbs (Kata Kerja or KK), Adjectives (Kata Adjektif or KA) and Function word (Kata Tugas or KT).

There are three modules involved in the analysis of the Pattern Based Malay Text Corpus namely;

1. Preprocessing,
2. Sentence Alignment and
3. Frequent Pattern (FASPe) Discovery.

Figure 3 depicts the flow of our Pattern Based Corpus model for Malay Language.

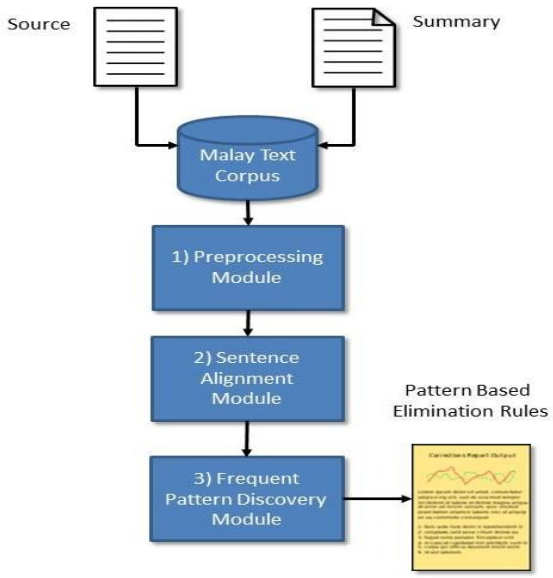

Source
Summary
Malay Text Corpus
1) Preprocessing Module
2) Sentence Alignment Module
Pattern Based Elimination Rules
3) Frequent Pattern Discovery Module

Figure 3 A Pattern Based Malay Text Corpus Model

3.2.1 Preprocessing Module

The preprocessing includes tasks such as tokenizing; punctuation removal; removal of Malay stop words and converting all terms to lowercase. No word stemming process is applied to the Malay text collection since our aim is to preserve the flow and linguistic pattern of the terms [36]. During the preprocessing, each document is tokenized into individual tokens using *space* to split each word and a full stop (.) as the delimiter to split a full sentence.

Table 1 shows the dataset statistics from the news source files, where 300 manual summaries is produced by human summarizer are used in this experiment to discover the linguistic elimination rules based on the Pattern-Growth approach. The number of unique terms is presented with and without the application of Malay stop words remover.

Table 1 Datasets Statistics

# of unique	Without SW	With SW
terms	5234	5116
Source Files	100	
Summary Files	300	

3.2.2 Sentence Alignment Module

In the Sentence Alignment Module, each sentence in the summary is matched or aligned with the news article source sentences. Some related reference in Sentence Alignment/Matching process using English corpus can be found in [9, 37]. Meanwhile, in our Malay Text corpus, the process to align the summary sentences with the original source sentence is divided into two;

1. Direct Aligned Sentences and
2. Join Aligned Sentences process.

The Direct Aligned Sentences process is done by matching the most overlapped terms from the summary sentences against the original source sentence; or one to one sentence match. In this process, each term from the summary sentence is compared if it is a subset from one of the original source sentence. This straight forward matching task counts how many terms from the summary sentences is contained or extracted verbatim from the original sentences.

The score for each matching term between the original news source sentence and summary sentences are calculated using Equation in (1); where t_i = 1, if the summary sentence, *s* contains the term *t* and 0 otherwise.

$$score(t_i) = \begin{cases} 1; \; if\ term\ t_i\ occur\ in\ s \\ 0 \quad otherwise \end{cases} \tag{1}$$

[1] http://blis.bernama.com/

The sum of score for each term in the aligned summary and source sentence process must be above overlapped threshold that is pre-set to 0.9 to maximize the direct matching of each term between the source and summary sentences.

If there is no Direct Aligned matching sentences between the summary and the original source is found for a certain corpus sentences; then further analysis is carried out to find out either the sentences is built upon joining few sentences.

The unmatched summary sentences are converted into a word vector and compared again to find the top *n*-most similar source sentence using Cosine Similarity Measure. The highest adjacent similar source sentences indicate the matching Join Aligned Sentences. Here the value of *n* is pre-set to 2, meaning the sentences in built upon joining 2 similar sentences in adjacent.

