Empirical Comparison of Concept Indexing and Latent Semantic Indexing on the Content Analysis of Filipino Essays

Darrel Alvin N. Ong¹, Abigail R. Razon¹, Rowena Cristina L. Guevara², Prospero C. Naval, Jr.¹
¹Department of Computer Science
²Electrical and Electronics Engineering Institute
University of the Philippines, Diliman, Quezon City, PH
pcnaval@dcs.upd.edu.ph

ABSTRACT
This paper proposes the first Automated Essay Grader (AEG) for the Filipino language with results that are competitive with human checkers. In this study, the authors focused on content evaluation using Concept Indexing (CI) and Latent Semantic Indexing (LSI). The two NLP algorithms were compared based on their accuracy and speed. The effects of spell checking, stop word removal, stemming, sub-clustering and normalization weighting schemes were also examined.

Categories and Subject Descriptors
H.3.1 [Information Storage and Retrieval]: Linguistic Processing

General Terms
Automatic Essay Grading, Latent Semantic Analysis, Concept Indexing

1. INTRODUCTION
The Filipino language is the national language of the Philippines whose base language is Tagalog which is an Austronesian language spoken by one-third of the country’s population as their first language [8][12]. The Filipino language is used as the official medium of communication and is required by law as the language of instruction in educational institutions [12]. This study addresses the need for an automatic means of essay evaluation and scoring of essays written in Filipino.

Currently, Filipino language teachers manually check each essay based on its content, grammar, and organization. The checker examines if the body of the essay appropriately and completely elaborates and discusses a given topic, taking into account the smoothness of transition of ideas from paragraph to paragraph. The teacher also needs to check the spelling and grammar of each sentence. Essay checking requires meticulous analysis and demands a lot of time from the checker. This discourages teachers from giving out more writing activities to their students as they do not want to spend most of their time in essay checking.

Statistics from the Department of Education about the current status of public high schools show that average teacher-student ratio is about 1:40 per class [4]. Teachers may handle as many as seven sections or up to 280 students [7]. A teacher needs at least 45 hours given he can check an essay in ten minutes. Manually checking a large number of essays does not only take too much time, but is also vulnerable to human errors and subjectivity. A teacher may experience fatigue, and boredom which may affect the evaluation of essays resulting in inconsistent scoring.

These problems motivate the development of Automated Essay Graders (AEG). Various AEG systems have been implemented for different languages. Some of these are English, Chinese, and Japanese. However, to the authors’ knowledge, there is none for the Filipino language [5, 6, 11, 13]. In this study, the authors developed the first AEG system for the Filipino language with a performance comparison of CI and LSI for this language. The authors focused only on the analysis of the essay content.

The rest of the paper is organized as follows: Section 2 discusses the existing tools in the Filipino for NLP. Section 3 describes the important steps in the implementation of the two algorithms while the next section presents the results and analysis. Section 5 gives the conclusion and recommendations.

1.1 Existing NLP Tools for the Filipino Language
The following describe current approaches in correcting misspelled Filipino words and deriving their root words. Spell checking and stemming for the Filipino language are still under active research. The spelling checker called SpellChef detects misspelled Filipino words and generates a list of word suggestions [2]. It uses a combination of dictionary-
lookup, n-gram analysis, soundex and character distance measurements, utilizing a database containing a total of 45,548 words. According to its authors, the system has an error rate of 7% such that 7% of the words marked as incorrect are actually correctly spelled. However, they only used three documents namely a student’s essay, an entertainment article, and a sports article [2]. These can be considered as a very small corpora. More experiments are needed to test the system on other documents under different domains. The stemming algorithm called TagSA derives the root word of most forms of Tagalog words, which included Tagalog affixation, reduplication and compounding [1]. The system was based on a dictionary that contains 1575 root words and was manually derived from three test samples, namely, the Philippine Constitution, the Babilonia Wilner Foundation-Balikas website and a dissertation [1]. The authors claimed that the system achieved less than 15% and 0.005% understemming and overstemming errors, respectively. However, they tested their system only on 6,382 words including duplications. Also since the system is dictionary-based, the system requires a comprehensive list of Tagalog words.

