Automatic Enemy Item Detection

Using Natural Language Processing

BY

FANG PENG
B.A., Tsinghua University, 2010
M.A., Tsinghua University, 2013

DISSERTATION

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Educational Psychology
in the Graduate College of the
University of Illinois at Chicago, 2020

Chicago, Illinois

Defense Committee:

Everett V. Smith, Chair and Advisor
Kirk A. Becker, Pearson VUE
Ken A. Fujimoto, Loyola University Chicago
Yue Yin
Rebecca M. Teasdale

This dissertation is dedicated to my beloved parents, and to everyone who have supported me throughout my education.

Thank you for walking with me through this journey.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my heartfelt gratitude to my advisor, Prof. Everett Smith, for the continuous support of my doctoral studies and research. Your patience, inspiration, and insight has guided me throughout this research and writing of this dissertation. I can still remember the excitement you shared when we first talked about this research, and your invaluable insight has made it possible for me to transform a loose collection of ideas into a worthy research project. I could not have come this far without your mentoring and guidance.

Besides my advisor, I would like to thank each member of my dissertation committee. Dr. Kirk Becker, you not only showed me the door to this research topic, but also shared your inspiring ideas and suggestions with me generously, which shaped the foundation of this research. Thank you for lending your expertise and knowledge to my research. Prof. Yue Yin, you have always been an amazing supporter of my doctoral study. You are a great teacher and mentor who prepared me with knowledge in psychometrics and encouraged me to think more independently as a young researcher. Prof. Ken Fujimoto, you were one of my first friends and teammates as I started my journey at UIC. I have always looked up to your expertise in psychometrics and enjoyed your great personality throughout my doctoral years. Thank you for continuously sharing your insights and wisdom with me, both in research and in life. Prof. Rebecca Teasdale, thank you for offering your valued expertise, advice, and constructive feedback, which have greatly improved my research design and dissertation writing.

I would like to thank Prof. Rachel Gordon for continuously supporting me and training me in research throughout numerous projects. You taught me what it meant to be a knowledgeable, organized, and meticulous scholar, and I am deeply indebted to you for your coaching and support.

I would also like to thank my colleagues at NCSBN for their support of this research, especially to Dr. Xiao Luo, Dr. Hong Qian, Dr. Doyoung Kim, and Dr. Shu-chuan Kao, for helping me formulate this research project and offering continuous data support. I want to give my sincere thanks to our subject matter experts, Jose, Julie, and Kristin, for their diligence in reviewing the enemy item pairs and their invaluable insights on enemy relationship identification. This research would not have been possible without all of your support.

Finally, I would like to thank my family for your unconditional love and support through everything. Mom, Dad, Ryan, Carrie, David – you are my world.

TABLE OF CONTENTS

CHAPTER PAGES

LIST OF TABLES

TABLE PAGES

LIST OF FIGURES

FIGURE PAGES

LIST OF ABBREVIATIONS

AES	Automated Essay Scoring
ANN	Artificial Neural Network
AUC	Area Under the Curve
CAT	Computerized Adaptive Testing
CBT	Computer-Based Testing
FN	False Negative
FP	False Positive
FPR	False Positive Rate
IRT	Item Response Theory
LOFT	Linear on-the-fly Testing
LSA	Latent Semantic Analysis
LDA	Latent Dirichlet Allocation
MLE	Maximum Likelihood Estimation
MSE	Mean Square Error
NLP	Natural Language Processing
PLSA	Probabilistic Latent Semantic Analysis
ROC	Receiver Operator Characteristic
SME	Subject Matter Expert
SMOTE	Synthetic Minority Over-sampling Technique
SVD	Singular Value Decomposition
TF-IDF	Term Frequency-Inverse Document Frequency

LIST OF ABBREVIATIONS (CONTINUED)

TN	True Negative
TP	True Positive
TPR	True Positive Rate

SUMMARY

This study explores the effectiveness of using Natural Language Processing (NLP) techniques in automatically detecting enemy item pairs within item banks. The overarching goal was to compare and evaluate the performance of various automatic enemy item detection procedures using an operational item pool. To answer the research questions, this study examined the classification results across three conditions: (a) NLP techniques, including Vector Space Model, Latent Semantic Analysis, and Latent Dirichlet Allocation; (b) classification algorithms, including the logistic regression classifier and the Artificial Neural Network classifier, and (c) probability cutoffs ranging from .60 to .90. The classification results were further evaluated by subject matter experts (SMEs), and the models were re-trained using the input from the SMEs.

The findings from this study showed the robustness of the NLP techniques in automatic identification of enemy item pairs. The automatic detection process successfully identified additional enemy relationships previously untagged in the item bank. The classification results from the numerous conditions suggested that the LSA and the VSM models consistently outperformed the LDA models and yielded optimal results at the cutoff of .90. Integrating feedback from SMEs further improved the performance. This iterative process greatly reduced the time and manual labor needed for enemy relationship monitoring and offered flexibility for SME review.

1. INTRODUCTION

Computer-based testing (CBT) is dramatically shaping how we construct and deliver educational assessments and licensure examinations. Compared to traditional paper-and-pencil testing, CBT meets the demand for continuous assessment, and it is able to integrate multiple media and support innovative item formats. Computerized adaptive testing (CAT) also has the capacity to tailor the test to the examinee's level of ability and to increase measurement accuracy of test scores (Van der Linden, 2010).

The demand for more timely and frequent testing requires that testing programs maintain a sufficient item bank in order to (a) ensure the coverage of test contents and difficulty levels, (b) support dynamic test construction and administration process, and (c) increase test security for repeated testing. An operational item pool for large testing programs usually consists of thousands of items, making item bank management an onerous task.

1.1 Enemy Items

One of the crucial aspects of item bank management is monitoring the inter-item dependencies (VeldKamp & Van der Linden, 2010), which includes flagging enemy item sets. Some enemy item sets occur when two or more items share such similar content that they become duplicative in nature (Gibbons et al., 2016). Others are considered enemy items, because one may provide clues about the answer to another (Ackerman & Spray, 1986). It is beneficial to produce and retain such enemy item sets in the item bank, as the variation of items allows test developers to use equivalent yet not identical items across test forms which helps prevent unwanted memorization of items (Woo & Gorham, 2010). However, it is important to ensure

that enemy item sets are not administered together on one test to the same examinee, as this compromises the measurement precision and diminishes the test validity.

Inclusion of enemy items on the same test violates the local independence assumption of Item Response Theory (IRT), which governs most computer-based testing (Lai & Becker, 2010; Muckle & Becker, 2018). The local independence assumption requires that items be conditionally independent of each other; that is, the response to one item should not be contingent on the response to another item after controlling for the underlying trait (Embretson & Reise, 2000). Enemy items introduce this inter-item dependency, as the examinee has an increased probability of answering correctly or incorrectly to enemy sets on the same test. Consequently, using responses to enemy items as independent information for estimating the examinee's ability will bias the resulting score, which undermines the measurement precision.

Enemy items also introduce threats to the test validation process, as enemy items ask almost the same question. The inclusion of such items will place too much weight on one domain of knowledge or competency, thereby reducing the breadth of the test. Sometimes enemy items can affect test validity in a more subtle way. Receiving similar items on a single test may lead the examinee to perceive the items as redundant and, thus, to question the credibility of the test (Kane, 2006). The resemblance of enemy items may cause confusion or distraction during the test, which introduces construct-irrelevant variance to test scores (Downing & Haladyna, 2006).

1.2 Current Practice for Identifying Enemy Items

Typically, the identification of enemy items for testing programs has relied on the manual efforts during item writing, item review, and test form assembly process (Drasgow et al., 2006). Processing the text in an item bank is time-consuming and labor-intensive. Using manual effort to review enemy items is also susceptible to subjectivity and human error. For large-scale testing,

the sheer volume of item banks makes it an unrealistic expectation that human reviewers will catch and flag each enemy item pair correctly. For an item bank consisting of *n* items, the number of item pairs in the bank amounts to $\frac{n(n-1)}{2}$. When the size of the item bank increases, it becomes impossible for human reviewers to compare each item to every other item. In addition, items are being written and added to the item bank periodically, making enemy item monitoring an iterative process.

The manual processes of enemy item review also create challenges for testing programs that have not fully implemented adaptive testing. It precludes any type of dynamic or adaptive testing form construction or administration, such as linear on-the-fly testing (LOFT) or CAT, because enemy item pairs are sometimes identified after forms have been constructed, and additional time is then used to substitute replacement items for enemy sets before form publication or test administration. The lack of enemy item identification in the item bank, coupled with the time-consuming manual process of enemy item review, creates significant limitations to the test construction and administration.

Others statistical approaches were proposed for identifying enemy item pairs. Ackerman and Spray (1986) proposed a general model for detecting violations of local independence by analyzing the shift in item characteristic curves and item parameter estimates. Yen's Q3 statistic (Yen, 1984) can also be used to detect local item dependencies which may indicate an enemy association (Pommerich & Segall, 2008). However, such approaches must be conducted after the test administration. At the time of confirming enemy items, the test forms have been exposed to the examinees. In addition, these post hoc statistical methods are limited to items that have been administered together and are unlikely to fully capture the enemy relationships across the item bank.

Thus, it is desirable to develop an automatic process to create a system that can process the text of an entire item bank and generate probabilistic statistics that indicate item pairs that are likely to be enemy sets. This automatic process can assist human reviewers by providing a list of potential enemy pairs and significantly reducing the time and resources required for manual review.

1.3 Automatic Enemy Item Detection

An item bank is essentially a large corpus consisting of text, phrases, and concepts from test items. Recent advances in the field of Natural Language Processing (NLP) have made it possible to process, analyze, and index large text corpora with the assistance of computer programs. Modern applications of NLP in testing include automated essay scoring (Shermis & Burstein, 2003) and automatic item generation (Gierl & Haladyna, 2013). Some studies have been designed to explore the capability of NLP for test development and item bank maintenance, including item difficulty modeling (Sheehan et al., 2006; Belov & Knezevich, 2008; McLeod et al., 2015), verifying exposed items (Becker & Kao, 2009), assisting subject matter experts (SMEs) in item writing (Becker & Olsen, 2012), identifying reference works of test items (Becker & McLeod, 2013), and enemy item detection (Lai & Becker, 2010; Peng et al., 2018; Weir et al., 2018; Peng et al., 2019).

Studies of automatic enemy item detection using NLP techniques typically involve two stages. In stage one, text-vectorization methods are used to transform a pool of items into mathematical representations, allowing the item text to be easily processed and analyzed systematically. Similarity indices are then computed between each item pairs based on the NLP technique used. In the second stage of the analysis, the similarity indices and other item meta-data are included as input of a classification model to predict the enemy status of each item pair.

The model prediction of enemy membership is compared with the true enemy status in the item pool to evaluate the accuracy of the classification results.

A number of factors determine the accuracy of this automatic detection process. First, various NLP techniques can be used for analyzing the transformed text corpus and calculating similarity indices. Some methods focus on measuring the lexical and syntactic similarities between items (Lai & Becker, 2010), and others on semantic similarities using topic modeling methods (Peng et al., 2018; Weir et al., 2018; Peng et al., 2019; Weir, 2019). Second, the choice of classification algorithm could have an impact on the classification results. Researchers have applied the logistic regression model (Peng et al., 2018; Peng et al., 2019), random forest model (Weir et al., 2018; Weir, 2019), and artificial neural network (ANN) model (Lai & Becker, 2010; Peng et al., 2019) in the predicative stage of enemy item detection. These methods use different underlying algorithms in computing the class membership probabilities. In addition, the classification results could vary depending on what probability cutoff is used to determine enemy item classification. The existing studies to date have evaluated the classification results at various probability cutoffs. The determination of an optimal cutoff often takes into account the accuracy of classification results, the practical goal of the enemy item classification, and resources allocated for enemy item review.

This study also aims to investigate whether the automatic enemy item detection procedure will help reveal more true enemy items previously not identified in the item pool and provide guidance for SME review. The classification results will be further evaluated by SMEs in order to detect more enemy relationships previously not flagged in the item bank due to the challenges of manual review. The enemy status between item pairs will be updated accordingly to improve the enemy relationship monitoring in the item bank.

Little research has been devoted to a systematic evaluation of various NLP techniques, classification models, probability cutoffs used for the classification, and whether the automatic procedure improves the enemy item identification in the item bank. Furthermore, existing studies for such automatic processes were each conducted on a different item pool, making it unfeasible to evaluate the relative performance of the automatic detection methods.

1.4 Research Objectives and Significance

The purpose of this research is to investigate and evaluate the performance of various automatic enemy item detection procedures using a single operational item pool. The study will compare the classification results across different conditions, varying the choice of NLP technique, classification algorithm, and classification cutoff. The results will be evaluated by SMEs in order to detect more enemy item pairs in the item bank. This research aims to provide findings on the effectiveness of the automatic enemy item detection process.

While the automatic enemy item detection process will not replace human SMEs in identifying and confirming enemy item relationships, it will provide SMEs with guidance and directions in enemy item review and thereby reduce the burden of enemy relationship monitoring in large item banks.

The research on item similarity will also offer insights into how the applications of NLP can address various challenges for bank management and maintenance for large-scale testing, such as identifying the topic coverage of the item bank and automatically detecting exposed test items.

2. REVIEW OF LITERATURE

2.1 Background of Natural Language Processing

Natural Language Processing is devoted to deciphering, understanding, and constructing the natural language for achieving human-like language processing for a wide range of applications (Liddy, 2009; Kurdi, 2016). First introduced for machine translation in the 1940s, the early application of NLP has urged the development of revolutionary language theories (Quillan, 1963; Chomsky, 1965, 1967; Fillmore, 1968) that enabled the translation of natural language into formal representations that are usable by computers. NLP prototypes were developed and demonstrated effectiveness in analyzing, replicating, and generating natural language (Weizenbaum, 1966; Woods, 1970; Winograd, 1971). However, these earlier applications of NLP were restricted to particular principles and are limited to isolated solutions.

Until the 1980s, the majority of NLP systems used complex, manual-defined rules that were not easily generalizable to broader applications in real-world contexts. A shift of focus from the closed domains of the earliest NLP research to open domains was made possible by the increasing availability of computational power, extensive textual resources of the internet, and digitalization of large-scale text corpora. The field began to move towards relying on empirical methodologies rather than theory-driven generalizations of natural language.

During recent decades, there was also a major shift in the understanding of the fundamental goal of NLP. It was discovered that a system that aimed to understand each and every word and extract the complete meaning of every input often resulted in complete success or complete failure. On the other hand, a system that tried to extract partial meaning from every input turned out to be more successful for real-world application (Bates, 1995). The field started

to accept this "partial correctness" as a meaningful and useful approach to NLP. At the center of this shift lies the understanding that the nature of language processing is too complex to be captured by hand-written rules. Rather, statistical processing and machine learning methods allow computer programs to infer patterns about sampled textual data and make predictions about new data, which could accomplish some language analysis tasks at a level comparable to human performance. Statistical techniques have become standard practices of NLP because they succeeded in handling many problems in computational linguistics (Hirschberg & Manning, 2015).