The Cosine Similarity is a normalized dot product between two vectors (v and w) or documents (d1 and d2) with measurement on the scale of (0,1). The higher the cosine value, the more similar the documents are. In this case, the more similar the original sentence with summary sentence. The equation for Cosine Similarity metric is defined in Equation (2):

$$sim_{cosine}\,(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}} \qquad (2)$$

After completing this module it is observed that there are several ways human summarizer composes a sentence summary which can be categorized further into 3 general methods;

1. Direct Aligned
2. Direct Aligned and Split
3. Join Aligned.

Table 2 shows the example of different summary composing method where eliminated phrases are struck through for easier visualization (S1, S2 is the source sentence, M1 is the summary sentence). For example in the Direct Aligned method, the human summarizer eliminates the phrase (*pada pukul 1.30 pagi semalam*) and extracted one to one sentence between the source articles to generate a summary sentence. This style is similar with the "*cut-and-paste*" method analyzed by [9].

Meanwhile, in the second method that is the Direct Aligned and Split, the expert eliminates the phrases such as (*salah satu*) and (*katanya…*) in the first sentence and split it into two summary sentences. In contrast, the Join Aligned method, the major content of the first sentence is joined with the phrase (*berdasarkan isyarat diperoleh radar tentera*), which is extracted adjacent from another sentence. This shows that there is an information linkage between the two join sentences that in the experts' opinion is important and needs to be added into the summary generation; while other irrelevant terms are eliminated based on the experts' linguistic knowledge.

Table 2 Example of manual summary sentence generation methods by human summarizer

1. Direct Aligned S1: Tentera Udara Diraja Malaysia (TUDM) mengesahkan pesawat MH370 yang hilang ~~pada pukul 1.30 pagi~~ semalam berpatah balik ke Lapangan Terbang Antarabangsa Kuala Lumpur (KLIA) sebelum terputus hubungan. M1: Tentera Udara Diraja Malaysia mengesahkan pesawat MH370 yang hilang semalam berpatah balik ke Lapangan Terbang Antarabangsa Kuala Lumpur sebelum terputus hubungan.
2. Direct Aligned and Split S1: ~~Salah satu~~ kemungkinan mengenai kehilangan pesawat ini adalah berpatah balik ke KLIA berdasarkan isyarat iperoleh radar tentera ~~dan buat masa ini~~ kita akan bekerjasama dengan agensi antarabangsa bagi mendapatkan gambaran lebih jelas, ~~katanya dalam sidang akhbar mengenai insiden kehilangan pesawat MH370 di Hotel Sama-Sama, di sini hari ini.~~ M1: Kemungkinan pesawat ini berpatah balik ke klia berdasarkan isyarat diperoleh radar tentera. M2: Kita akan bekerjasama dengan agensi antarabangsa bagi mendapatkan gambaran lebih jelas.
3. Join Aligned S1: Tentera Udara Diraja Malaysia (TUDM) mengesahkan pesawat MH370 yang hilang pada pukul 1.30 pagi semalam berpatah balik ke Lapangan Terbang Antarabangsa Kuala Lumpur (KLIA) sebelum terputus hubungan. S2: ~~Salah satu kemungkinan mengenai kehilangan pesawat ini adalah berpatah balik ke KLIA~~ (berdasarkan isyarat diperoleh radar tentera) ~~dan buat masa ini kita akan bekerjasama dengan agensi antarabangsa bagi mendapatkan gambaran lebih jelas, katanya dalam sidang akhbar mengenai insiden kehilangan pesawat MH370 di Hotel Sama-Sama, di sini hari ini.~~ M1: Tentera Udara Diraja Malaysia (TUDM) mengesahkan pesawat MH370 yang hilang pada pukul 1.30 pagi semalam berpatah balik ke Lapangan Terbang Antarabangsa Kuala Lumpur (KLIA) sebelum terputus hubungan berdasarkan isyarat diterima oleh radar tentera.

After the completion of this module, the findings are depicted in Table 3. From the 2,058 summary sentences; it is found that 95.7% of the sentences were extracted *verbatim* or a Direct Aligned from the source sentence by eliminating unimportant terms (noise). The balance of 4.3% from the summary sentences was generated using the Join Aligned sentences. The Direct Aligned and split method is still considered as Direct Aligned based on the overlapped threshold value. After analyzing through each datasets it is found that the highest percentage of terms elimination performed by the experts is 51%, the average is 29% and the lowest is 11%.

This findings has suggested that the terms are eliminated based on the linguistic knowledge of the panels to reduce the length of summary sentence by preserving only the gist of information or by joining information from different sentences that has the same meaning. Next, the task in discovering FASPe is discussed in detail.

Table 3 Statistics from the Sentence Alignment Module

Sentence Style	**# of sentences**
1) Direct Aligned Sentences	1,970 (95.7%)
2) Join Aligned Sentences	88 (4.3%)
Total Summary Sentences	**2,058**
Word Count (Source)	49,532
Word Count (Summary)	35,347
(% number of words eliminated per set)	
Highest	51%,
Average	29%
Lowest	11%

3.2.3 Frequent Pattern (FASPe) Discovery Module

In this module, each of the 2,058 summary sentences is compared against the source sentence to find a list of terms (length-1 or 1-gram) that was eliminated.

