DESCRIPTION OF THE MIKE\textsubscript{2} ALGORITHM FOR PRESENTATION MINING

Vinothini Kasinathan\textsuperscript{1} and Aida Mustapha\textsuperscript{1,2}
\textsuperscript{1}Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, UPM Serdang, Selangor Darul Ehsan, Malaysia
\textsuperscript{2}Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Malaysia
E-Mail: vinothini@apu.edu.my

ABSTRACT
This paper describes a keyphrase extraction algorithm in Presentation Mining called MiKe\textsubscript{2}. The algorithm extracts keyphrases and keywords from a collection of presentation slides to be generated into a visual knowledge display looks like a mind map. MiKe\textsubscript{2} takes a statistical approach by combining the n-grams frequency count and weight from the C-Value approach. The algorithm is hoped to improve performance in Presentation Mining by automatically generating a high quality mind map that could improve teaching and learning in general.

Keywords: presentation mining, keyphrase, mining keyphrase.

INTRODUCTION
Slide presentationssuch as the PowerPoint are usually prepared by the subjectmatter expert or packaged as a book companion in a linear sequence (Kinchin \textit{et al.}, 2008). The integrated knowledge structure of the subject matter expert is transformed into presentation slides, which are in linear sequence. Nonetheless, post-presentation, the knowledge is actually reconstructed differently by the learners depending on their understanding in Figure-1. Due to this, Kinchin (2009) proposed a concept mapping to help learners visualize the content hence shifting the focus from linear structure to network of expert knowledge.

Presentation Mining (Kasinathan and Mustapha, 2015) is an approach to reduce the misinterpretation between the original expert structure (original contents of the slides) by the instructor and the different audience (learners). The approach is to automatically generate a visual knowledge display such as mind map based on important keywords and key phrases extracted from the presentation slides. The objective of this paper is to introduce a new keyphrases extraction algorithm called MiKe\textsubscript{2} that will further improve the quality of mind maps produced.

The remainder of this paper is organized as follows. Related work presents the works related to keyphrase extraction algorithms. Description of MiKe\textsubscript{2} Algorithm describes the presented MiKe\textsubscript{2} algorithm followed by its application to Presentation Mining. Finally, concludes with some direction for future works.

RELATED WORK
Keyphrases are sequence of words that provide a brief abstraction on the document content (Witten \textit{et al.}, 1999), (El-Beltagy, 2006), (Kumar \textit{et al.}, 2008). The number of words in a phrase usually ranges from one to three, while the sequence of four words in a phrase is usually rare, unless in a specific domain such as medicine. Research has shown that manually assigning keyphrases are costly and time consuming due to the demand of domain specialist, which are individuals who have to read through all the content inorder to look up for the keyphrases (Lim \textit{et al.}, 2013). While the task has now become nearly impossible to achieve due to the present situation of document overloading, automatic keyphrase extraction systems are highly required. Automatic keyphrase extraction is far more cost effective, and the process of identifying key phrases found within a document that are most likely to be assigned by a human (El-Beltagy, 2006).

Implementation of automatic keyphrase extraction can be broadly categorized into two approaches; learning and non-learning (Kumar \textit{et al.}, 2008). Among well-known keyphrase extraction systems are GenEx (Turney, 2000), KEA (Witten \textit{et al.}, 1999) and KP-Miner (El-Beltagy, 2006). GenEx and KEA both treated the task of extracting keyphrases as a supervised learning approach where training documents with known keyphrases are trained inorder to build a model for identifying the probabilities of identified candidate keyphrases to be a keyphrase. KP-Miner, on the other hand, uses a non-learning approach, whereby no training documents are required in order to identify keyphrases within a given document (Lim \textit{et al.}, 2013).

The main difference among the extraction algorithms lies in the calculation of weightage for each
candidate phrases produced. GenEx uses the TF, position of first occurrence, and number of words in a phrase. KEA uses TF-IDF and position of first occurrence. Meanwhile, KP-Miner uses TF-IDF, position of first occurrence, and two boosting factors which boost the weight of longer phrases, as well as phrases which occur earlier. KP-Miner also proposed an N-gram filtration approach that uses TF and position of first occurrence. KP-Miner outperformed GenEx and KEA in terms of accuracy rate attributed by the N-gram filtration approach as well as the processing time since no learning is involved.