2.2 Natural Language Processing in Testing

2.2.1 Automated Essay Scoring

As modern testing made its rapid transition to computer-based testing, researchers began to adopt NLP methods in assisting various aspects of testing. Page (1966) developed the first automated essay scoring (AES) system to emulate human raters in evaluating and scoring essays. The goal of AES was to create a cost-effective method that could provide timely feedback on the writing performance and prevent drawbacks of human assessment (e.g. subjectivity, fatigue, speed) (Page, 2003; Myers, 2003). Page's method used measures derived from the surface features of the essay, such as average word length and counts of punctuation, as an estimate of intrinsic quality of writing (Chung & O'Neil, 1997; Rudner & Gagne, 2001). This method received criticism for neglecting the semantic aspect of writing and focusing on superficial structure (Kukich, 2000; Chung & O'Neil, 1997).

Subsequent scoring engines took advantage of NLP and Artificial Intelligence to capture more sophisticated features of essay writing. Landauer, Laham and Foltz (2003) developed the *Intelligent Essay Assessor*, a software application that uses the Latent Semantic Analysis (LSA)

to score the quality of conceptual content in the essay. LSA is a natural language processing technique that allows comparisons of the semantic similarity between textual documents. Using matrix algebra technique, LSA analyzes a corpus of texts in similar content and size and constructs a high-dimensional semantic space. Each piece of textual data is mapped onto this space, and the distance/similarity between any two pieces can be computed on a semantic level (Berry et al., 1995; Landauer & Dumais, 1997; Landauer et al., 1998; Foltz et al., 1999). To assess the quality of essays, LSA was applied to a corpus of domain-representative text. Domain-representative text involved materials (textbooks, articles, etc.) from which a student learned their vocabulary, concepts, and knowledge about the content domain. The written essay was then compared with texts of known quality with regard to their conceptual similarities. Compared to other methods that focus on mechanical and syntactic features, LSA demonstrates superiority with its capability to evaluate written essays on semantic aspects such as content, style, comprehensibility, and relevance. An overview of LSA is presented in section 2.3.2.

Meanwhile, the Electronic Essay Rater (E-rater) was developed to evaluate the quality of written essays and short answers using a corpus-based approach (Burstein et al.,2001; Burstein, 2003; Shermis & Burstein, 2013). Its scoring features a syntactic module, a discourse module, and a topical analysis module, each focusing on different aspects of the quality of writing (Burstein et al., 1998; Burstein, 2003; Shermis & Burstein, 2013). E-rater used part-of-speech tagging technique (Brill & Pop, 2000) to capture the syntactic variety of the essay. The output from the syntactic module was further utilized by the discourse module to annotate the discourse relations of the arguments presented in the essay (e.g. contrast relation, parallel relation). The discourse annotations were then used in the subsequent topical analysis to evaluate the content of the essay. The core of the topic analysis is based on the Vector Space Model (VSM) commonly

applied in information retrieval (Salton, 1989). E-rater used a representative collection of human scored sample essays as training essays, which were converted into vectors of word frequencies. To score each test essay, it was vectorized in a similar fashion, and a search was performed to identify sample essay most similar to the test essay through their cosine similarity indices. The test essay was assigned a score based on the quality of the closest matches among the training essays. An overview of Vector Space Model and cosine similarity measure is provided in section 2.3.1.

2.2.2 Automatic Item Generation (AIG)

Following its earlier application in automated essay scoring, NLP has been an increasingly active research area for automatic item generation. To reduce the time and cost of manual item writing, AIG was proposed to efficiently generate items for timely item bank replenishment. The previous application of AIG was focused on item modeling in which SMEs played a crucial role in organizing and structuring the content required for item generation using a cognitive item model. Based on the cognitive item model, an algorithmic process is derived to generate new item instances (Bejar et al., 2003; Embretson, 2002; Embretson & Yang, 2007; Gierl et al., 2008; Gierl & Lai, 2013, 2015; Embretson & Kingston, 2018).

Recent developments of AIG have utilized the strength of NLP to offer an alternative method of item generation. Mitkov and Ha (2003) proposed an NLP-based approach for automatic construction of test items. This approach identified important terms from instructive text (e.g. textbook chapters and encyclopedia entries), and transformed declarative sentences into questions to form the item stem. Lexical database[1] (e.g. WordNet[2]) was used to mine for terms

[1] A lexical database records lexical information and semantic relationships of a large collection of words (e.g. part of speech, synonym, antonym).
[2] WordNet is one of the largest publicly available lexical databases for the English language.

which are semantically close to the correct answer to create distractors[3]. Subsequent studies have extended this method to combine a corpora-based approach featuring topic modeling techniques[4] and a graphic-based approach featuring lexical databases and ontologies[5], to identify semantically close terms. The NLP-based methods that focused on identifying semantic similarity were shown to be successful in automatically generating valid test items with homogenous distractors (Mitkov et al., 2009; Aldabe & Maritxalar, 2014; Susanti et al., 2015; Liu et al., 2018; Shin et al., 2019).

The latest AIG research has utilized the deep learning approach, which is commonly used in the NLP application. Von Davier (2018) analyzed over three thousand publicly accessible personality items using the recurrent neural network model, and he found that the resulting automatically generated items are comparable with those selected from a manually generated item pool. This suggests that deep learning models could be a valuable source for replenishing item pools.

2.2.3 Item Bank Management and Maintenance

NLP techniques have also been adopted in emerging research areas for test development and item bank management/maintenance. Sheehan et al. (2006) applied Latent Semantic Analysis to link critical item stimulus to specific required skills targeted at different difficulty levels. Belov and Knezevich (2008) used semantic similarity measures computed based on WordNet to predict the item difficulty efficiently. Becker and Olsen (2012) applied Latent Semantic Analyisis in identifying the coverage of textbook content and glossary terms in the

[3] The distractors refer to the incorrect options in a multiple-choice item.
[4] A topic modeling technique is an algorithm for extracting the abstract topics in a collection of documents. It is frequently used for discovering hidden semantic structure in textual data.
[5] Ontology refers to a database designed to categorize sets of concepts in a content area and to show their definitions, properties, and the relationships between them.

item bank in order to facilitate targeted, evidence-centered item writing. This study showed that LSA was efficient in detecting the gap in content coverage for the item bank, which helped provide specific targeted topics for SMEs in item writing. McLeod et al. (2015) found that some linguistic features such as readability, sentence structure, parts of speech, and tense significantly predicted item difficulty.

2.2.4 Item Similarity

To address the time-consuming and resource-intensive process of manual review on large item banks, Becker and Kao (2009) proposed an NLP-based approach for automatic identification of similar item sets. The Vector Space Model was applied to produce a word-embedding matrix from a pool of items, and based on this, the cosine similarity measures between each item pair can be calculated. The resulting cosine measures were used as key indices in evaluating the degree of between-item similarity. This automated approach has shown promise in addressing several challenging tasks presented by large-scale item bank management and test security concerns. It was used to compare allegedly exposed items with the actual item bank of a national exam, and it successfully flagged a number of potentially compromised items for further SME review. This approach was also applied to detect enemy and duplicate items in an item bank. The study found that 98% of enemy item pairs were identified above the mean of the cosine similarity indices. In the same study, the authors also explored using the resulting cosine measures in predicting the content area of each sampled item. The results successfully matched 74% of the items to their actual content area classification in the item bank.

A series of subsequent studies was devoted to the automatic detection of enemy items. Lai and Becker (2010) extended the research by utilizing a deep learning model known as Artificial Neural Network (ANN) for the classification process and including additional

structural linguistic similarity measures, such as word-overlap and longest common subsequence, as major predictors of enemy status.

The automatic approach based on Vector Space Model mainly relied on the matching of words between items and was limited in capturing the conceptual similarity between item pairs. Since enemy item pairs typically contain different terms pertaining to the same concept, measuring item similarity on a semantic level remained a challenge. Li et al. (2012) utilized a lexical database and a specialized medical ontology database to calculate the semantic similarity between items in a medical examination. The study suggested that though these similarity indices cannot replace human reviewers, they were able to assist them in detecting enemy items through iterations. Peng et al. (2018) made use of Latent Semantic Analysis to explore the possibility of measuring semantic similarity between items. The findings suggested that LSA was able to extract meaningful semantic concepts from the item pool data. The automatic approach using semantic similarity measures computed from LSA successfully recovered 76% of enemy item pairs in the bank. Some of the false positive item pairs were sent to SMEs for enemy status review, and the majority were confirmed to be true enemy pairs that were not previously flagged in the item bank. An extended study (Peng et al., 2019) examined the effectiveness of LSA technique on a different item pool and compared classification results across various statistical and deep learning models. The study found consistently high recall and precision rate for the classification results across the models. The model using an ANN classifier appeared to out-perform that with a logistic regression classifier when using a higher probability cutoff for classification, but the performance was reversed at a lower probability cutoff. An overview of logistic regression and Artificial Neural Network model is presented in section 2.4.

Some studies have derived semantic similarity measures using another NLP topic modeling technique called Latent Dirichlet Allocation (LDA) and utilized the random forest model as the predictive method (Weir et al., 2018; Weir, 2019). LDA was shown to demonstrate promising capability in extracting meaningful topics from the item pool data. The classification results showed high recall rate, predicting over 70% of the true enemy items in the sample. An overview of Latent Dirichlet Allocation is provided in section 2.3.3.

2.3 Overview of NLP Techniques for Measuring Textual Similarities

This section reviews the three NLP techniques featured in this study for deriving the similarity indices between items: Vector Space Model, Latent Semantic Analysis, and Latent Dirichlet Allocation.

2.3.1 Vector Space Model

Vector Space Model is an algebraic model commonly used for computing text similarity in information retrieval (Salton, 1968, 1989; Salton et al., 1975; Manning et al., 2008) and natural language processing (Levy & Bullinaria, 2001; Lowe, 2001; Padoo & Lapata, 2004). The basic premise underlying VSM is that the meaning of a document can be derived from the key words constituting the document (Aswani et al., 2012), and that the context surrounding a word contribute valuable information to its meaning (Harris, 1968). The popularity of Vector Space Model lies in its ability to represent large textual data by using distributional statistics. Using matrix algebraic techniques, VSM analyzes a corpus of text containing vocabulary, concepts, and knowledge relevant to the content area, and it constructs a high-dimensional vector space where every word and text document are mapped in relation to the conceptual components found in the corpus. The similarity between any two documents can be estimated by comparing their relative positions in the vector space.

2.3.1.1 Document-Term Matrix

Like most NLP techniques, VSM requires that the text data be transformed into formal representations that can be processed and analyzed by computer (Jurafsky & Martin, 2000). In this step, textual documents from a corpus are projected onto a vector space by mapping the distributional patterns of word co-occurrence. This information is typically captured in a frequency matrix, where each row represents a piece of textual information in the corpus, commonly referred to as a *document*, and each column corresponds to a unique word in the corpus, commonly referred to as a *term*. Each element in this frequency matrix represents the frequency of a given term that occurs in a given document. The document-term matrix is also known as the word embedding matrix (Bengio et al., 2003; Collobert & Weston, 2008; Mikolov et al., 2013). It presents text documents in a multidimensional space where each *term* corresponds to a dimension. An example vector space constructed from *m* documents with *n* unique terms defined by a matrix A is given as follows:

$$\begin{matrix} n\ terms \\ \downarrow \end{matrix}$$

$$\mathrm{A} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \quad \leftarrow m\ documents$$

Figure 1 Document-term matrix constructed from *m* documents with *n* unique terms

Each row in this document-term matrix corresponds to a document (d_i) in the corpus and can be represented by a vector with *n* dimensions:

$$\overrightarrow{d_i} = \{x_{i1}, x_{i2}, x_{ij}, \ldots, x_{in}\} \tag{1}$$

Each element x_{ij} represents the frequency of the j^{th} unique term observed in the i^{th} document. In the context of an item bank, an item can be treated as a document and a unique word observed in the item bank as a term, based on which a document-term matrix can be constructed to form a vector space. Each item vector can be projected onto the multi-dimensional vector space. Figure 2 shows an example of vector space constructed from a hypothetical item bank with three items (vectors) and three unique terms (dimensions).

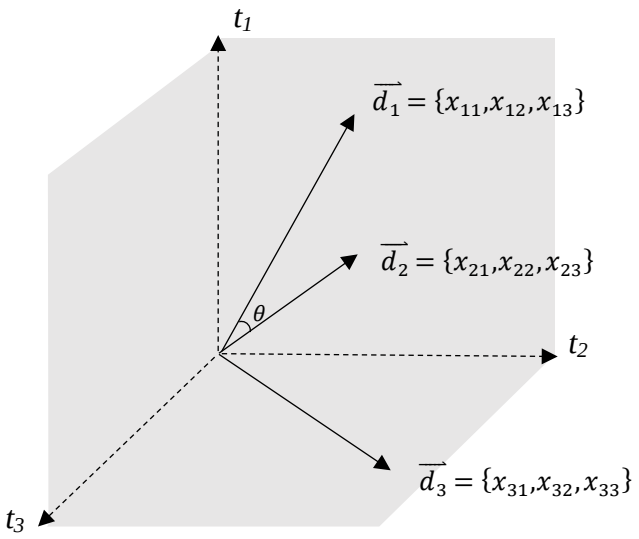

Figure 2 Vector representation of documents in a vector space

2.3.1.2 Cosine Similarity Index

As depicted in Figure 2, item 1 ($\overrightarrow{d_1}$) and item 2 ($\overrightarrow{d_2}$) are more likely to be similar compared to item 3 ($\overrightarrow{d_3}$), as their vectors appear to be closer to each other. There are different ways to measure vector similarity. Commonly used measures of vector distance include Euclidean distance, Manhattan distance, Hellinger, Bhattacharya, and Kullback-Leibler. One popular method for computing the similarity of two frequency vectors in NLP is the cosine similarity measure. Bullinaria and Levy (2007) compared cosine similarity measure with the five distance measures on a number of tasks involving word similarity and found that the cosine

similarity measure out-performed the rest. In other words, the inner angle between two vectors in the document space conveys essential information about the similarity between them. The cosine of the angle θ between two n- dimensional vectors $\overrightarrow{d_j}$ and $\overrightarrow{d_k}$ is calculated as follows:

$$\text{cosine}\,(\theta) = \frac{\overrightarrow{d_J} \cdot \overrightarrow{d_k}}{|\overrightarrow{d_J}||\overrightarrow{d_k}|} = \frac{\sum_{i=1}^{n} d_{ij}\, d_{ik}}{\sqrt{\sum_{i=1}^{n} d_{ij}^2}\sqrt{\sum_{i=1}^{n} d_{ik}^2}} \quad (2)$$

The cosine similarity is the dot product of the two vectors, scaled by their magnitude. Cosine similarity measure ranges from -1 to 1. When two vectors are highly similar, with the inner angle between their vectors approaching 0 degree, their cosine index will approach 1. When two vectors are unrelated, the inner angle between their vectors will be close to 90 degrees and their cosine index will approach 0. When the two vectors are pointing in different directions (with an inner angle larger than 90 degrees), the cosine index will be negative.