2. METHODOLOGY
The following subsections briefly discuss the details of CI and LSI. Both algorithms use the same pre-processing, occurrence matrix construction, normalization weighting scheme, similarity calculation and scoring steps except for sections 2.5 and 2.6 which discuss the CI’s and LSI’s respective implementations in constructing the reduced dimensional space and folding-in.

2.1 Data Collection
The dataset consists of three sets of essays on different topics written by students from Parañaque National High School (PNHS). The topics are: (1) Mga Paraan na Maaari mong Gawin upang Makatulong sa ating Bansa bilang Mag-aaral, (2) Positibong Pananaw ng Kahirapan, and (3) Kabayanihan.

### Table 1: Dataset

<table>
<thead>
<tr>
<th>Set</th>
<th>NumEssay</th>
<th>WordPerEssay</th>
<th>YrLvl</th>
<th>NumSec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>600</td>
<td>750</td>
<td>1st to 4th</td>
<td>all</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>150</td>
<td>3rd</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>150</td>
<td>1st</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 2: Filipino Language Teachers

<table>
<thead>
<tr>
<th>Filipino Language Teacher</th>
<th>Years of Teaching Exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>more than 30</td>
</tr>
<tr>
<td>B</td>
<td>28</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1 shows the number of essays per set, the approximate number of words per essay, the year level of the students who wrote the essays, and the number of sections from which the essays were gathered. Table 2 shows the years of teaching experience for each Filipino language teacher.

All of the essays were manually typed and printed before they were given to Filipino language teachers for scoring. Teacher A checked all three sets. Teacher B, C, and D only checked 29 of 600 essays for the 1st set, and all essays for 2nd and 3rd. The Filipino language teachers checked each essay based on content, grammar, and organization. Only the scores for content are used in this study. These were normalized. The essays were then pre-classified according to a five-point grading scale.

2.2 Pre-Processing
The collected data was randomly divided into training and test sets. For each score category, 60% of the essays were randomly chosen for the training set. The remaining essays went to the test set. A ten-fold cross validation scheme was applied. For each topic and teacher, the training and test sets were randomly created ten times. Each test is repeated ten (10) times for CI to take into account the stochastic nature of sub-clustering.

The sentences for each document were converted into tokens or words. Only words found at the start of the sentence were lowercased. The spelling checker, stop word removal and stemming may be applied for all words that are in lowercase. The spelling checker corrects misspelled Filipino words. Stop words are common words that may improve the system’s performance when taken away from the vocabulary. Stemming takes the root form of a word.

The spelling checker algorithm uses a variant of the American soundex in which a code is determined for each word in the dictionary. Words with the same code are grouped under that code. Then, the algorithm determines the code of any input word to find the code group where it should belong. The similarity between the input word and words in the code group are calculated using the Levenstein Edit Distance algorithm [20]. Finally, the word having the least distance is returned.

The stemming algorithm creates a list of words for each rule that is satisfied. Consider the word “pinagkainan”. A rule will identify the suffix “-an” pushing “pinagkain” and “pinakain” into the candidate list. Another rule will identify the prefix “pinag-” for both words in the candidate list pushing new words to the list. The candidate list now has “pinagkainan”, “pinakain”, “kainan”, “kain”. These words are compared to the dictionary containing only root words and the longest word is returned. In this example, the first three words won’t be found in the dictionary. Hence, the returned word is “kain”.

2.3 Occurrence Matrix and Normalization
The occurrence matrix is a matrix generated from the documents. The rows and columns correspond to the vocabulary words and documents respectively. The values for each cell are the counts of the words found in each document.