Since our goal is to discover a set of "Frequent Eliminated Pattern", the eliminated list should not consist terms that is the content or salient information of the summary. Thus, some filtering process has been applied to the raw list of eliminated terms in this stage before setting it as the prefix_EliminatedTerms set.

The steps in finding the FASPe followed the basis of PrefixSpan Algorithm by modifying on the generation of prefix_EliminatedTerms as mentioned in Section 2.1 and also in the pattern projection step. The steps in this module are as follows;

1. Get support of each eliminated terms, must be ≥ 2.
2. Set the eliminated terms as prefix_EliminatedTerms or α.
3. For each summary sentence, project the next Frequent Eliminated Pattern by joining the prefix_EliminatedTerms with the next frequent sequence (that also belongs to prefix_EliminatedTerms) in sequential order.

The sets of FASPe that was projected in step 3 are in sequential order of the source sentence to preserve the semantic and its grammatical roles. The generation of a FASPe for each document follows the equation in (3);

$$\mathrm{FASP_e} = \alpha + t_{(k+1)} \tag{3}$$

For example, the compressed sentence "~~Mengulas lanjut, beliau berkata~~, waris akan sentiasa dimaklumkan mengenai perkembangan terkini kes tersebut"; originally it has a sequence of term consist of "Mengulas lanjut", where it is a FASPe that was eliminated by human summarizer. This means that both of the term "Mengulas" and "lanjut" belongs to the prefix_EliminatedTerms set. The term "lanjut" is also a valid next frequent sequence based on the order of the sentence, thus by joining both terms, a new FASPe is discoved.

Next, the FASPe are sorted to find the support and confidence of each pattern. The preliminary findings from Malay Summary corpus analysis are presented in Table 4. The total of 202 FASPe or Frequent Eliminated Patterns has been found until the length 3 or tri-gram such as the phrase "dalam pada itu" in this stage.

Table 4 Number of FASPe discovered

FASPe	**# of eliminated patterns**
prefix_EliminatedTerms	147
length 2 (bigram)	49
length 3 (trigram)	6
Total FASPe	**202**

From Table 4, it is noted that as the length of terms increases, lesser FASPe is discovered. This is because, we only generate or project the sequence of FAPSe based on its usage and frequency in the source article; this is to avoid any additional cost in candidate generation and testing.

4.0 PATTERN BASED ELIMINATION RULES

Once the FASPe has been discovered, it can be used to generate rules to describe the relationship between different eliminated sequence terms in the Malay Text Corpus.

Definition 4:

Given a sentence-summaries sequence database, with parameter min_sup and min_conf, a Sequential Elimination Rule is the FASPe having the support and confidence higher than the min_sup and min_conf value.

A Sequential Rule X==>Y is a sequential relationship between two sets of terms or items where the term X and Y is in a sequential order. Since we observed two supports value in study, the confidence value (Conf) of a rule derived from dividing the elimination support (eSupp) against the global support (gSupp).

The basis of the Sequential Elimination Rule is to discover that for each time some eliminated terms or FASPe occurs in the news source, how many times it was eliminated by the human summarizer.

The Sequential Rule is defined in Equation (4);

$$\begin{gathered} X \rightarrow Y \text{ is a Sequential Rule if;} \\ Conf\ (X \rightarrow Y\ ) \geq min\ conf;where \\ Conf(X \rightarrow Y) = eSupp(X \rightarrow Y)/\ gSupp(X \rightarrow Y) \end{gathered} \quad (4)$$

Table 5 shows some example of FASPe rules discovered in the Malay Text Corpus with min_sup value ≥ 2 and min_conf value ≥ 0.4 or 40%. For example the term "mengulas" has the gSupp value of 13, indicating the term occurs in 13 sentences of news source articles. The eSupp value is 11 indicating that out of 13 times it occurs, it was removed by experts 11 times giving the confidence value or Conf of 0.84 or 84%. If the next sequence found is "lanjut", the Conf value calculated is 0.85 or 85%. Another example is the term "dalam" was eliminated 152 times upon 325 occurrences, if the term "dalam" and "pada" occurs together there is 75% confidence level indicating that the phrase "dalam pada" will be eliminated. Next, if the term "itu" follows it, the sequential rules can be written as "dalam pada" -> "dalam pada itu" has a 65% elimination confidence value.