2.3.1.3 Bag-of-Words Approach

It is worth noting that the Vector Space Model and the other two NLP techniques (Latent Semantic Analysis and Latent Dirichlet Allocation) that will be applied in this study are all based on the "bag-of-words" assumption. That is, the order or structure of the terms in a document is insignificant (Harris, 1954). This assumption implicitly treats documents as a mixture of ideas and, as such, some topic models can be generated based on this structure (Steyvers & Griffiths, 2007). The bag-of-words approach is simple to understand and implement. It offers flexibility for comparing text documents as well as the concepts behind them. Therefore, bag-of-words models are suitable for the purpose of this study to detect item similarity. However, the bag-of-words method suffers from the drawback that some context and meaning can be stripped from the document with this form of representation.

2.3.2 Latent Semantic Analysis

Latent Semantic Analysis is a topic modeling NLP method based on the Vector Space Model (Furnas et al., 1988; Dumais, 1994), aimed at addressing the limitations that plagued the Vector Space Model. Consider the following sentences:

The flight departs around noon.

The plane leaves at twelve o'clock.

These sentences use different words to convey essentially the same meaning, but a classic VSM would have difficulty detecting the similarity between them because of the word difference. Different words can be associated with the same concept, and most words have multiple meanings (Deerwester et al., 1990). Therefore, the similarities between two pieces of textual information cannot be fully captured when comparing lexical features[6] such as vocabulary and punctuation (Furnas et al., 1988). LSA was developed to capture the semantic aspect of documents. It assumes that natural language has a latent structure which is often obscured by word usage (e.g. synonym, polysemy[7]) and aims to acquire a mathematical representation of the relationships among words, documents, and concepts through statistical computations based on a large corpus (Landauer et al., 2003). Using statistical methods, LSA has the ability to extract the latent semantic structure underlying the textual data and derive the similarity between texts on a conceptual level. Not only does LSA identify synonyms, it also recognizes polysemes (i.e. words with multiple meanings). When appropriately trained, LSA is able to distinguish *fly* as a verb and *fly* as a noun, thereby associating its different meanings with separate semantic concepts.

---

[6] *Lexical features* refer to words or vocabulary of a language.
[7] P*olysemy* refers to a word with multiple meanings.

To extract semantic concepts underlying the text, LSA applies a core mathematical operation known as Singular Value Decomposition (SVD) on the document-term matrix. Similar to Principal Component Analysis, SVD conducts dimension reduction on the matrix and is more efficient for a sparse matrix such as a document-term matrix. Any rectangular matrix A with $m \times n$ dimensionality can be decomposed into the product of three other matrices:

$$A = U \Sigma V^T \tag{3}$$

where U is an $m \times r$ matrix having the m document as its rows and r factors (factorized dimensions) as its columns, V^T is an $r \times n$ matrix having the r factors as its rows and the n terms as its columns, and Σ is an $r \times r$ diagonal matrix having the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(r,r)}$ of the factors in order along its diagonal.

Matrix U is a lower rank matrix with each row representing a document and each column representing a factorized dimension. Substantively, this is analogous to grouping m terms into r semantic concepts and representing each item with these r concepts.

On the other hand, each element of the matrix V^T represents the score, or loading, of a given document on a given concept. Matrix V^T represents each of the r concepts as a combination of the n terms. It shows the score of a given term on each of the r concepts.

The singular values on the diagonal of matrix Σ represent the strength of the factors, and it can be used to calculate the variance explained by each concept. This information is helpful in determining the number of the $r$concepts to include (or exclude) in the analysis. In LSA, the three matrices produced by SVD are truncated to retain k concepts/dimensions explaining the most variance in the textual data, so that noises are excluded from the reduced matrices. The number of retained concepts/dimensions k can be chosen arbitrarily, but it was suggested that selecting about 300 to 1000 dimensions, depending on the size of the corpus, results in good

performance for LSA (Landauer & Dumais, 1997). The reduced matrix *U* with k concepts is considered a sufficient approximation of the original matrix A. Figure 3 illustrates the mathematical representation of truncated SVD retaining k concepts.

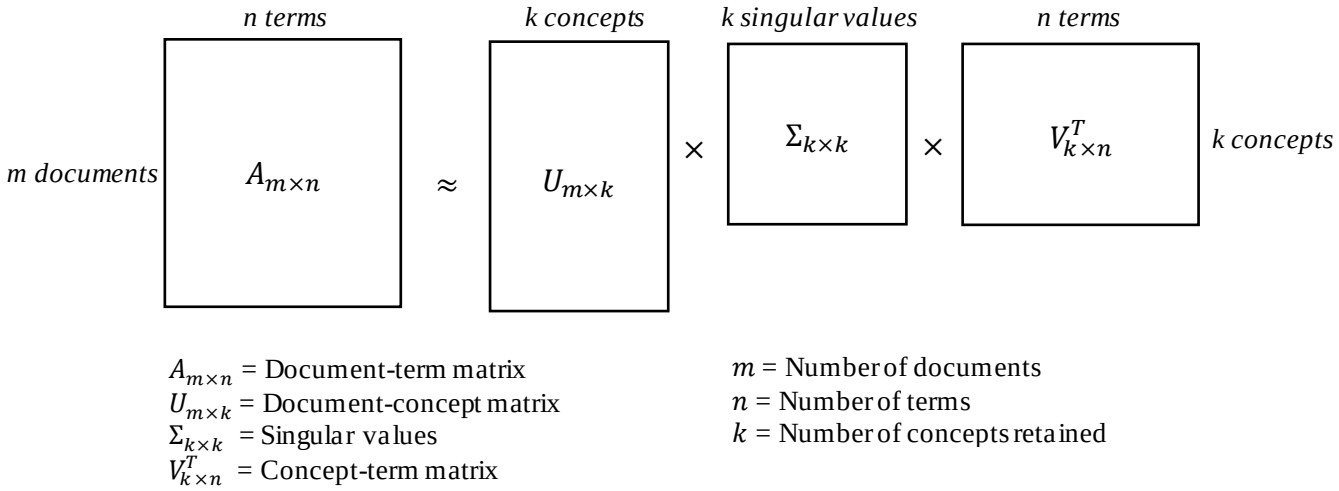

Figure 3 Mathematical representation of truncated Singular Value Decomposition

Matrix *U* is a crucial product of LSA, based on which the similarity index between any two documents can be derived. In our case, each row of matrix *U* is a *k*-dimension vector corresponding to an item in the item bank. The cosine similarity measure can be calculated between any two item vectors in a similar fashion as in Vector Space Model. The resulting similarity measures capture the semantic similarity because it is based on the conceptual factors derived through dimensional reduction. In other words, LSA has the ability to recognize the similarity between two different terms with similar meanings through the underlying semantic structure derived from SVD. LSA can also recognize a term with multiple meanings, because the term will have separate scores on different associated concepts. Furthermore, LSA can identify terms frequently co-occurring and closely associated with a concept. For example, the terms *bow*

and *arrow* will both have higher scores on an extracted concept labeled *archery* and thus appear to be closer in the semantic space.

Compared to the Vector Space Model, Latent Semantic Analysis has the advantage of capturing conceptual similarities between the terms and documents. Simulation studies have shown that LSA closely reflects human cognitive phenomena, such as sorting and categorization of words and judgement of word similarities (Landauer et al., 1998).

2.3.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation is another topic modeling technique that takes a generative probabilistic approach towards capturing the semantic topics underlying the textual data (Blei et al., 2003). It perceives each document as a mixture of several latent topics and assumes that topic distribution in all documents share a common Dirichlet prior. Each latent topic in the LDA model is also represented as a mixture of terms and the term distributions of topics share a common Dirichlet prior as well.

As a generative approach, LDA takes into account how observed data is generated in order to build a model. It attempts to synthesize the observed information (e.g. word frequencies) using an approximated generation procedure to uncover hidden topics without any labels. The generative process of LDA consists of three layers. Given a corpus D consisting of M documents, with document d having N_d terms:

(a) For each topic $k = 1, \dots, K$, draw a multinomial distribution over a vocabulary of terms from a Dirichlet distribution with parameter β, $\phi_k \sim Dirichlet(\beta)$;

(b) For each document $d = 1, \dots, M$, draw a multinomial distribution over topics from a Dirichlet distribution with parameter α, $\theta_d \sim Dirichelet(\alpha)$;

(c) For each term t_n $(n = 1, \dots, N_d)$ in document d,

(i) draw a topic $z_n \sim Multinomial\ (\theta_d)$;

(ii) generate a term from the corresponding distribution over terms, $t_n \sim Multinomial\ (\phi_{z_n})$.

In this generative process, terms in documents are the only observed variables while others are latent variables (ϕ and θ) and hyper parameters (α and β). The probability of observed data D is computed and obtained as follows:

$$P(D|\alpha,\beta) = \prod_{d=1}^{M} \int P(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_n} P(z_n|\theta_d)\, P(t_n|z_n,\beta)\right) d\theta_d \,, \quad (4)$$

where α is a parameter of topic Dirichlet prior and the distribution of terms over topics is drawn from the Dirichlet distribution with parameter β. Figure 4 shows a graphical representation of LDA.

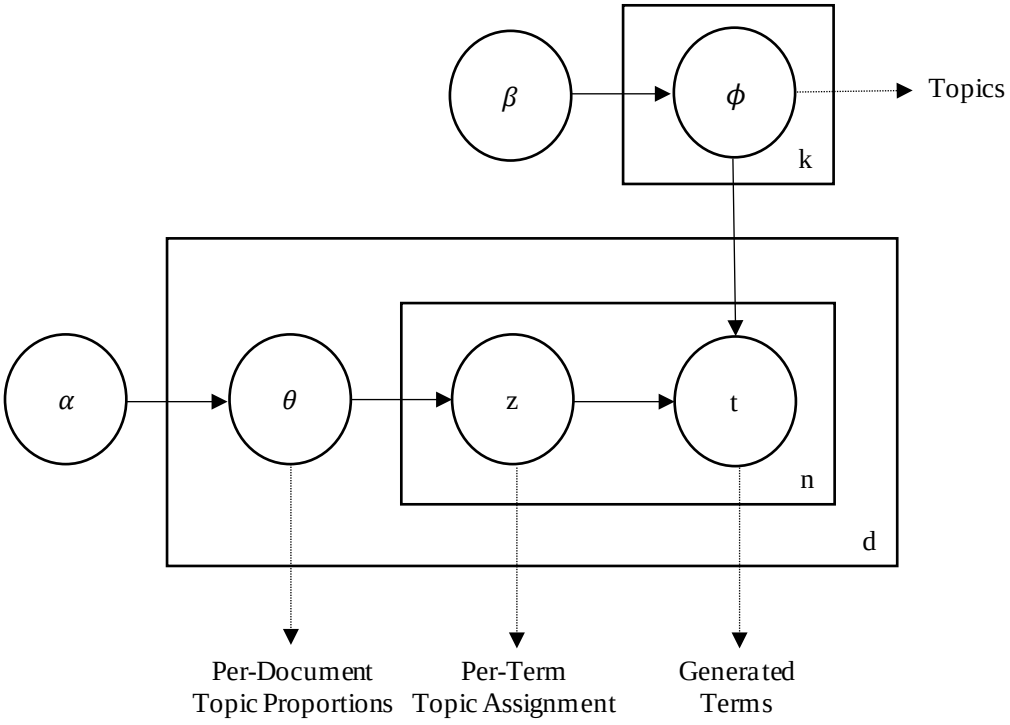

Figure 4 A conceptual representation of Latent Dirichlet Allocation

The training of LDA requires the input of two symmetrical priors (α_{prior} and β_{prior}). The α_{prior} is equivalent to the inverse of the number of topics (dimensionality K of the document-topic distribution), which assumes that each of the topic has an equal chance of being selected in a document prior to the LDA training. The number of topics (K) is assumed to be known and fixed. The β_{prior} is equivalent to the inverse of the number of terms (dimensionality N_d of the topic-term distribution), which is known after obtaining the document-term matrix. This indicates that each of the terms has an equal chance of being selected in a topic prior to the LDA training. Once the number of terms and the number of topics are determined, a topic distribution is specified (e.g., 45% psychology, 35% neurology, and 20% education). Next, a topic is drawn from the topic mixture distribution and a term is selected according to the topic distribution over words. The LDA repeats this process until all the terms are generated for each document.

With this process, LDA aims to compute a posterior distribution of the latent variables in a document to discover the structure of hidden topics. Due to an enormous number of possible topic distribution, computing the probability of specific terms under a certain topic becomes extremely difficult. To address this problem, LDA uses Gibbs sampling (Griffiths & Steyvers, 2004; Porteous et al., 2008) to conduct conditional distribution sampling. It starts the process by randomly assigning each term in the document to one topic, which produces an initial guess of the term-topic and term-document distribution. Next, it makes successive attempts to find the conditional distribution of a term's topic assignment conditioned on the rest of the topic assignments bay updating the assignment of the current term. This conditional sampling is repeated until a stationary state of assignment is reached, which is then used to estimate the topic

mixtures for each document. The cosine similarity measure between any two documents can be calculated based on the topic distributions of two given documents.

2.4 Classification Modeling

At the second stage of the automatic enemy item detection, the cosine similarity measures between items, obtained from the NLP methods described above, are entered into a classification model to predict the enemy status. Classification modeling takes the input variables and attempts to generate discrete output through a mapping function. Predicting the enemy status of item pairs is a binary classification problem where the outcomes may take two values: 1 (enemy) and 0 (non-enemy).

In a typical scenario of statistical or machine learning classification, the data are split into a training set and a test set. The training set of data contains measurements for a set of objects (in our case, item pairs) with known class membership. A prediction model is fit on the training data. This step is often referred to as *training* or *learning* in machine learning. Successively, the fitted model is applied to the test set of data to predict the outcome for new unseen objects (Hastie et al., 2009; Alpaydin, 2014).

Once the prediction model fit the training data, a probability $P(y|X)$ was calculated for each object where X represents the measures/predictors of the objects and y represents the class membership (often represented by 0 or 1 for binary outcomes). We may establish a classification rule based on this class membership probability, for example:

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1|X) > .50, \\ 0 & \text{if } P(y = 1|X) \leq .50. \end{cases} \tag{5}$$

This rule assigns the object as a member of class 1 if the probability exceeds .50, and otherwise as a member of class 0. The probability cutoff can be chosen arbitrarily, and it often

varies depending the goal of the classification. In many cases, the classification model does not produce a perfect classification in which every object in the sample is assigned to the class to which they truly belong. Some errors are likely to occur during the approximation of the predictive model, in which objects are assigned to a wrong class. In a binary classification model, these errors are referred to as *false positives* and *false negatives*. False positives represent cases incorrectly reported to be positive, but are in fact negative. For example, a classification model may return a positive result indicating that a pair of items are enemies even if they are not flagged as enemies in the item bank. Similarly, false negatives refer to cases incorrectly reported to be negative.