Each occurrence matrix was normalized using either normalized raw-term frequency (raw tf) or normalized term frequency-inverse document frequency (tf-idf) weighting schemes given by the formulas below. The former normalizes each document vector using the weights of the actual frequency of each word. The latter uses weights, which are
a function of the word frequency, number of essays that contains the word, and total number of essays. A word is given a low weight if it is likely to be present in the other documents. The following are their formulas:

Normalized raw-term frequency weighting [11]:

\[ x_{t,d} = \frac{tf_{t,d}}{\sqrt{\sum q_{t,d}^2}} \]

Normalized tf-idf weighting [3]:

\[ w_{t,d} = (1 + \log_{10}tf_{t,d})\log_{10}\left(\frac{N}{df_t}\right) \quad x_{t,d} = \frac{w_{t,d}}{\sqrt{\sum w_{t,d}^2}} \]

where \( x_{t,d} \) is the normalized value for term \( t \) in document \( d \) and quantity \( tf_{t,d} \) is the frequency of term \( t \) in document \( d \); \( N \) is the number of train essays and total number of essays when used for training and test sets respectively, \( df_t \) is the number of essays that contain term \( t \).

### 2.4 The Concept Indexing Algorithm

#### 2.4.1 Sub-Clustering and Concept Matrix:

After normalizing the occurrence matrix, each score category was sub-clustered using either K-Means [10] or Fuzzy C-Means [9]. These are unsupervised clustering algorithms that group similar documents given a pre-defined number of clusters. The centroids for each group of similar documents are calculated and become the concept vectors. The concept matrix \( (C_p) \) is a matrix whose columns are the centroids of each subcluster of documents with similar meanings.

#### 2.4.2 Concept Decomposition and Folding-In:

The concept matrix is used to project the normalized occurrence matrix to the reduced dimensional space. The closed-form formulas are shown below [11]:

\[ X^* = (C_p^T C_p)^{-1} C_p^T X \]

\[ Q^* = (C_p^T C_p)^{-1} C_p^T Q \]

The \( X \) and \( Q \) matrices were the normalized occurrence matrices of the training and test sets respectively. On the other hand, \( X^* \) and \( Q^* \) matrices correspond to the projected matrices of the training and test set.

### 2.5 The Latent Semantic Indexing Algorithm

#### 2.5.1 Singular Value Decomposition:

The Singular Value Decomposition (SVD) is computed for the normalized occurrence matrix of the training set given by this formula [11]:

\[ X = UDV^T \]

where \( D \) is a matrix whose diagonals are the singular values of \( X \) which is the normalized occurrence matrix of the training set. \( U \) and \( V \) are matrices whose columns contain the left and right singular vectors of \( X \) respectively.

#### 2.5.2 Dimensionality Reduction and Folding-In:

This step computes for the rank \( k \) approximation of \( X \) by taking the top \( k \) singular values from the \( D \) matrix. The training and test sets are then projected to the reduced dimensional space taken from the previous step. These approximations are

\[ X^* = X^T U_k D_k^{-1} \]

\[ Q^* = Q^T U_k D_k^{-1} \]

The \( X^* \) and \( Q^* \) matrices are the projected \( X \) (train) and \( Q \) (test) matrices respectively. \( U_k \) is the reduced \( U \) matrix while \( D_k \) is the reduced \( D \) matrix.

### 2.6 Similarity Values and Scoring

Cosine similarity was used to compute for similarity between training and test documents. It is the dot product of the two documents divided by the corresponding vector lengths. The similarity values for each test document were ranked. The score for the test essay is the same as the score of the training essay that has the highest cosine similarity value. The similarity between a query document \( q_i \) with another document \( d_i \) is given by this formula [11]:

\[ sim(q_i, x_i) = \frac{q_i \cdot x_i}{|q_i||x_i|} \]

### 2.7 Performance Metrics

The scores given by the system were compared to the scores provided by the human checker and the Exact Agreement Accuracy, Adjacent Agreement Accuracy and Overall Accuracy (OA) were computed as follows:

**Exact Agreement Accuracy (EAA)**

\[ EAA = \frac{\text{num_human_equal_aeg}}{\text{Total number of test essays}} \]

where num_human_equal_aeg is the number of test essays with Human score equal to AEG score.