Table 5 Sample of FASPe rules discovered in Malay Text Corpus

#	FASPe	eSupp	gSupp	Conf
1	mengulas	11	13	0.84
2	mengulas -> lanjut	6	7	**0.85**
3	sehubungan	5	6	0.83
4	sehubungan -> itu	5	6	0.83
5	beliau	52	88	0.59
6	beliau -> berkata	21	28	**0.75**
7	berkata	120	206	0.58
8	dalam	152	325	0.46
9	dalam -> pada	15	20	0.75
10	dalam pada -> itu	13	20	0.65

We also perform some Morphological POS analysis on the FASPe that was discovered in this study. Figure 4 shows that 60% of FASPe from the Malay Text Corpus consist of KT (Function Word), 18% is KN (Noun), 12% is KK (Verb) and KA (Adjectives) is 10%. This result is tally with the description given by Nik Safiah Karim, Farid M Onn, and Musa [35]; where the Function word class only acts as a complement to a phrase, sentence or clause as a Preposition (Kata Sendi) or Conjunction (Kata Hubung).

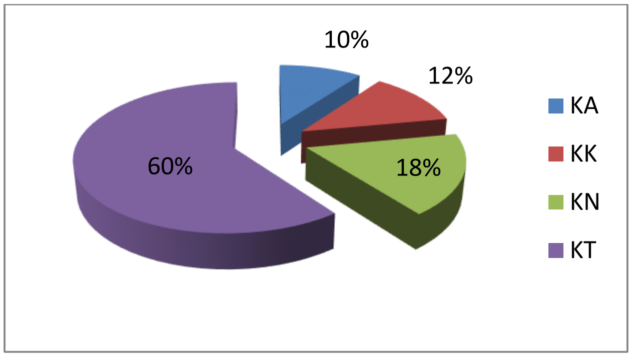


Figure 4 Morphological POS Analysis on FASPe

Thus, by eliminating terms that belongs to this class has insignificant effect to the rest of the summary sentences.

Table 6 FASPe POS Class

FASPe	POS class
mengulas -> lanjut	KK -> KT
beliau -> berkata	KN -> KT

Referring back to this first example in Section 1; where the summary sentence; "~~Mengulas lanjut, beliau berkata~~, waris akan sentiasa dimaklumkan mengenai perkembangan terkini kes tersebut." The derivation from the FASPe POS class is given in Table 6, where it has demonstrated that the *Compressed Summary Sentence* that eliminates the discovered FASPe is still informative without sacrificing the grammatical roles.

5.0 CONCLUSION

In this study we have presented the work in Sentence Compression area by analyzing the pattern in human summarizer using Malay Text Corpus. Some heuristic linguistic rules has been discovered using our extended Pattern-Growth technique or FASPe with resulting high confidence value; indicating that the knowledge can be useful and practical yet simple to be applied to different domain. Next, we would like to continue to apply this heuristic knowledge to train our Malay Text Summarization System to further investigate the effectiveness of these findings.

Acknowledgement

This work is supported by Universiti Sains Malaysia (USM), Research University Grant (RU) by project number 1001/PKOMP/811295. We also like thank our panels of Malay language experts in providing the summary extract.

References

[1] Das, D. and A. F. T. Martins. 2007. A Survey on Automatic Text Summarization. *Literature Survey for the Language and Statistics II Course at CMU 4*. 192-195.

[2] Hahn, U. and I. Mani. 2000. The Challenges of Automatic Summarization. *Computer*. 33(11): 29-36.

[3] Lloret, E. and M. Palomar. 2012. Text Summarisation in Progress: A Literature Review. *Artificial Intelligence Review*. 37(1): 1-41.

[4] Nenkova, A. and K. McKeown. 2011. Automatic Summarization. *Foundations and Trends® in Information Retrieval*. 5(2-3): 103-233.

[5] Saggion, H. and T. Poibeau. 2013. *Automatic Text Summarization: Past, Present and Future*. Springer Berlin Heidelberg.

[6] Jing, H. 2000. Sentence Reduction For Automatic Text Summarization. *Proceedings of the Sixth Conference on Applied Natural Language Processing*. 310-315.

[7] Perera, P. and L. Kosseim. 2014. Evaluation of Sentence Compression Techniques against Human Performance. In *Computational Linguistics and Intelligent Text Processing*. 553-565.

[8] Conroy, J. M., *et al*. 2006. Back to Basics: CLASSY 2006. *Proceedings of DUC*. 150.

[9] Jing, H. and K. R. McKeown. 1999. The Decomposition of Human-Written Summary Sentences. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 129-136.