Given the fact that most items in a given item bank do not have an enemy relationship, the number of true non-enemy item pairs is typically very large. Therefore, existing studies usually found a Type I error rate of under 1%. However, the number of false positive item pairs could still amount to hundreds or thousands (Lai & Becker, 2010; Li et al., 2012; Peng et al., 2018; Peng et al., 2019; Weir, 2019). In the context of this study, because the flagging of enemy relationships in large item banks is likely to be incomplete, the false positive item pairs would ideally undergo SME review to determine if the enemy status flagging was previously incorrect or overlooked. The goal of the automatic detection process is to identify as many true enemy item pairs as possible while keeping the number of misclassified cases manageable for content review. The decision of the probability cutoff often reflects a balance between classification accuracy and cost for manual review. In existing studies which have employed SMEs in subsequent review of high-probability enemy item pairs predicted by the model, the probability cutoff selected to qualify item pairs for SME review ranged from .50 to .90 (Li et al., 2012; Peng et al., 2019; Weir, 2019). Based on the result of the review, newly confirmed enemy

relationships can be updated in the item bank. Each iteration of this process increases the accuracy of enemy item flagging in the item bank.

The classification model utilizes an algorithm that maps input data to a class. The algorithm that implements the classification is known as a *classifier*. No one algorithm is superior to the others for every problem. Therefore, it is common to apply different classifiers on the training data and then use the test set of data to evaluate performance for the purpose of selecting the best classifier. The following sections will provide an overview of the two classifiers evaluated in this research.

2.4.1 Logistic Regression Classifier

The logistic classifier is perhaps the most popular for solving binary classification problems (Hosmer et al., 2013). The logistic classifier is designed to estimate the probability of class membership as a logistic function of linear combinations of predictors (Hardle & Simar, 2012).

Consider the vector *y* of observations on a binary response variable. The logistic model assumes that the probability for observing $y_i = 1$ on a particular vector of $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$ is given by the logistic function of a linear combination of x:

$$P(x_i) = P(y_i = 1|x_i) = \frac{\exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})}{1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})}. \quad (6)$$

This entails the probability of the absence of the trait:

$$1 - P(x_i) = P(y_i = 0|x_i) = \frac{1}{1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})}, \quad (7)$$

which implies

$$\log\left\{\frac{P(x_i)}{1 - P(x_i)}\right\} = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}. \quad (8)$$

The estimates of βs can be obtained by Maximum Likelihood Estimation (MLE) (Eliason, 1993). The logistic classifier will then produce the probability of possessing the trait $P(y_i = 1|x_i)$ for each subject, and the class membership can be determined with the classification rule defined by Eq. (5).

2.4.2 Artificial Neural Network Classifier

In logistic regression, classification is achieved as the result of a linear transformation. However, the classification of certain data is not linearly separable. The artificial neural network classifier is another tool with the capability of solving complex nonlinear classification problems (Rumelhart et al., 1986). ANN is also known as the multilayer perceptron algorithm. It is inspired by the biological neural system immanent in the human brain and therefore designed to mimic the complex process of human learning and classifying (McCulloch & Pitts, 1943). The ANN algorithm has been applied to solve problems in speech processing (Gorin & Mammone, 1994), pattern recognition (Jain et al., 2000), clustering and classification (Zhang, 2000), and function approximation (Selmic & Lewis, 2002), etc. An artificial neural network consists of a large collection of units, commonly referred to as *neurons* (*nodes*), that are interconnected to allow communication between the neurons. Each neuron relates to other neurons through connection links that carries information about the input signal. These connections are analogous to the *synaptic links*[8] that exchange information between neurons through a neural system. A weight is associated with each connection that usually excites or inhibits the signal that is passing through. Each neuron has an activation function that processes the input signals with an activation rule to produce the output signals that may be sent to other neurons.

[8] A synaptic link is a connection that permits a neuron to pass an electrical or chemical signal to another neuron in a neural system.

The neurons are organized into three types of layers: input layer, hidden layer(s), and output layer. The architecture of an ANN is illustrated in Figure 5. The predictor variables enter the network as input nodes, which form the input layer. Each node is capable of processing and passing the information forward through the connecting weights to the nodes in the next layer.

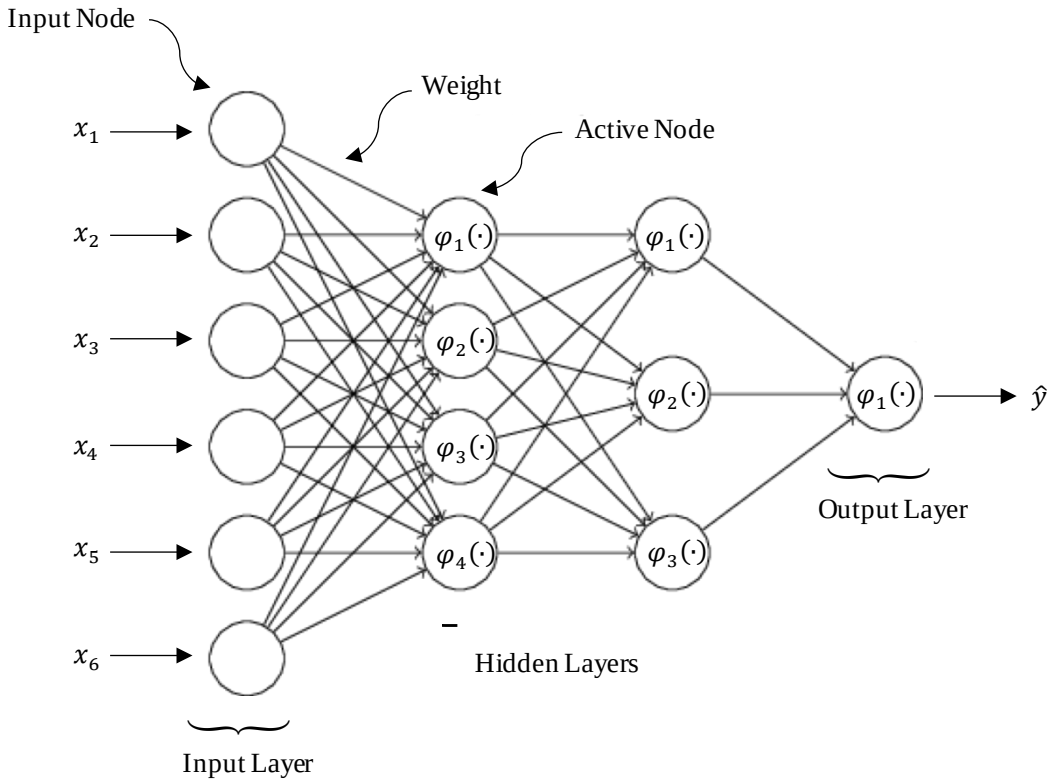

Figure 5 Structural representation of an Artificial Neural Network for binary classification

The hidden layer(s) lie between the input layer and output layer. A neural network can have multiple hidden layers depending on the complexity of the problem. The nodes in the hidden layers receive the information coming through the connecting weights from the previous

layer and calculate an output to pass on to nodes in the next layer. The output of a node y_i is computed as:

$$y_i = \varphi_i \left(\sum_{j=1}^{n^i} w_j^i z_j^i + b^i \right), \quad (9)$$

where n^i is the total incoming connections, z^i is the input, w^i is the weight associated with the input, b^i is the bias, and $\varphi_i(\cdot)$ is the activation function. The node calculates a weighted sum of the incoming input, adds a bias, and uses the activation function to evaluate whether the node should be activated. Based on the activation function and the threshold it applies, a neuron's output rate is controlled. The activation function eliminates weak connections between neurons and limits the rate the information passes through the hidden layers. Eventually, the resulting neurons in the output layer yield the class membership probability.

ANN is optimized by minimizing the difference between the desired output (y_i) and the model output ($\hat{y}_i$) using a cost function. One algorithm typically used to compute the set of weights that minimizes the cost function is the gradient descent method (Dreyfus, 1990; Kingma & Ba, 2015). Backpropagation is then used to iteratively update the weight of neurons based on the steepest descent direction (Rumelhart et al., 1986; Werbos, 1994).

2.5 Summary of Literature and Proposed Study

This study aims to evaluate various approaches for identifying enemy item pairs. Three NLP methods for measuring text similarities will be compared as a component of the classification model: Vector Space Model, Latent Semantic Analysis, and Latent Dirichlet Allocation. The Vector Space Model is a simplistic approach used to measure the attributional similarity of words. Latent Semantic Analysis enhances the VSM approach by dimensionality reduction to extract meaningful concepts for measuring semantic similarity between texts. LSA is one of the many methods for extracting semantic dimensions from sparse, noisy frequency-

based data. Researchers have applied Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999), Non-negative Matrix Factorization (Lee & Seung, 1999), and Iterative Scaling (Ando, 2000) to this problem. Results showed that none of these methods performs substantially better than LSA scaled up to large matrix sizes (Turney, 2006). Latent Dirichlet Allocation is a generative probabilistic approach to measure text similarity based on the Probabilistic Latent Semantic Analysis. It is a special case of PLSA in which the mixture probabilities follow the Dirichlet multinomial distribution. LDA is generally recognized as an improvement over PLSA (Blei et al., 2003; Fernandez-Beltran & Pla, 2018). Therefore, PLSA will not be covered in this study.

This study will also compare two classifiers in the classification stage of enemy status by comparing the popular logistic regression algorithm to the machine learning algorithm commonly used in the field of natural language processing. The classification results produced from each combination of the NLP methods and classifiers will be examined at different probability cutoffs. Previous studies have evaluated the classification results at multiple cutoffs (Lai & Becker, 2010; Li et al., 2012; Peng et al., 2018; Weir et al., 2018; Peng et al., 2019; Weir, 2019). Given the limited number of studies comparing the NLP methods and classification models at different probability cutoffs for the purpose of automatic detection of enemy item pairs, additional research is required to evaluate the merits of each.

This research will contribute useful implications to the field of large-scale testing on the relative strength of these approaches for item bank management and maintenance. Additionally, the findings may be beneficial to examination contexts outside of licensure and certification.

3. METHOD

The chapter describes the methods employed in this study to address the following research questions:

a) Do natural language processing techniques adequately capture item similarity?

b) Compared to a logistic regression classifier, does the Artificial Neural Network classifier improve the accuracy of classifying enemy item pairs?

c) What probability cutoff is considered optimal for classifying a sufficient number of existing enemy item pairs while keeping the number of falsely classified item pairs manageable?

d) Does the automatic enemy item detection procedure help reveal more true enemy items previously not identified in the item pool? Does retraining the model, using the input from SME review of false positive item pairs, help improve the accuracy of classifying enemy item pairs?

3.1 Item Data

The National Council of State Boards of Nursing provided the item data used in this study and access to the SMEs who offered feedback on the results of enemy item pair prediction. The test items aim to test the capability of examinees to manage and fulfill specific medical care needs, including concepts and processes fundamental to the work performed by nurse aides. The main role of a nurse aide is to provide basic care to patients under the supervision of registered nurses or licensed practical nurses, including assisting patients in their daily activities, such as bathing, dressing, and moving the patients; measuring and monitoring vital signs and intake/output of the patients; ensuring bed safety and sanitization, as well as answering call lights. The requirements for nurse aide candidacy vary from state to state. Typically, a candidate

is required to have a high school diploma or a certificate of General Educational Development. To be qualified to take the examination, a candidate also must undergo a training that includes coursework and clinical practice hours, which can take four to 14 weeks to complete.

The data consists of 1,461 multiple-choice items from the item bank of a national licensure examination for nurse aides. On average, each of the multiple-choice items contains 2 to 3 sentences in the item stem, with approximately 7.5 words per sentence (before the NLP data processing and transformation). Each item provides four response options, including one correct answer key and three distractors, with the average length being 5 words per option. The test plan requires that the readability of an operational item pool should not exceed 1,200 Lexiles based on the Lexile Framework for Reading, which is equivalent to high school graduate level (MetaMetrics Inc., 2020). Only entry level cognitive ability (Bloom et al., 1956) is required to solve the majority of items, which involves memorization and recall of care protocols, and the main objective of the items is to test the candidates' knowledge to recall, execute and abide by these protocols.

Some enemy item relationships have been previously identified within the item bank through item review and test form assembly processes. A total of 453 items have at least one enemy item within the sample. However, it is unlikely that all enemy item relationships were captured in the sample, because the items were not systematically compared with each other and scrutinized for enemy item relationships prior to this study.

Each multiple-choice item in the sampled item bank consists of an item stem, an answer key, and three distractors. These item components were the initial input data for the natural language processing methods. The dataset also includes information regarding the content area

and difficulty parameter of each item, which were later entered into the classification model as predictors along with the calculated similarity indices.

3.2 Data Transformation

To facilitate the item similarity analysis, the text of the sampled items was transformed into a document-term matrix, which is required by the three proposed NLP approaches as the initial input matrix. To enable the calculation of different similarity indices between each item pair, the stem and answer key were treated as separate documents in the document-term matrix. Four cosine similarity indices were calculated for each item pair: stem to stem similarity, key to key similarity, the stem of the first item and the key of the second item, and vice versa. These distinctive types of similarity indices captured different characteristics of enemy relationships. For example, the similarity between the stem of an item and the answer key of another item captured enemy relationships due to clueing. The distractors of each item were not included in this study as they are not typically considered strong indicators of enemy relationships and are not evaluated in the enemy item screening. A previous study by Becker and Kao (2009) also showed that the exclusion of distractors increased the similarity estimates of clueing enemy items. As a result, a total of 2,922 documents were included in the document-term matrix, including 1,461 item stems and the corresponding 1,461 answer keys. The four similarity indices described above were calculated between each item pair. These similarity indices were later matched with each item pair and were used as key predictors in the classification stage.

The data transformation process included multiple steps of text cleaning and processing. The first step was parsing the text of each document into distinct terms. This step also removed the punctuation and case for each term. Next, each document was pruned of stop words. Stop words refer to common words such as articles, pronouns, adjectives, adverbs, and prepositions.

These words are so common in the English language that they do not provide useful information for deriving the meaning of the documents. In addition to the common stop words, frequent words and phrases in the given content domain were identified and added to the list of stop words. For example, many items in this bank contained the words “client” and “nurse aide”. Therefore, these words did not contribute useful information to the distinct meaning of the item. Some common phrases, or templates, were used in the item stems. For example, phrases such as “what is the best response” and “which of the following” were commonly used as the prompt. These words or phrases were filtered out so that they would not artificially inflate the similarity between items using the same templates.