**Adjacent Agreement Accuracy (AAA)**

The Adjacent Agreement Accuracy is defined as the ratio of the number of test essays with a human score equal to AEG score \( \pm 1 \) over the total number of test essays and is given by the formula below [11]:

\[ AAA = \frac{\text{num_human_equal_aeg} \pm 1}{\text{Total number of test essays}} \]

**Overall Accuracy (OA)**

The OA is the sum of EAA and Adjacent Agreement Accuracy (AAA) that is defined as the ratio of the number of test essays with a human score equal to AEG score \( \pm 1 \) over the total number of test essays [11]:

\[ OA = EAA + AAA \]

In comparing two independent experiments to determine which experiment performs better, the unpaired t-test was used. It is a statistical test used to determine if the mean of the two samples are statistically different by computing the p-value. A p-value of less than 0.05 means that there is evidence to reject the null hypothesis which implies that difference of the two means are statistically significant within a 95% confidence interval.

### 3. RESULTS

The system was implemented using R statistical software (Version x64 2.12.2), which is free software [17]. The “e1071” package was installed for Fuzzy C-Means clustering. The experiments were conducted using an Apple laptop with 1.86 GHz Intel Core 2 Duo and 2GB RAM.
Experiments were conducted to examine the effects of the spell checking, stop word removal, stemming and normalization weighting scheme. Spell checking and stemming do not show significant impact the performance of the system.

For stop word removal and normalization weighting, the raw tf give better performance than tf-idf when stop word removal is applied. This is because there are common words present in essays such as “ang”, “sa”, “mga”, etc. that occur frequently but are not content words. These words take large weights in the document vectors which degrades the performance of the system.

Table 3 shows a comparison between two experiments involving stop word removal. The other parameters used in these experiments are no spell checking, no stemming, raw tf normalization weighting scheme, and 60-40 training-test ratio. The p-value is calculated using t-test to determine if the difference between the two EAA values is significant. A value less than 0.05 means that the difference is significant at 95% confidence interval. The table shows that the p-values are quite significant. Thus, removing stop words improves the performance of the system.

Table 3: Stop Words Removal Performance

<table>
<thead>
<tr>
<th>Method</th>
<th>with Stop Words</th>
<th>w/o Stop Words</th>
<th>p</th>
<th>Sig?</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP</td>
<td>EAA</td>
<td>EAA</td>
<td>0.07</td>
<td>Quite</td>
</tr>
<tr>
<td>CI,cmeans</td>
<td>0.485</td>
<td>0.543</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI,kmeans</td>
<td>0.456</td>
<td>0.547</td>
<td>0.01</td>
<td>Yes</td>
</tr>
<tr>
<td>LSI</td>
<td>0.529</td>
<td>0.612</td>
<td>0.05</td>
<td>Quite</td>
</tr>
</tbody>
</table>

Table 4: Human - Human Overall Accuracy

<table>
<thead>
<tr>
<th>Teacher</th>
<th>Teacher</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>0.733</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>0.548</td>
</tr>
<tr>
<td>A</td>
<td>D</td>
<td>0.584</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>0.646</td>
</tr>
<tr>
<td>B</td>
<td>D</td>
<td>0.671</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>0.656</td>
</tr>
</tbody>
</table>

Each set of essays were checked by four different teachers. These scores were compared to the other teachers and their OA are computed and shown in Table 4. It shows that the OA among human checkers ranges from 0.548 to 0.733.

Experiments are also conducted to test system on different teachers. Tables 5 to 8 are the results when the system’s performance is compared to the human checkers. The tables show that the system can perform the same or even better than the other human checkers.

The performance of CI-Cmeans, CI-Kmeans, and LSI are also analyzed. Tables 9 to 12 show the comparison among the three algorithms for all four teachers. The p-values are calculated to determine if the difference in OA is significant. Results show that there is no significant difference in the performance between Fuzzy C-Means and K-Means sub-clustering for CI. Between CI and LSI, results show that CI performs better than LSI only for teacher C. For all other teachers, there is no evidence that one outperforms the other. This implies that CI may perform better than LSI for some teachers.

The above experiments show the performance of Fuzzy C-Means and K-Means sub-clustering for CI. These sub-clustering algorithms are similar such that both create k number of sub-groups for each score category. The algorithms try to minimize the sum of squares of the distance of points from the centroid for each sub-cluster. The main difference of the two is that in Fuzzy C-Means, a point has a degree of belonging to a cluster instead of classifying the point to only one cluster in K-Means. The experiments show no significant difference in the performance between the two clustering algorithms.