DESCRIPTION OF MiKe<sub>2</sub> ALGORITHM

MiKe<sub>2</sub> keyphrase extraction algorithm takes a statistical approach to identify the most accurate and meaningful keyphrases based on the C-value (Frantzi and et al., 2000) for the highest n-gram generated for each candidate phrase. N-gram is very useful in keyphrase extractions because some words or terms are more probable to follow a word in certain contexts, hence forming a phrase of certain number of words. However, previous research has shown that n-gram is insufficient to differentiate meaningful phrases such as ‘no explicit loop’ vs. ‘explicit loop’. In summary, MiKe<sub>2</sub> is shown in Algorithm 1.

```
Algorithm 1 Presentation Mining
for each input slide (* .pptx) do
    Perform Pre-Processing
    Select candidate phrases
for everypcandidate phrase do
    generate n-grams
    calculate c-value
    weigh candidate phrases
end for
end for
Generate visual knowledge display
```

From the algorithm, pre-processing in MiKe<sub>2</sub> involves standardization, sentence segmentation, tokenization, lemmatization, part-of-speech tagging, words removal, phrase recognition and chunking. During standardization, the collection of input slides will be tranformed into ASCII-English. Single and double quotations as well as hyphens are converted into a readable form. Newlines are replaced with tab and whitespaces are trimmed. Next, in sentence segmentation, the sentences are split into newline, period, exclamation marks and question marks using API. During tokenization, digits and letters in on-alphanumeric characters are separated, hyphens are joined with words, whitespaces and continuous symbol are removed.

MiKe<sub>2</sub> uses API to lemmatize tokens into their base forms and refine the results by taking characters after the plus symbol. After lemmatization, API part-of-speech tagging is performed and the results are refined by taking the characters after the underscore symbol. During words removal, symbols, words less than 4 characters (except those capitalized and tagged with ‘CD’ during POS-tagging), as well as stop words are removed. Finally, during phrase recognition and chunking, full sentences are sent to perform API chunking and the chunked tags are modified to cross marked tokens to “0”. Based on the chunk tags, tokens are also joined to form a phrase.

Once pre-processing is completed, words are gathered from all slides to form a list of distinct words. Finally, candidate phrases are selected based on the three conditions; the candidate phrase cannot be a substring of duplicated, it has to be a noun, and it must not contain words with ‘CD’ POS-tag. MiKe<sub>2</sub> then uses the candidate’s phrases to generate the n-grams as shown in Algorithm 2.

```
Algorithm 2 Generate N-Grams
min n = (word_count = 1) ? 1 : 2
max n = (word_count >= 3) ? 3 : word_count
for each n in min, max do
    for each word in phrase do
        int gram_count = n
        int pick_index = word_index
    end for
    while gram_count > 0 do
        add picked word to phrase
        go to next word in gram
    end while
end for
```

Application to presentation mining

Presentation mining in an approach to extract keyphrases from presentationslides such as PowerPoint
and generate a visual knowledge display using the extracted keywords. Figure-2 shows the steps in Presentation Mining which is undertaken in this research. To illustrate the application to MiKe2 in a Presentation Mining approach, a collection of presentation slides for Artificial Intelligence course at introductory level across different universities worldwide are used as the input slides. The scope will be using the Artificial Intelligence: Modern Approach text book which is written in American English which will be the lexicon used.

After the process of selecting slides and keyphrases, the system will visualize the selected keyphrases into a SmartArt diagram in Microsoft PowerPoint, which is similar to mind map.

Output generated mind map using the PowerPoint SmartArt is shown in Figure-3.

The list of keyphrases extracted by MiKe2 will then be compared with the output from KP-Miner to see the difference. Table-1 shows the comparison of keyphrases extracted by the KP-Miner and the proposed MiKe2 algorithm on Chapter 13 of the Artificial Intelligence: A Modern Approach textbook (Russel and Norvig, 2003). This Table shows that the MiKe2 algorithm overcomes some issues of KP-Miner as an on-learning algorithm by being able to extract more meaningful keyphrases. MiKe2 has used the strength of C-value algorithm that brings up keyphrase ranking to the top after re-ranking it. It also has the strengths of N-gram which avoids the bias extraction process in which KP-miner uses. Therefore the outputs from MiKe2 are much more meaningful then KP-Miner algorithm.