The next step of the data transformation was lemmatization, which converted the remaining terms into their lemmas, thereby reducing different variations (e.g. tenses and forms) of a word to its dictionary form. For grammatical purposes, some words have several inflected forms that convey the same meaning, and the word that usually represents this set of words in the dictionary is the lemma. For example, *runs*, *ran*, and *running* all share the same lemma, *run*. Lemmatization is closely related to stemming, which was commonly used in the previous studies of enemy item identification (Becker & Kao, 2009; Lai & Becker, 2010; Weir, 2019). The main difference between lemmatization and stemming is that lemmatization takes into account the intended part of speech and meaning of a word, while stemming simply converts a word into its root form by applying a series of inflection rules without knowledge of the context (e.g. removing the prefix or suffix of a word). For instance, lemmatization will recognize the difference between *happy* and *happiness* and keep them as they are, but stemming will convert both into the same root *happi*; lemmatization will convert both *geese* and *goose* to *goose*, while stemming will convert *geese* to *gees* and *goose* to *goos*. In general, lemmatization is more

precise than stemming, because it will always return a meaningful dictionary word that is appropriate for the context, but this accuracy comes at the sacrifice of speed. The NLTK package in Python 3.7 was used to lemmatize the terms, utilizing the WordNet corpus to identify the lemma of each word (Bird et al., 2009). Eventually, each document was represented by a shorter list of lemmas capturing the essential meaning of the document.

The above steps produced a list of unique terms occurring across the documents. These terms were then mapped onto the document-term matrix in which each row represented a document and each column represented a unique term in the item bank. The frequency of occurrence for each unique term in each document was recorded as elements within the matrix. Next, a technique known as *Term Frequency–Inverse Document Frequency (TF-IDF)* weighting was applied to the raw frequency. The rationale of TF-IDF transformation lies in the fact that the raw frequency of a term does not necessarily reflect its importance to a document. The more prevalent a term is across the whole corpus, the less contribution it will make to distinguish certain individual documents from the remainder of the collection. In the context of a nurse aide examination, terms such as *care*, *assist*, and *help* are commonly observed across the items because they describe the routine work of a nurse aide. These terms may occur more frequently in certain documents than other terms, but their raw frequencies do not provide much information on the similarity (or difference) between the items. The less frequent terms across the collection, on the other hand, are more helpful in discriminating certain documents from the other documents. This implies that the importance of a term should take into account both its frequency in a document and its frequency across the whole collection. Therefore, TF-IDF becomes a common method to adjust the weight of the terms based on the term frequency and inverse document frequency. The TF-IDF weight consists of two statistics. The first part is the

normalized term frequency (TF), which is the number of times a term appears in a document divided by the total number of terms in that document. This normalized term frequency is then offset by the inversed document frequency (IDF), which is the logarithm of the number of the documents across the corpus divided by the number of documents in which the specific term appears. The TF-IDF weight is calculated as the product of TF and IDF. The TF-IDF transformation was applied on the raw document-term matrix to adjust the raw frequency values based on each term's occurrences in the documents as well as its frequency across the entire item bank. The resulting weighted document-term matrix was passed to the three NLP approaches as initial input. This matrix contained 2,922 rows, with each row representing an item stem or an item key. Each column in this matrix corresponded to a unique term in the item bank.

3.3 Computation of Similarity Indices Using NLP Techniques

3.3.1 Cosine Indices Based on the Vector Space Model

The similarity indices between each item pair for vector space model were computed based on the weighted frequencies in the transformed document-term matrix. To compute the similarity measures of i^{th} and j^{th} items, the program first identified the stem vectors and key vectors of the two items. The cosine similarity measures were calculated between the stem vectors of both items, between the key vectors of both items, between the stem vector of the i^{th} item and the key vector of the j^{th} item, and vice versa, resulting in four cosine similarity indices: $\text{cosine}(stem_i, stem_j)$, $\text{cosine}(key_i, key_j)$, $\text{cosine}(stem_i, key_j)$, and $\text{cosine}(key_i, stem_j)$.

3.3.2 Cosine Indices Based on the Latent Semantic Analysis

Latent Semantic Analysis was performed on the document-term matrix, which applies the Truncated Singular Value Decomposition on the document-term matrix. Semantic concepts were extracted through this dimension reduction process.

Selecting the optimal number of concepts/dimensions retained from LSA has always been an open research issue. A low value of k dimensions is not sufficient to capture the relationships between the terms and documents, whereas a large value induces noise or irrelevant details. Some studies have suggested to select the optimal number of dimensions empirically. Deerwester et al. (1990) examined the prediction performance over varying numbers of dimensions for a collection of 1,033 medical abstracts with 5,823 unique terms. They found that the mean precision more than doubled as the number of dimensions increases from 10 to 100 and peaked at 100. Bradford (2008) explored the effects of varying dimensionality in large collections of up to five million documents and suggested that the optimal number of dimensions ranged from 300 to 500. This study also used a scree plot (Cattell, 1966) to identify where the singular values stabilize. The results indicated that there was relatively little gain when increasing the number of dimensions over 300. The proportion of variance explained by each dimension can also be calculated from the singular values and be used to determine the number of dimensions to retain. Existing literature suggested that a cumulative proportion of 70% to 90% variance explained usually indicates sufficient representation of the original data (Jolliffe, 2002; Cangelosi & Goriely, 2007).

To ensure that the *k* extracted concepts are representative of the document-term matrix, a minimum of 300 dimensions were retained from the Latent Semantic Analysis. The singular values of dimensions were graphed in a scree plot to identify the location where the values

stabilized. The cumulative proportion of variance explained by the dimensions were examined at varying numbers of k to ensure that at least 70% percent of the variance in the document-term matrix was preserved.

Once the optimal number of concepts was selected, LSA produced the loadings of each documents on all of those concepts (document-concept matrix), as well as the loadings of each term on those concepts (concept-term matrix). The extracted concepts which explained the most variance in the data were examined. The first five terms with the highest loadings on each of these concepts were scrutinized to ensure that the LSA extracted meaningful concepts.

Similarly, the four cosine similarity indices for each item pair were calculated based on the document-concept matrix produced from the LSA. Each document vector used for this calculation consisted of the document's scores on each extracted concept rather than on each term.

3.3.3 Cosine Indices Based on the Latent Dirichlet Allocation

The document-term matrix was also analyzed with the LDA to identify the topics within the item bank. The probability distribution over these topics was produced for each document. Deriving the probability distribution involves computing posterior distribution on a large discrete state space because of the sparseness of the document-term matrix. This problem was addressed by using a Gibbs sampling procedure which is easy to implement, requires little memory, and is competitive in speed and performance compared to other approximation methods such as variational Bayes or expectation propagation (Griffiths & Steyvers, 2004). Gibbs sampling is a Markov chain Monte Carlo algorithm widely applicable to calculating a complex posterior distribution. A Markov chain is constructed to converge to the target distribution, and samples are then taken from that Markov chain. In each state of the chain, a set of values are assigned to

variables being sampled, and transitions between states follow a rule. With Gibbs sampling, the next state is reached by sequentially sampling all variables from their distribution when conditioned on the current values of all other variables and the data. To ensure that stationary distribution of the Markov chain was reached, Gibbs sampling was run with a burn-in period of 1,000 iterations.

As described in the previous section 2.3.3 regarding the generative process of the LDA, the number of latent topics K is assumed to be known a priori. Therefore, an empirically determined $alpha$ prior equivalent to the inverse of the number of topics ($\alpha_{prior} = \frac{1}{K}$) is typically used. This will provide a starting point for the LDA with the assumption that only one topic is likely to contribute to any given document. Based on existing literature discussing the appropriate range of number of topics for LDA models, studies have shown that the optimal results can be achieved with as few as 20 topics, and the number of topics typically would not exceed 100 topics even for large text corpus with thousands of documents (Tang et al., 2014; Zhao et al., 2015; Weir, 2019). Zhao et al. (2015) examined the meaningfulness of topics at various specifications of number of topics. This study found that when the number of topics exceeded 100, a larger number of topics was judged to be less meaningful such that the topics were unable to represent a unique and salient theme, compared to LDA models with fewer topics. While there is no one best way to determine the optimal number of topics, this study will use an empirical approach to assess the fit of several models across a range of specified topics. The perplexity of each model will be evaluated at varying number of topics K. Perplexity is a standard measure of how well a probability model predicts a sample and can be used to compare the performance of LDA models (Griffiths & Steyvers, 2004; Zhao et al., 2015). Perplexity is

defined as the reciprocal geometric mean of the term likelihoods in the documents (D) given the model:

$$\text{perplexity}(D) = \exp\left\{-\frac{\sum_{m=1}^{M}\log P(t_m)}{\sum_{m=1}^{M}N_m}\right\} \quad (10)$$

Lower values of perplexity indicate lower misrepresentation of the terms by the topics. The model perplexity was examined over a varying number of topics to determine the most reasonable number of topics, which was then used to determine the prior of the final LDA model. Once the LDA model was fit to the data, a document-by-topic matrix was generated where each row represented the topic distribution of a given document in the item bank.

To validate the meaningfulness of the topics extracted by the LDA, the top five most likely terms associated with each topic were inspected. The substantive meaningfulness of the topics was evaluated by the SMEs as another indication of goodness of fit of the LDA.

The four cosine similarity indices for each item pair were calculated based on the document-topic matrix generated from the LDA. Each document vector used for this calculation consisted of the likelihood of each topic occurring in this document.

3.4 Enemy Item Pair Classification

In the classification stage, the enemy status for each of the item pair was predicted using the logistic regression and Artificial Neural Networks classifiers. The classification dataset included 1,066,530 item pairs (based on the 1,461 multiple-choice items available in the bank), with some enemy relationships pre-determined by SMEs through previous test form review. The enemy status of each item pair, as indicated in the item bank, was the outcome variable in the classification model.

The key predictor variables were the four cosine similarity indices computed from the NLP models. The average length of each item pair was also calculated and included as a

predictor. The total number of lemmas identified in the stem and key of each item was first calculated, and the average number of lemmas between each item pair was computed as the average length. In the previous literature, the item length was shown to be related to the prediction of enemy probability. In addition, two other predictors were calculated based on the item meta data: the content area and the item difficulty level. The content area of each item was available in the dataset. Typically, each item was assigned to the most relevant content area on the test blueprint during the initial item development process, and the categorization into the same content area may be an indicator of between-item similarity. The item difficulty, on the other hand, was calibrated based on pretest data later in the item development process, prior to its inclusion to the operational item pool. The content area information is universally available for test items because the assignment of content area takes place in the initial stage of item development, while the difficulty parameters are only available after the items have successfully undergone the pretesting stage and were calibrated using the data obtained during pretesting. In other words, item difficulty information is specific to operational items. This study utilized an operational item pool, and therefore the absolute difference in the calibrated difficulty level (*b* parameter) of each item pair was also included as a predictor. In the context of adaptive testing, enemy items with large difference in calibrated difficulty parameters may not be as threatening as items with a smaller difference, as it is very unlikely that two items with great difference in difficulty level are administered in the same adaptive test (Lai & Becker, 2010).

3.4.1 Synthetic Minority Over-Sampling for Imbalanced Data

The outcome variable of the classification model was the binary enemy status pre-determined by SMEs. Among the 1,066,530 item pairs, a total of 327 were enemy pairs, and the rest were non-enemies. Given the characteristics of an item bank, the proportion of enemy item

pairs was expected to be very low, and the majority of the item pairs will not be enemies, which leads to an imbalanced dataset in which the majority class (non-enemies) significantly out-numbers the minority class (enemies). Such imbalanced data is commonly seen in many real applications of medical diagnosis, fraud detection, and defect prediction, etc. (Wei et al., 2013; Belarouci & Chikh, 2017; Han et al., 2019; Malhotra & Kamal, 2019). Traditional classification techniques tend to be overwhelmed by the majority class and ignore the minority class, when in many cases, the prediction of minority class is of much more interest. Research found that classifiers tend to provide a severely imbalanced degree of accuracy between the majority and minority classes, with the majority class having close to 100% accuracy and the minority class having an accuracy of 0 to 10% (Woods et al., 1993; Chawla et al., 2002).

Resampling techniques were proposed to counter the challenges presented by imbalanced data through over-sampling the minority class or under-sampling the majority class. However, random over-sampling is prone to overfitting, whereas random under-sampling is also likely to remove observations that are important to classification performance (Komori & Eguchi, 2019). The Synthetic Minority Over-sampling Technique (SMOTE) was developed to improve the over-sampling technique and has been proved to be effective in dealing with class imbalanced problems (Chawla et al., 2002). It generates a certain number of artificial minority class samples based on the similarities between existing minority examples to balance the distribution between the samples of majority and minority class. The process of SMOTE is as follows:

a) For each minority sample $x_i \in D_{minority}$, compute k nearest neighbors with minority class samples according to Euclidean distance;

b) Select a neighbor x_j randomly from the k nearest neighbors of x_i;

c) Compute the difference $dif = x_j - x_i$;

d) A new minority sample is generated between x_i and x_j according to

$$x_{new} = x_i + \delta \cdot dif \quad (11)$$

where $\delta \in [0, 1]$ is a random parameter used to control the position of the new generated sample.

Since the distribution of enemy and non-enemy item pairs was highly imbalanced in the study data, the SMOTE technique was applied to increase the accuracy of the minority class prediction. Synthetic enemy item pairs were generated to balance the distribution of enemy and non-enemy item pairs. The data were divided into 80% training data and 20% test data. The SMOTE technique was applied on the training data. The classifier was trained on the training data, and the fitted model was applied on the test data to predict the enemy status. It is worth noting that none of the previous studies have employed the SMOTE technique to address the data imbalance issue. When the SMOTE was applied in this study to generate a balanced classification dataset, the variance of the variables was inevitably altered. As a result, the predicted probabilities appeared different from the results of the previous studies. Therefore, the optimal range of classification cutoffs needed to be re-examined and determined empirically based on the classification results of this study.

3.4.2 Logistic Regression Classification

The four cosine similarity indices and the average item length derived from the NLP models, along with the difference in difficulty parameters and content overlap indicator, were entered as predictors for the classifiers to predict the enemy relationship between each item pair.

The parameters of logistic regression were estimated using maximum likelihood estimation. The estimation procedure begins with an expression for the likelihood of observing the enemy status ($y_i = 1$) and non-enemy status ($y_i = 0$) in the data:

$$\text{Likelihood Function} = \prod_{i=1}^{n} \{P(x_i)^{y_i} \cdot (1 - P(x_i))^{1-y_i}\} \quad (12)$$

where $p(x_i)$ is the likelihood of observing enemy status and $1 - p(x_i)$ is the likelihood of observing non-enemy status as illustrated in Eq. (6) and Eq. (7). The product of the likelihood of observing the enemy status of each item pair becomes the joint likelihood of observing the dataset. The goal of maximum likelihood estimation is to identify a set of β values that maximize the likelihood of observing the dataset, which involves taking the derivative of the likelihood function and setting it to zero. This process can be simplified by taking the derivative of the log likelihood, since log is a monotonically increasing function.

Maximizing the log likelihood can sometimes be an unsolvable problem in closed form. An iterative procedure could then be used to fit a new set of parameters through each iteration. This process continues until the increase in the log likelihood function from choosing new parameters becomes so small that little gain comes from continuing any further.

The logistic regression was applied on the post-SMOTE training dataset. Once the MLE estimation converged through the iterations, the logistic regression classifier produced a set of parameters for each predictor, and the probability of observing the enemy relationship for each item pair was generated. The probability values were used to produce the classification results for further evaluation.