[10] Zajic, D., *et al*. 2007. Multi-Candidate Reduction: Sentence Compression as a Tool for Document Summarization Tasks. *Information Processing & Management*. 43(6): 1549-1570.

[11] Galley, M. and K. McKeown. 2007. Lexicalized Markov Grammars for Sentence Compression. *HLT-NAACL*. 180-187.

[12] Knight, K. and D. Marcu. 2000. Statistics-based Summarization-Step One: Sentence Compression. *AAAI/IAAI*. 703-710.

[13] Knight, K. and D. Marcu. 2002. Summarization Beyond Sentence Extraction: A Probabilistic Approach To Sentence Compression. *Artificial Intelligence*. 139(1): 91-107.

[14] Nguyen, L. M., *et al*. 2007. A New Sentence Reduction Technique Based on a Decision Tree Model. *International Journal on Artificial Intelligence Tools*. 16(01): 129-137.

[15] Turner, J. and E. Charniak. 2005. Supervised And Unsupervised Learning For Sentence Compression. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 290-297.

[16] Clarke, J. and M. Lapata. 2008. Global Inference for Sentence Compression: An Integer Linear Programming Approach. *Journal of Artificial Intelligence Research*. 399-429.

[17] Cohn, T. and M. Lapata. 2008. Sentence Compression Beyond Word Deletion. *Proceedings of the 22nd International Conference on Computational Linguistics*. 137-144.

[18] Boudin, F. and E. Morin. 2013. Keyphrase Extraction for N-Best Reranking in Multi-Sentence Compression. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

[19] Filippova, K. 2010. Multi-Sentence Compression: Finding Shortest Paths in Word Graphs. *Proceedings of the 23rd International Conference on Computational Linguistics*. 322-330.

[

[20] Filippova, K. and M. Strube. 2008. Dependency Tree Based Sentence Compression. *Proceedings of the Fifth International Natural Language Generation Conference*. 25-32.

[21] Gupta, M. and J. Han. 2011. *Applications of Pattern Discovery Using Sequential Data Mining*. IGI Global.

[22] Li, Y., S. M. Chung, and J. D. Holt. 2008. Text Document Clustering Based on Frequent Word Meaning Sequences. *Data & Knowledge Engineering*. 64(1): 381-404.

[23] Ning, Z., L. Yuefeng, and W. Sheng-Tang. 2012. Effective Pattern Discovery for Text Mining. *Knowledge and Data Engineering, IEEE Transactions*. 24(1): 30-44.

[24] Lin, C.-Y. 2004. Rouge: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. 74-81.

[25] Lin, C.-Y. 2003. Improving Summarization Performance by Sentence Compression: A Pilot Study. *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages*. Association for Computational Linguistics. 11: 1-8.

[26] Agrawal, R. and R. Srikant. 1995. Mining Sequential Patterns. *11th International Conference on Data Engineering (ICDE'95)*. Taipei, Taiwan.

[27] Mabroukeh, N. and C. I. Ezeife. 2010. A Taxonomy of Sequential Pattern Mining Algorithms. *ACM Computing Surveys (CSUR)*. 43(1): 1-41.

[28] Srikant, R. and R. Agrawal. 1996. Mining Sequential Patterns: Generalizations and Performance Improvements. *Proceedings of the Fifth International Conference on Extending Database Technology*. Avignon, France.

[29] Zaki, M. J. 2001. SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning Journal*. 42(1): 31-60.

[30] Han, J., *et al*. 2007. Frequent Pattern Mining: Current Status and Future Directions. *Data Mining and Knowledge Discovery*. 15(1): 55-86.

[31] Mooney, C. H. and J. F. Roddick. 2013. Sequential Pattern Mining – Approaches and Algorithms. *ACM Computing Surveys*. 45(2): 1-39.

[32] Pei, J., *et al*. 2004. Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. *Knowledge and Data Engineering, IEEE Transactions*. 16(11): 1424-1440.

[33] Han, J., *et al*. 2000. FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 355-359.

[34] Han, J., *et al*. 2004. Mining Frequent Patterns Without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery*. 8(1): 53-87.

[35] Nik Safiah Karim, Farid M Onn, and H. H. Musa. 2008. *Tatabahasa Dewan Edisi Ketiga*. Kuala Lumpur: Dewan Bahasa dan Pustaka.

[36] Vadivel, A. and S.G. Shaila. 2014. Event Pattern Analysis and Prediction at Sentence Level using Neuro-Fuzzy Model for Crime Event Detection. *Pattern Analysis and Applications*. 1-20.

[37] Kupiec, J., J. Pedersen, and F. Chen. 1995. A Trainable Document Summarizer. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Seattle, Washington, USA. 68-73.