<table>
<thead>
<tr>
<th>Slide</th>
<th>KP-miner</th>
<th>MIKE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Planning and Acting</td>
<td>Planning and Acting</td>
</tr>
<tr>
<td>5</td>
<td>Solutions</td>
<td>Solutions</td>
</tr>
<tr>
<td></td>
<td>Conditional planning</td>
<td>Observation actions</td>
</tr>
<tr>
<td></td>
<td>Assume normal states</td>
<td>Failure</td>
</tr>
<tr>
<td></td>
<td>Observation actions</td>
<td>Conditional planning</td>
</tr>
<tr>
<td></td>
<td>Check progress</td>
<td>Unanticipated outcomes</td>
</tr>
<tr>
<td>4</td>
<td>Things go wrong</td>
<td>Things go wrong</td>
</tr>
<tr>
<td></td>
<td>Incomplete information</td>
<td>Incorrect information</td>
</tr>
<tr>
<td></td>
<td>Unknow preconditions</td>
<td>Unknow preconditions</td>
</tr>
</tbody>
</table>

Figure-3. Output of the powerpoint SmartArt.

Figure-2. Steps in presentation mining.
### CONCLUSIONS

Slide presentations have been widely used in current teaching and learning process. While text-laden slides might give a comprehensive feel over the materials, the slides full of key points are not useful without the presenter (Kasinathan et al., 2013). The objective of Presentation Mining is to improve the teaching and learning process by transforming the slide contents into a visual knowledge display because the main challenge lies in the fact that slides already contains keywords and keyphrases. Visual knowledge display such as the mind map reorganizes the keywords/keyphrases in the slides from sequential to network-based while keeping the relationships from the slides intact.

This paper presents a new keyphrase extraction algorithm called MiKe² that capitalized on the statistical information in the words. MiKe² was applied to Presentation Mining and its outputs are compared with the output of KP-Miner. Based on the comparisons, MiKe² was at par to KP-Miner with more meaningful keyphrases such as ‘conditional step’ as opposed to ‘insert conditional step’ by KP-Miner. In the future, this research will strive to improve the keyphrase extraction algorithm in Presentation Mining approach by considering contextual knowledge within the slides.

### ACKNOWLEDGEMENT

This project is sponsored by University Putra Malaysia under Geran Putra-Inisiatif Putra Siswazah (GP/IPS/2013/9399844).

### REFERENCES


Kumar, N., and Srinathan, K. 2008, September. Automatic keyphrase extraction from scientific documents using N-

---

<table>
<thead>
<tr>
<th>Disjunctive effects</th>
<th>Required preconditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorrect information</td>
<td>Incorrect postconditions</td>
</tr>
<tr>
<td>Qualification problem</td>
<td>Current state</td>
</tr>
<tr>
<td>6</td>
<td>Conditional planning</td>
</tr>
<tr>
<td>Insert conditional step</td>
<td>Observation outcome</td>
</tr>
<tr>
<td>Complete plan</td>
<td>Conditional step</td>
</tr>
<tr>
<td>Observation outcome</td>
<td>Current KB</td>
</tr>
<tr>
<td>13</td>
<td>Monitoring</td>
</tr>
<tr>
<td>Executive monitoring</td>
<td>Action monitoring</td>
</tr>
<tr>
<td>Action monitoring</td>
<td>Execution monitoring</td>
</tr>
<tr>
<td>Preconditions of remaining plan</td>
<td>remaining plan</td>
</tr>
<tr>
<td>Robot bump sensor</td>
<td>Robot bump sensor</td>
</tr>
<tr>
<td>2</td>
<td>Outline</td>
</tr>
<tr>
<td>Real world</td>
<td>Real world</td>
</tr>
<tr>
<td>15</td>
<td>Re-planning</td>
</tr>
<tr>
<td>No explicit loop</td>
<td>Explicit loop</td>
</tr>
<tr>
<td>Simplest</td>
<td>Best continuation</td>
</tr>
<tr>
<td>Scratch scratch</td>
<td></td>
</tr>
</tbody>
</table>

### OUTLINE

- Real world
- No explicit loop
- Simplest
- Scratch scratch
gram filtration technique. In Proceedings of the eighth ACM symposium on Document engineering (pp. 199-208). ACM.