3.4.3 Artificial Neural Network Classification

The data were also trained on the ANN classifier. The structure of an ANN consists of several neurons (nodes) arranged in a layer-by-layer network, and the neurons in each layer have connections (weights) from the neurons in the previous layer. The predictors were entered as neurons on the input layer of the neural network, and the weighted connections pass forward the information through the hidden layers where the data were processed.

3.4.3.1 Number of Hidden Layers and Neurons

The specification of hidden layers and neurons has been one of the major challenges of neural networks. According to the universal approximation theorem, any continuous function can be uniformly approximated by an ANN with a single hidden layer containing a finite number of neurons (Cybenko, 1989; Funahashi, 1989), and an ANN with two hidden layers can represent any arbitrary decision boundary in classification and approximate any smooth mapping to any accuracy (Hornik et al., 1989; Hornik, 1991). Therefore, neural networks with one and occasionally two hidden layers are widely used to model complex problems. The choice of number of hidden neurons also has an impact on the performance of the network. If an inadequate number of neurons is used, the network will be unable to model complex problems. Excessive hidden neurons, on the other hand, will result in over fitting, causing the neural network to over-estimate the complexity of the problem (Ke & Liu, 2008). Some rule-of-thumb methods were proposed for determining the sufficient number of hidden neurons (Heaton, 2008): a) the number of hidden neurons should be between the number of input neurons and output neurons; b) the number of hidden neurons should be 2/3 the size of the input neurons plus the size of the output neurons; c) the number of hidden neurons should be less than twice the size of the input neurons.

In the previous studies of enemy item detection using ANN classifier, Lai and Becker (2010) used one hidden layer of 19 neurons for approximately 30 input neurons to derive the binary output of enemy status; Peng et al. (2019) assigned five neurons in the first hidden layer and three in the second hidden layer for a network of eight input neurons and a binary output neuron.

Based on the literature, the current study employed two hidden layers in the ANN structure, which were sufficient to approximate any arbitrary classification decision to any accuracy. The seven predictors entered the input layer. The ANN model was configured to have four neurons in the first hidden layer and three neurons in the second hidden layer, as illustrated in Figure 5. The connecting weights ran from one layer to the next and carried the output of each neuron to the neurons in the next layer. Eventually, the network predicted the probability of enemy status for each item pair.

3.4.3.2 Activation Function for Neurons

As previously illustrated in Eq. (9), the output of a neuron is governed by an activation function $\varphi_i(\cdot)$. This activation function controls the amplitude of the neuron's output into a certain range. Two popular activation functions are the step function and the sigmoid function (Haykin, 1999; Du & Swamy, 2019). Let x_i denote the original unscaled output of a neuron i:

$$x_i = \sum_{j=1}^{n^i} w_j^i z_j^i + b^i. \tag{13}$$

The step function determines that

$$\varphi(x_i) = \begin{cases} 1 & \text{if } x_i \geq 0, \\ 0 & \text{if } x_i < 0. \end{cases} \tag{14}$$

This implies that the step function is a hard limiter that turns the neuron "on" or "off". The sigmoid activation function, on the other hand, is a logistic function:

$$\varphi(x_i) = \frac{1}{1+e^{-\beta x_i}}, \tag{15}$$

where β is a gain, typically selected as unity, and is used to control the steepness of the activation function (Du & Swamy, 2019). The sigmoid activation function controls the output rate of the neuron.

In this study, the step function was used as the activation function for neurons in the hidden layers to eliminate weak connections, and the sigmoid function was used for the final neuron in the output layer to produce the probability of enemy status.

3.4.3.3 Optimization of Artificial Neural Network Classification

The goal of ANN is to find a set of weights that can minimize the cost function; that is, the Mean Square Error (MSE) of the neural network:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2, \tag{16}$$

where N is the total sample size. In order to minimize the cost function, a gradient descent procedure was used to locate the minimum point of the cost function. This method calculates the slope of the function and then adjusts the weights and biases accordingly to achieve a lower loss through iterations. Each iteration is an attempt to move closer to the minimum of the cost function. This is analogous to descending an object on a surface in the weighted space. At any point of the descent, one can calculate the slope and take steps towards the descending direction of the slope. Each step moves towards the bottom of the surface until a stable minimum point is reached. To avoid finding the local minimum of the cost function, the gradient descent procedure was conducted multiple times in order to find the global minimum. In machine learning, each of these gradient descent procedures is termed an *epoch*. In this study, the ANN classifier was run for 20,000 epochs until the MSE was below 0.001.

3.5 Model Evaluation

The classification results will be evaluated and compared across models at various probability thresholds. Based on the literature, the optimal classification results of enemy status tend to occur at a lower probability threshold (Becker & Kao, 2009; Li et al., 2012; Peng et al., 2018; Peng et al., 2019). Since this study applied the SMOTE technique to achieve a balanced

classification dataset that was not employed in any of the previous studies, the probability cutoffs suggested by previous literature were not applicable to this study. The probability distribution was re-examined, and the classification results were evaluated based on the following metrics.

Given a probability cutoff for the classification (i.e. $\geq .50$ as enemy, and $< .50$ as non-enemy), the number of item pairs correctly classified and/or misclassified by the model for each enemy class can be calculated and presented in a confusion matrix showing:

TABLE I
EXAMPLE CONFUSION MATRIX

	Actual Status in Item Pool	
Predicted Status	Enemy	Non-Enemy
Enemy	True Positive (TP)	False Positive (FP)
Non-Enemy	False Negative (FN)	True Negative (TN)

A series of classification performance metrics can be calculated based on this matrix. Below are the formulas for four commonly used classification metrics:

$$\text{Recall/Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \tag{17}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{18}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \tag{19}$$

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{20}$$

Another performance metric closely tied to the precision and recall rates is the F_1 score. F_1 score is the weighted average of precision and recall:

$$F_1 \text{ Score} = \left(\frac{\text{Recall}^{-1}+\text{Precision}^{-1}}{2}\right)^{-1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \tag{21}$$

An F_1 score reaches its best value at 1 and worst score at 0. An F_1 score is usually used when imbalanced class distribution exists. Although the SMOTE technique was employed to adjust for the class imbalance for the training dataset, a large number of cases are still expected to be True Negatives when the prediction is made on the test dataset. In this study, an F_1 score is a better reference for model evaluation.

To determine a reasonable range of probability cutoffs to compare across, a Receiver Operator Characteristic (ROC) curve analysis was performed. An ROC curve is generated by plotting the False Positive Rate (FPR) against the True Positive Rate (TPR), also known as the *recall rate* and *sensitivity rate*, at various threshold settings.

$$\text{FPR} = \frac{\text{FP}}{\text{FP+TN}} \tag{22}$$

$$\text{TPR/Recall/Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \tag{23}$$

The FPR is equivalent to $1 - Specificity$. Therefore, the ROC curve illustrates the tradeoff between the sensitivity and specificity. In this study, the ROC curve was conducted as an initial search for the range within which the optimal probability cutoff was located.

In addition to the ROC curve analysis, this study also examined the Precision-Recall Curve (PRC). A Precision-Recall Curve plots the recall rate against the precision rate and visualizes the tradeoff between precision and recall rates for every possible cutoff. The Precision-Recall Curve is informative for this study since the calculation of both recall and precision rates is based on True Positive, True Negative, and False Positive cases.

The area under the curve (AUC) for both ROC and PRC was also examined. The AUC evaluates the global performance of a classification model in terms of the metrics plotted in the ROC or the PRC over the full range of possible cutoff values. The AUC measure can be used to compare across competing classification models. An AUC of .5 represents a classification test

with little discriminating ability, while an AUC of 1.0 represents a test with perfect discrimination.

Based on the results from the ROC and PRC analyses, four probability cutoffs within the optimal range were determined for further examination. The identification of the optimal probability cutoff typically required a simultaneous assessment of sensitivity, precision, specificity, accuracy, and F_1 score of the classification models. Depending on the goal of the study, the evaluation may focus on different performance metrics. In this study, the goal was to identify as many enemy item pairs as possible, while keeping the false negative cases at a manageable number for further SME review. Therefore, this study focused on performance metrics based on the True Positive cases and False Positive cases. Given that the proportion of enemy item pairs was low and the majority of the item pairs were not enemies, accuracy and specificity were expected to be high regardless of the classifier, due to the large number of true negative cases. On the other hand, sensitivity (the TPR), precision, and the F_1 score were considered more important indicators of fit in the context of this study.

3.6 SME Review and Retraining of Classification Models

Since the enemy item pairs flagged in this dataset were only identified during the test assembly process by SMEs, it is likely that some enemy pairs were never reviewed and identified in the dataset. That is, because enemy item pairs were only labeled as enemies if they had been assessed by SMEs and unlabeled if they had not, the full truth about the enemy relationships among the items is unknown. As a result, the initial classification results are likely to be imprecise, and some of the False Positive item pairs could indeed be true enemies.

After training the classification model, the False Positive item pairs that were predicted to have a high probability of being enemies were sent to SME review. A list of these False Positive

item pairs, sorted by the predicted enemy probability from high to low, were presented to two SMEs with item review experience. As the volume of False Positive cases tends to be large, it is unrealistic to review all the False Positive item pairs. A stopping rule was thereby employed to control the workload of the review. Within the descending order of enemy probability, the False Positive item pairs were grouped into sets of 20 item pairs. The SMEs conducted the review from higher probability sets to lower probability sets and determined true enemy relationships for each item pairs. The SMEs were asked to stop the reviewing process when they encountered less than 10% (only one) true enemy pair(s) within a 20-item pair set.

The two SMEs who conducted the enemy item review both have a background in nursing and have over five years of experience working in the test development for the National Council of State Boards of Nursing. They were trained to look for characteristics of enemy item pairs, such as degree of content overlap and presence of clueing. They both have over four years of experience screening for enemy item pairs within the operational item pools of national nursing licensure tests. The SMEs will conduct the enemy item review independently and compare their enemy relationship decisions. When disagreement of enemy status for certain item pairs arises, a discussion will take place between the SMEs to finalize the enemy status for these item pairs.

After the SME review, the enemy status for those newly confirmed enemy item pairs was updated in the item bank, as well as in the classification dataset. The updated dataset reflected more accurate enemy relationships among item pairs. The models were retrained on the updated dataset, and the classification results were re-evaluated. This iterative process provided insight as to whether the proposed automatic detection process can improve the enemy relationship tagging and monitoring in the item bank.

3.7 Summary of Methods and Research Questions

To address the research questions, this study used a crossed 3 (NLP) × 2 (Classifier) × 4 (Probability Cutoff) design with a total of 24 experimental conditions to evaluate the performance and accuracy of each combination of methods for detecting enemy item pairs within an operational item bank. The methods employed in this study were summarized and linked to each research question below.

At the first stage of the study, the three NLP techniques were applied to the document-term matrix. The concepts and topics within the item bank were extracted from the two topic modeling NLP approaches–the Latent Semantic Analysis and the Latent Dirichlet Allocation, respectively. The optimal number of concepts/topics was evaluated based on the appropriate statistical criteria described above. In addition, SMEs reviewed the associated terms with the top concepts/topics to ensure these topic modeling approaches captured substantially meaningful concepts/topics. Each NLP technique produced the cosine similarity indices for all the item pairs.

At the second stage, the similarity indices, along with other item meta data predictors, were entered into the logistic regression classifier and the Artificial Neural Network classifier to predict the probability of enemy status for each item pair. The classification models were trained using the training data and then made prediction on the test data. The classification results were first evaluated with the ROC curve analysis, which provided an initial indication on the range of optimal probability cutoffs. Four probability cutoffs were chosen within the range for further examination. The classification performance metrics were evaluated across the models.

At the third stage, the most probable False Positive enemy pairs underwent SME review in order to detect additional enemy relationships that were not previously flagged in the item bank, based on which the enemy status of the item pairs was updated. The models were re-

trained based on the updated data, and the classification results under each condition were re-evaluated.

Research question a): Do natural language processing techniques adequately capture item similarity? The distributions of the cosine similarity indices from the three natural language processing techniques were examined across the enemy and non-enemy item pairs. Furthermore, the classification performance of the three NLP approaches were compared across the classifiers and probability cutoffs. The meaningfulness of the concepts and topics extracted served as additional evidence of the ability of the NLP techniques in capturing conceptual similarity.

Research question b): Compared to a logistic regression classifier, does the Artificial Neural Network classifier improve the accuracy of classifying enemy item pairs? This study also compared the classification results from the logistic regression classifier and the Artificial Neural Network classifier across the probability cutoffs to address the research question of whether the ANN classifier improves the accuracy of enemy relationship classification over the traditional logistic regression classifier.

Research question c): What probability cutoff is considered optimal for classifying a sufficient number of existing enemy item pairs, while keeping the number of falsely classified item pairs manageable? The ROC curve analysis revealed a reasonable range of probability cutoff values, based on which four probability cutoffs were chosen for further evaluation. Numerous classification performance metrics were evaluated at each of these cutoffs to determine the optimal cutoff point. The evaluation criteria of the optimal cutoff focused on increasing the recall rate of enemy item pairs and keeping the number of False Positive item pairs manageable for SME review.

Research question d): Does the automatic enemy item detection procedure help reveal more true enemy items previously not identified in the item pool? Does retraining the model, using the input from SME review of false positive item pairs, help improve the accuracy of classifying enemy item pairs? Two SMEs reviewed the False Positive item pairs classified by the automatic detection procedure and identified additional true enemy item pairs. The results of the SME review were summarized to inform whether the automatic process can help catch more enemy items previous not identified in the item pool. After the SMEs' review, the enemy status was updated, and the analysis was re-run. The updated classification results under each condition were re-examined in order to provide insight into whether this iterative process helps improve the accuracy of enemy item pair classification.

4. RESULTS

This chapter describes the results found in the process of 1) item data transformation, 2) the application of the three NLP methods, 3) classification of item pairs, 4) SME review of False Positive item pairs, and 4) the re-training and re-evaluation of models.

4.1 Item Data Transformation

The item data consisted of 1,461 multiple-choice items from the operational item bank of a national licensure examination for nurse aides. The data included the text of the stem, key, and distractors of each item. The enemy associations between items, if previously identified through item review and test assembly process, were also recorded in the data. Also available were the content area code of each item and its difficulty parameter, calibrated from the IRT model.