The time complexity of CI is $O(ike^2)$ where i is the number of iterations until convergence is achieved, e is the vocabulary count, k is the number of sub-clusters, and n is the number of essays. On the other hand, the time complexity for LSI is $O(en^2)$. In general, the value of k is often small since sub-topics for a given set of essays are few. Thus, CI is faster than LSI. Measurements of computation time for both algorithms (Fig. 1) support this conclusion.

4. CONCLUSION

In this paper, the proposed system is the first AEG system for evaluating and scoring the content of Filipino essays. It learns and adapts to how a teacher evaluates and scores the content of an essay. Human-AEG agreement is comparable with Human-Human agreement and the AEG is not
Table 9: Teacher A

<table>
<thead>
<tr>
<th>Method</th>
<th>CI, cmeans</th>
<th>CI, kmeans</th>
<th>Sig?</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI, cmeans</td>
<td>0.692</td>
<td>0.678</td>
<td>0.69</td>
</tr>
<tr>
<td>CI, kmeans</td>
<td>0.692</td>
<td>0.717</td>
<td>0.23</td>
</tr>
<tr>
<td>LSIF</td>
<td>0.678</td>
<td>0.717</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Table 10: Teacher B

<table>
<thead>
<tr>
<th>Method</th>
<th>CI, cmeans</th>
<th>CI, kmeans</th>
<th>p</th>
<th>Sig?</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI, cmeans</td>
<td>0.831</td>
<td>0.823</td>
<td>0.74</td>
<td>No</td>
</tr>
<tr>
<td>CI, kmeans</td>
<td>0.831</td>
<td>0.804</td>
<td>0.48</td>
<td>No</td>
</tr>
<tr>
<td>LSIF</td>
<td>0.823</td>
<td>0.804</td>
<td>0.25</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 11: Teacher C

<table>
<thead>
<tr>
<th>Method</th>
<th>CI, cmeans</th>
<th>CI, kmeans</th>
<th>p</th>
<th>Sig?</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI, cmeans</td>
<td>0.860</td>
<td>0.859</td>
<td>0.95</td>
<td>No</td>
</tr>
<tr>
<td>CI, kmeans</td>
<td>0.860</td>
<td>0.773</td>
<td>0.00</td>
<td>Yes</td>
</tr>
<tr>
<td>LSIF</td>
<td>0.859</td>
<td>0.773</td>
<td>0.00</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 12: Teacher D

<table>
<thead>
<tr>
<th>Method</th>
<th>CI, cmeans</th>
<th>CI, kmeans</th>
<th>p</th>
<th>Sig?</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI, cmeans</td>
<td>0.726</td>
<td>0.737</td>
<td>0.70</td>
<td>No</td>
</tr>
<tr>
<td>CI, kmeans</td>
<td>0.726</td>
<td>0.686</td>
<td>0.08</td>
<td>Quite</td>
</tr>
<tr>
<td>LSIF</td>
<td>0.737</td>
<td>0.686</td>
<td>0.16</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure 1: CI and LSI speed comparison

The proposed system is not intended to replace the teachers but instead, serve as a tool to aid them in checking the essays. The teachers will still have to check a number of essays that will serve as the training essays for the system. Then, they can use the system to score the remaining essays to be checked.

The proposed system yielded an overall accuracy of 0.686 to 0.860 while the overall accuracy among Filipino language teachers ranges from 0.548 to 0.733. This implies that the system can successfully analyze and score the content of essays in Filipino language.

Application of spell checking and stemming does not show significant improvement in the system. However, raw tf normalization weighting scheme performs better than tf-idf when stop words are removed. Between CI and LSI, the former can perform better and is generally faster than the latter.

5. ACKNOWLEDGMENTS
The authors wish to thank Ms. Jocelyn Buenavista, Ms. Esperanza T. de Guzman, Mr. Dan Agpaoa, and Mr. Alfred Mendoza who patiently evaluated and scored the essays in our corpora.

6. REFERENCES