The sample item bank covered 17 specific content areas laid out in the test blueprint. A test blueprint reflects the test objectives by specifying the proportion of each content area covered in the test. Items are selected from the item bank for test form construction based on the test blueprint. Therefore, the item bank needs to maintain enough items to ensure that multiple test forms meeting the test blueprint specifications can be constructed. Table 2 illustrates the content area proportions specified in the test blueprint and the distribution of the content areas within the actual item bank used in this study. The table lists the names of content areas 1-17 in the item data and their test blueprint target proportions in parentheses. The content areas were summarized into higher-level content domains in the test blueprint. The *Physical Care Skills* domain was assigned a higher weight in the test and was broken down into three sub-domains and 11 content areas. This domain differed from the other domains in that the test blueprint

targets were specified at the three sub-domain levels rather than at the content area level. The two columns on the

TABLE II

BLUEPRINT SPECIFICATIONS AND CONTENT AREA DISTRIBUTION

Content Area (Blueprint %)	Item Bank N	Item Bank %
I. Physical Care Skills		
A. Activities of Daily Living (14%)		**15.26**
1. Hygiene	75	5.13
2. Dressing and Grooming	23	1.57
3. Nutrition and Hydration	50	3.42
4. Elimination	34	2.33
5. Rest/Sleep/Comfort	41	2.81
B. Basic Nursing Skills (39%)		**37.30**
6. Infection Control	92	6.30
7. Safety / Emergency	131	8.97
8. Therapeutic and Technical Procedures	166	11.36
9. Data Collection and Reporting	156	10.68
C. Restorative Skills (8%)		**8.42**
10. Prevention	39	2.67
11. Self Care / Independence	84	5.75
II. Psychosocial Care Skills		
12. Emotional and Mental Health Needs (11%)	152	10.40
13. Spiritual and Cultural Needs (2%)	49	3.35
III. Role of the Nurse Aide		
14. Communication (8%)	131	8.97
15. Client Rights (7%)	96	6.57
16. Legal and Ethical Behavior (3%)	38	2.60
17. Member of the Health Care Team (8%)	104	7.12
Total	1,461	100.00

Note. Values in bold are aggregated percentage of corresponding content domains.

right show the number of items from each content area and the corresponding proportions in the item bank data. While the item bank does not strictly match the proportions allocated in the test blueprints, the table indicates that the item bank data generally aligns with the test blueprint specifications.

The item bank provided the initial text corpus for this study, which included 1,461 item stems and 1,461 item keys. These text documents were processed using the NLP data transformation technique to derive the document-term matrix. Each document was first parsed into distinct terms and stripped of punctuation and case. Next, stop words were removed from the documents. This study used a default list of stop words offered by the NLTK package in Python 3.7. This list of stop words includes 221 English words with little lexical content; their presence does not add to the semantic meaning of the document. Upon analyzing the items, eight item bank specific words and phrases were identified as additional stop words. The words *nurse aide* and *client* appear in most of the items. As a result, they do not contribute to the distinction between items. Moreover, phrases such as *which of the following*, *what is the best response,* and *what is the most appropriate* are commonly used in the item stems. During the item development process, editorial review was conducted to standardize the word usage across items, and these templates were employed in the item stem to prompt the response options. These words and phrases were also removed from the document so that they do not artificially inflate the similarity between documents using the same editorial templates.

Table 3 summarizes the types of stop words employed in this study. Both default stop words in the English language as well as item bank specific stop words are illustrated with

examples in the table. The most common type of stop words (28%) were prepositions, conjunctions, and/or adverbs. Compounds such as pronoun and verb (e.g. *you're*), verb and negation (e.g. *isn't*), and auxiliary and negation (e.g. *shouldn't*) were also typical stop words (24%). Some other common words in the English language, such as *one*, *many*, and *never,* were also included in the default stop words (18%). A total of 229 stop words were removed from the documents.

TABLE III

SUMMARY OF DEFAULT STOP WORDS AND ITEM BANK SPECIFIC STOP WORDS

Type	Examples	N	%
Default			
Articles	a, the	3	1.31
Auxiliaries	would, could	4	1.75
Common Words	one, many, never	42	18.34
Compounds	she's, haven't	55	24.02
Prepositions, Conjunctions, Adverbs*	between, but, again	63	27.51
Pronouns	you, himself	38	16.59
Verbs	have, do	16	6.99
Item Bank Specific			
Common Words within Item Bank	client, nurse aide, he/she, and/or	4	1.75
Templates	which of the following, best response	4	1.75
Total		229	100.00

Note. The overlap among prepositions, conjunctions, and adverbs is so common that separate classification is unnecessary.

After the stop words removal, lemmatization was performed to convert each word into its dictionary form. As a result, a total of 2,147 unique lemmas were identified across the item bank.

On average, an item stem document after the lemmatization had nine lemmas, and an item key document had three lemmas. The occurrences of all unique terms for each document were mapped onto the document-term matrix, resulting in a 2922×2147 matrix in which each row represented an item stem or an item key, and each column represented a unique term/lemma in the item bank. TF-IDF transformation was applied on the document-term matrix to normalize the raw frequency based on each term's occurrence across the item bank and its occurrence within each document.

4.2 Results from NLP techniques

4.2.1 Vector Space Model

Computing similarity indices between items was straightforward for the Vector Space Model. For any given item pair i and j, the program first identified their corresponding stem and key vectors in the document-term matrix. Four cosine similarity indices were then calculated based on different components of the item pair: $\text{cosine}(stem_i, stem_j)$, $\text{cosine}(key_i, key_j)$, $\text{cosine}(stem_i, key_j)$, and $\text{cosine}(key_i, stem_j)$.

Table 4 shows the descriptive statistics of the four cosine indices across all item pairs. There were a total of 1,066,530 item pairs between 1,461 items, among which 327 were flagged as enemy item pairs and 1,066,203 as non-enemy item pairs in the item bank. The cosine indices were grouped by the enemy status of item pairs in Table 4. The cosine indices for the VSM approach were computed from normalized term frequencies. As a result, the cosine indices all range from 0 to 1. On average, the cosine indices were higher within enemy item pairs than within non-enemy pairs. The average cosine between stems and the cosine between keys were both above .280 for enemy item pairs, with standard deviations of.266 and 0.334. In comparison, both the cosine between stem_i and key_j (M=.075, $S.D.$=.169) and the cosine between key_i and

$stem_j$ (*M*=.101, *S.D.*=.192) were roughly one third of the value, which was not surprising, as clueing is not a common type of enemy association.

The cosine indices were consistently low within non-enemy item pairs, with average values ranging from .003 to .007. The standard deviations across the cosines were also similar within non-enemy pairs, with a range between .026 and .036.

An independent *t*-test assuming unequal variances was conducted between the enemy and non-enemy groups for each of the cosine indices. The results showed that the enemy item pairs have significantly higher values on all four of the cosine similarity indices compared to the non-enemy item pairs with a *p*-value less than .001.

TABLE IV

DESCRIPTIVE STATISTICS OF COSINE SIMILARITY INDICES FROM THE VSM

Similarity Index	*N*	Mean	*S.D.*	Min.	Max.
Enemy					
$\text{cosine}(\text{stem}_i, \text{stem}_j)$	327	.320	.266	.000	1.000
$\text{cosine}(\text{key}_i, \text{key}_j)$	327	.280	.334	.000	1.000
$\text{cosine}(\text{stem}_i, \text{key}_j)$	327	.075	.169	.000	1.000
$\text{cosine}(\text{key}_i, \text{stem}_j)$	327	.101	.192	.000	1.000
Non-Enemy					
$\text{cosine}(\text{stem}_i, \text{stem}_j)$	1,066,203	.007	.035	.000	1.000
$\text{cosine}(\text{key}_i, \text{key}_j)$	1,066,203	.004	.036	.000	1.000
$\text{cosine}(\text{stem}_i, \text{key}_j)$	1,066,203	.003	.027	.000	1.000
$\text{cosine}(\text{key}_i, \text{stem}_j)$	1,066,203	.003	.026	.000	1.000

4.2.2 Latent Semantic Analysis

4.2.2.1 Determining the Number of Concepts

Model selection was first performed to determine the optimal number of concepts to retain for the Latent Semantic Analysis. As the literature suggested, a large text corpus typically contains over 300 distinct concepts. Therefore, this study fit the LSA model at varying number of concepts ranging from 300 to 1000.

Figure 6 plots the proportion of variance explained by the LSA model and the singular value associated with each concept against the number of concepts. The proportion of variance explained is a cumulative measure of how much variance in the data is accounted for by the retained concepts. As seen in the figure, the proportion of variance explained increased steeply between 0 and 200 concepts but slows down after 300 concepts. The percentage of variance explained reached 70% at 455 concepts and gradually increased to 90% at 955 concepts.

The singular value associated with each concept is also plotted on the dash line. These singular value estimates were obtained from the singular value decomposition of LSA, indicating how important each concept is. These values were sorted in a descending order, and each value corresponds to the singular value estimate of the n^{th} concept on the x-axis. The singular value started at a maximum of 0.0067 and dropped rapidly to less than 15% of the maximum (0.0009) at the 300^{th} concept. The value stabilized after the 450^{th} concept, after which point each additional concept added little gain to the model.

Existing literature suggested that a cumulative proportion of 70% to 90% variance explained usually indicates sufficient representation of the original data (Jolliffe, 2002; Cangelosi & Goriely, 2007). The above observations provided reasonable justification to select

455 as the optimal number of concepts to retain from the LSA model, at which point 70% of the variance was explained, and the singular value stabilized.

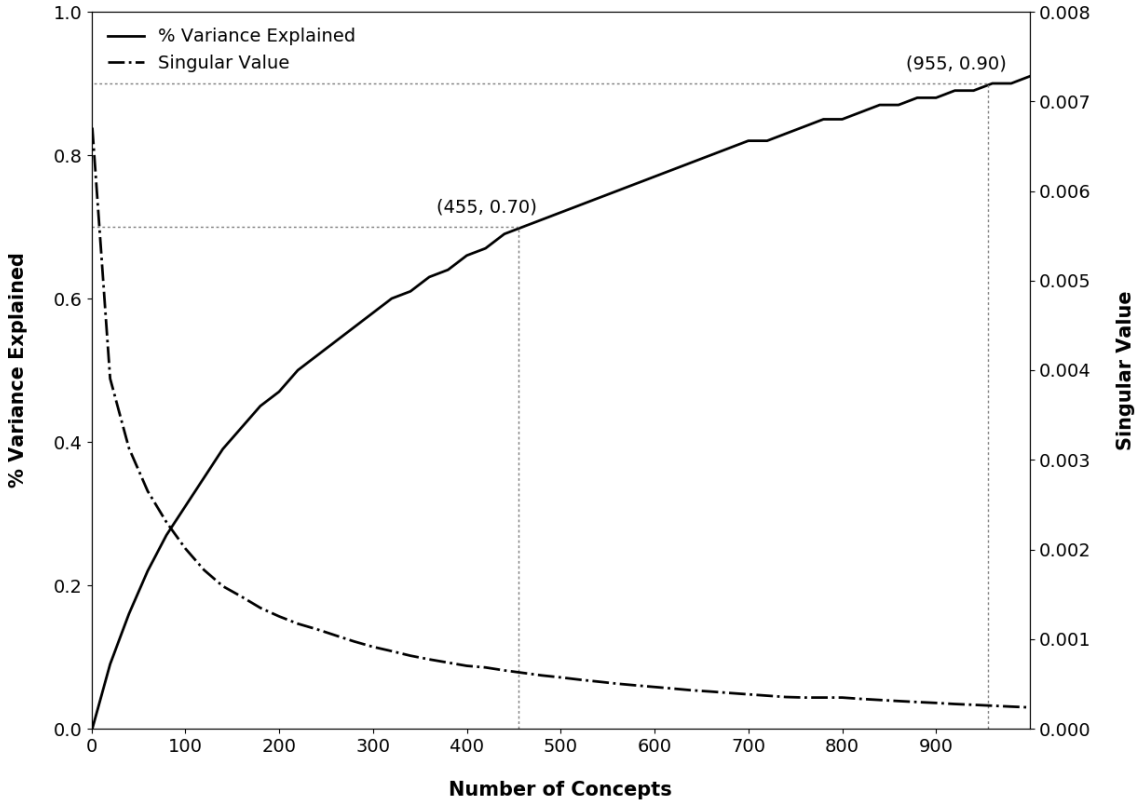

Figure 6 Latent Semantic Analysis Model Selection

4.2.2.2 Fitting the Final LSA Model

Once the optimal number of concepts was determined at 455, a final LSA model was applied to the document-term matrix. The document-term matrix was decomposed into the product of three matrices in Eq. (3). The left matrix (U) is a document-concept matrix based on which the cosine similarity indices between documents can be calculated. The middle singular value matrix (Σ) indicates the importance of each extracted concept and can be used to calculate

the percentage variance explained by each concept. The right matrix (V^T) is a concept-term matrix that records the associations, or loadings, of each term on the concepts. Table 5 lists the top 10 concepts extracted from the LSA model and the top five terms associated with each concept. This table was produced by first obtaining the top 10 concepts with highest singular values and then sorting the terms by their loadings in descending order within each concept. The SMEs reviewed these concepts and suggested tentative labels for each concept based on the associated terms. The suggested labels were attached with each concept in this table.

As shown in Table 5, the extracted concepts appear to be substantively meaningful and conceptually distinct from each other. For example, the first concept explaining the largest proportion of the variance pertains to routine practices encompassing the nurse aide profession, and the associated terms are commonly observed in most test items. The second concept taps into the nurse aide's responsibilities of abnormality monitoring and incident reporting, which is one of nurse aide's main responsibilities. Certain concepts target specific skills, such as assisting a patient to bed, providing emotional support, and monitoring a patient's intake and output, as would be expected in a nurse aide licensure examination. These concepts also correspond to various content areas specified in the test blueprint in Table 2.

TABLE V

TOP CONCEPTS AND ASSOCIATED TERMS FROM

THE LATENT SEMANTIC ANALYSIS

Concept 1 (0.78%) Routine Work		**Concept 2 (0.67%)** Report of Abnormality		**Concept 3 (0.69%)** Bed Safety		**Concept 4 (0.56%)** Emergency		**Concept 5 (0.57%)** Emotional Support	
report	.51	notify	.48	bed	.73	emergency	.49	talk	.65
charge	.50	incident	.42	position	.25	call	.40	encourage	.19
care	.43	abnormal	.14	move	.14	light	.21	feeling	.07
provide	.18	finding	.06	head	.13	ambulate	.18	listen	.07
plan	.17	observation	.04	linen	.12	activity	.17	provide	.06
Concept 6 (0.55%) Family Inquiry		**Concept 7 (0.51%)** Infection Prevention		**Concept 8 (0.49%)** Standard Precaution		**Concept 9 (0.48%)** Monitoring Intake		**Concept 10 (0.45%)** Catheter care	
ask	.60	hand	.93	glove	.40	intake	.48	change	.54
family	.29	wash	.21	wear	.31	fluid	.25	bag	.43
speak	.12	glove	.14	standard	.28	food	.25	bladder	.22
member	.10	prevent	.11	precaution	.19	record	.21	urinary	.21
need	.09	infection	.09	universal	.18	measure	.19	drainage	.15

Note. Values in the parentheses are percentages of variance explained by each concept. The values associated with the terms are concept-term loadings.

The four cosine indices were calculated for each pair of documents based on the document-concept matrix (U) from the Latent Semantic Analysis. Table 6 summarizes the cosine indices computed from the LSA within both enemy non-enemy item pairs. Some negative values were observed, indicating that the item pair conveyed distinct or opposite concepts. The cosine values ranged from $-.079$ to 1.000 within the enemy item pairs. Compared to the results from the VSM model, the four cosine values from the LSA model were generally higher on average. The differences in average cosine values were more pronounced within enemy item pairs, with the largest difference being .153 for enemy pairs and .005 for non-enemy pairs. The standard deviations were slightly higher compared to the results from VSM models, with a range of .233 to .380 within enemy pairs and .046 to .062 within non-enemy pairs. The result from the independent *t*-tests also showed that the enemy item pairs have significantly higher values on all cosine indices than the non-enemy item pairs with a *p*-value less than .001.

TABLE VI

DESCRIPTIVE STATISTICS OF COSINE SIMILARITY INDICES FROM THE LSA

Similarity Index	*N*	Mean	*S.D.*	Min.	Max.
Enemy					
cosine$(stem_i, stem_j)$	327	.473	.318	$-.036$	1.000
cosine(key_i, key_j)	327	.366	.380	$-.079$	1.000
cosine$(stem_i, key_j)$	327	.115	.233	$-.054$	1.000
cosine$(key_i, stem_j)$	327	.149	.254	$-.057$	1.000
Non-Enemy					
cosine$(stem_i, stem_j)$	1,066,203	.012	.057	$-.258$	1.000
cosine(key_i, key_j)	1,066,203	.006	.062	$-.993$	1.000
cosine$(stem_i, key_j)$	1,066,203	.006	.047	$-.989$	1.000
cosine$(key_i, stem_j)$	1,066,203	.006	.046	$-.988$	1.000

4.2.3 Latent Dirichlet Allocation

4.2.3.1 Determining the Number of Topic

The fitting of the LDA model required that the number of latent topics be specified beforehand. To determine the optimal number of topics, this study examined the perplexity of various LDA models with the number of topics ranging from two to 160. To train the model, Gibbs sampling was used with a burn-in period of 1,000 iterations. The perplexity of the LDA model was plotted against the number of topics in Figure 7.

The perplexity reflects how well the data are represented by the probability model, and a lower value of perplexity indicates less misrepresentation by the specified number of topics. The dash line in Figure 7 shows that the perplexity was perpetually low (less than 0.0004) across the number of topics. However, the size of perplexity decreased about 90% as the number of topics increased to 20. Due to the small value of perplexity score, the log perplexity was also plotted on the solid line to better illustrate the change of perplexity with the number of topics. As expected, the log perplexity echoed the same decreasing trend as the number of topics increased. In addition, several peaks were observed on the log perplexity line, indicating that the model fit became worse and bounced back at some points. It was not until the topic number reached 75 that the trend of perplexity score stabilized and began monotonically decreasing thereafter. Although the perplexity score continued to decline after 75 topics, I decided against settling on a higher number of topics because previous literature has found many topics to be less meaningful when the number of topics exceeded 100 (Tang et al., 2014; Zhao et al., 2015; Weir, 2019). Therefore, I considered setting the number of latent topics at 75 to be a reasonable starting point.

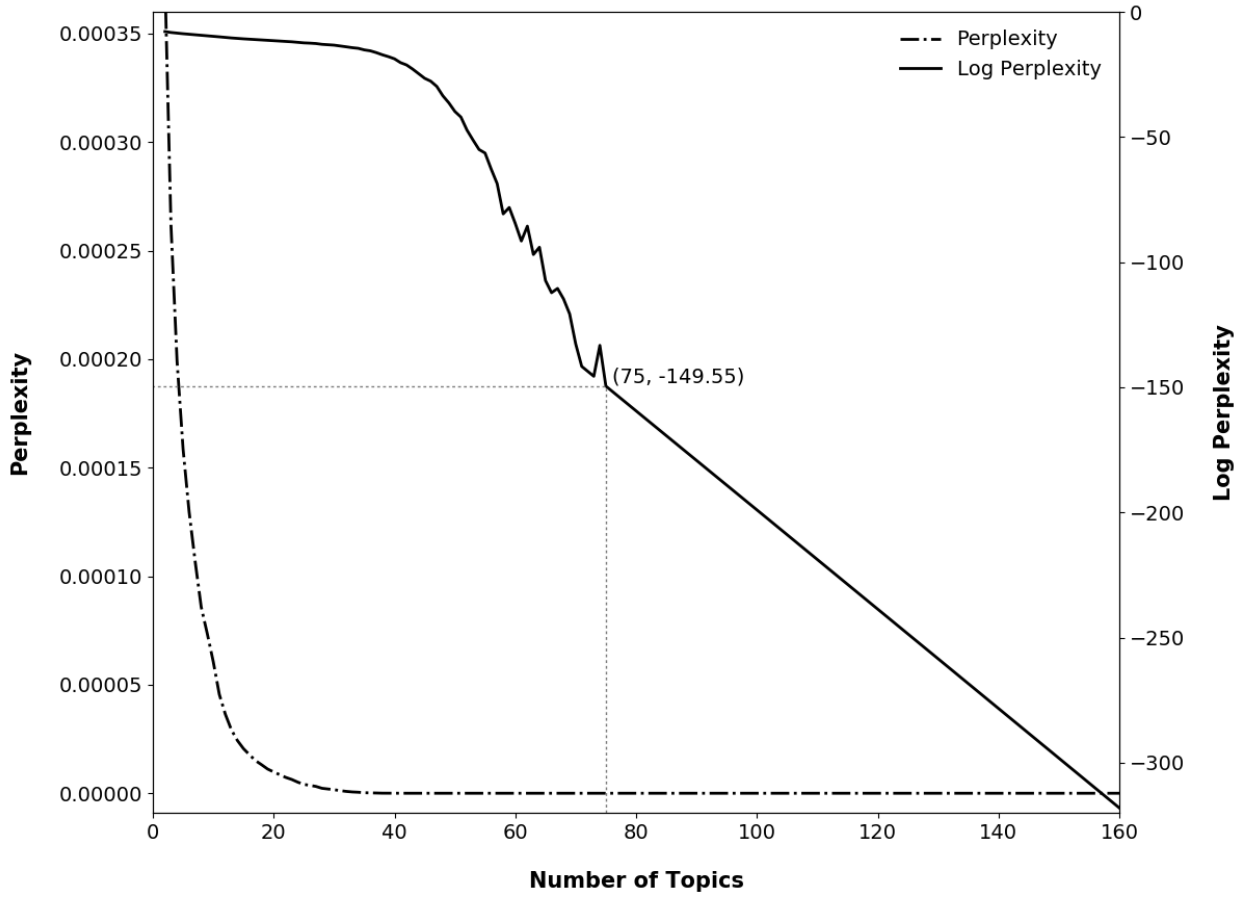

Figure 7 Latent Dirichlet Allocation Model Selection

4.2.3.2 Fitting the Final LDA Model

Having decided on 75 topics, a final LDA was applied to the document-term matrix. An *alpha* prior equivalent to the inverse of the number of topics ($\alpha_{prior} = \frac{1}{K}$) was specified. As in the prior step, a Gibbs sampling approach with a burn-in of 1,000 iterations was used to fit the model.

The final LDA model produced the following results. The *alpha* was estimated to be .076. The final estimation of *alpha*, multiplied by the specified number of topics (*K*), provided an estimate of the number of topics contributing to a document—in this case, about six topics.

A term-topic matrix was also generated from the LDA model, which provided the topic probability distributions over terms. Table 7 illustrates the 10 most probable topics across the documents and the distribution of the five most likely terms within each of these topics. The values in the parentheses show the marginal distribution of these topics in the sample item bank, which ranged from .016 to .024. SMEs were also asked to review these topics and provide tentative labels for each of the topics. According to the SMEs' feedback, the topics from the LDA were judged to be more general compared to those produced by the LSA. For some of these topics, the SMEs encountered more difficulty in distinguishing and labeling the topics. For example, both topic 5 and topic 6 pertain to various care routines performed by nurse aides and the standard precautions to be taken with these routines, such as skin care, body alignment, choke prevention, and putting on protective equipment. The SMEs expressed a higher level of uncertainty in providing labels for three of the topics (indicated by the asterisks). As mentioned above, topic 5 and topic 6 seemed to fall under the same general theme and, therefore, generating discriminating labels became difficult. Moreover, the SMEs were uncertain that "Emotional Support" was a proper label for topic 7, because some terms under this topic (i.e. *begin, collect*) seemed to detract from this theme.

TABLE VII

MOST PROBABLE TOPICS AND ASSOCIATED TERMS FROM

THE LATENT DIRICHELET ALLOCATION

Topic 1 (.024) Main Responsibilities		**Topic 2 (.019)** Incident Report		**Topic 3 (.018)** Use of Call Light		**Topic 4 (.017)** Pain Management		**Topic 5 (.017)** Care Routine*	
charge	.18	immediately	.10	call	.11	plan	.11	skin	.08
report	.12	activity	.10	light	.08	sore	.06	personal	.06
restraint	.06	tell	.06	bowel	.04	face	.06	communicate	.05
speak	.05	supervisor	.05	blood	.03	prevent	.05	equipment	.04
request	.04	dementia	.04	cradle	.03	pressure	.05	protective	.04
Topic 6 (.017) Care Routine*		**Topic 7 (.017)** Emotional Support*		**Topic 8 (.016)** Manner and Patience		**Topic 9 (.016)** Task Performance		**Topic 10 (.016)** Bed Care	
choke	.07	begin	.08	time	.13	task	.10	remove	.12
meal	.07	encourage	.08	allow	.10	perform	.10	linen	.07
proper	.06	feeling	.08	give	.09	procedure	.09	transfer	.06
exercise	.05	express	.06	important	.05	step	.06	soil	.05
alignment	.04	collect	.05	name	.04	include	.04	bed	.12

Note. Values in the parentheses represent the likelihood of observing each topic across the documents. The values associated with the terms represent the likelihood of observing the term within the topic. The asterisks indicate SMEs' uncertainty with the labels provided.

The four cosine indices were also computed based on the document-concept distribution matrix. The cosine computation followed the same procedure as that of the LSA model. The only difference was that the vectors used for the calculation consist of probabilities of observing the topics in a document instead of term-to-concept loadings. Table 8 summarizes the descriptive statistics of the four cosine indices calculated from the LDA model within both enemy pairs and non-enemy pairs. As expected, the cosine indices were all positive, as they were calculated based on probability distributions. On average, the cosines computed from the LDA models had the highest values across all NLP models, except for the cosine between item stems within enemy item pairs. Similarly, the cosine between stems and the cosine between keys were relatively higher than the cosines between stem and key within the enemy item pairs. The average cosine values within the non-enemy pairs were lower compared to those within the enemy pairs, but their values were about five times the size of the cosines from the VSM and LSA models. The variation of the cosine values from the LDA model was also the highest across all NLP approaches. The independent *t*-tests showed significant between-enemy group differences for all the cosine similarity indices with a *p*-value less than .01.

TABLE VIII

DESCRIPTIVE STATISTICS OF COSINE SIMILARITY INDICES FROM THE LDA

Similarity Index	*N*	Mean	*S.D.*	Min.	Max.
Enemy					
$\text{cosine}(\text{stem}_i, \text{stem}_j)$	327	.425	.375	.012	1.000
$\text{cosine}(\text{key}_i, \text{key}_j)$	327	.390	.418	.013	1.000
$\text{cosine}(\text{stem}_i, \text{key}_j)$	327	.144	.253	.013	1.000
$\text{cosine}(\text{key}_i, \text{stem}_j)$	327	.164	.262	.013	1.000
Non-Enemy					
$\text{cosine}(\text{stem}_i, \text{stem}_j)$	1,066,203	.051	.114	.009	1.000
$\text{cosine}(\text{key}_i, \text{key}_j)$	1,066,203	.059	.121	.010	1.000
$\text{cosine}(\text{stem}_i, \text{key}_j)$	1,066,203	.054	.116	.009	1.000
$\text{cosine}(\text{key}_i, \text{stem}_j)$	1,066,203	.054	.114	.009	1.000

4.3 Results from the First Round of Classification

4.3.1 Constructing Classification Dataset

Having completed the NLP analysis and calculated the cosine indices for each item pair, the study moved forward to the classification stage. The dataset for classification was constructed with all 1,066,530 item pairs. The dependent variable was based on the enemy status flagged in the item bank, with the value of 1 indicating enemy item pair and a value of 0 indicating non-enemy item pair. The predictors included the cosine similarity indices produced from the NLP approaches, the average item length, the difference in item difficulty parameters, and a dummy indicator of content overlap. Although all 12 cosine indices from the three NLP approaches were gathered in the classification dataset, only four cosine indices from the same

NLP approach were included in each run of the classification later in the study. The average length of each item pair was generated by first calculating the total number of lemmas identified in both the stem and the key for each item and then taking the average between the item pair. In addition, two other predictors were calculated based on the item meta data. The absolute difference in the calibrated difficulty level (*b* parameter) of each item pair was also included as a predictor. The dummy indicator of content overlap was derived by comparing the test blueprint content areas between each item pair. This indicator took the value of 1 if the content areas matched, and 0 if not matched. Like the cosine similarity indices, these additional predictors are between-item pair variables and are agnostic to the order of the items.

The data were randomly split into 80% training dataset and 20% test dataset. As a result of random splitting, the enemy pairs and the non-enemy pairs were split into the same proportion. The resulting training dataset consisted of 853,224 item pairs, and the test dataset consisted of 213,306 item pairs. Table 9 summarizes the descriptive statistics of all variables included in the classification. The table not only shows the descriptive statistics for the whole dataset but also for the training and test datasets. This table confirms that the splitting of data did not lead to significant alternation across variables and that the enemy pairs, as well as the non-enemy pairs, were split proportionally. The descriptive statistics for the training dataset and the test dataset were highly similar compared to the original dataset. As expected, the enemy status was highly skewed because there were so few enemy item pairs (327 enemy pairs out of all 1,066,530 pairs). All mean cosine similarity indices were low (ranging from .003 to .059), because most item pairs were not enemies. The average item length is approximately 12. The average absolute difference in difficulty parameters was about 1.51 logits. Only about 7.5% of the item pairs shared the same content area.

TABLE IX

DESCRIPTIVE STATISTICS OF VARIABLES IN THE FIRST ROUND OF CLASSIFICATION

	Whole Dataset					Training Dataset					Test Dataset				
Variables	*N*	Mean	*S.D.*	Min.	Max.	N	Mean	*S.D.*	Min.	Max.	N	Mean	S.D.	Min.	Max.
Enemy status	1,066,530	.001	.018	.000	1.000	853,224	.001	.018	.000	1.000	213,306	.001	.017	.000	1.000
VSM cosine ($stem_i$, $stem_j$)	1,066,530	.007	.036	.000	1.000	853,224	.007	.036	.000	1.000	213,306	.007	.036	.000	1.000
VSM cosine (key_i, key_j)	1,066,530	.004	.036	.000	1.000	853,224	.004	.036	.000	1.000	213,306	.004	.037	.000	1.000
VSM cosine ($stem_i$, key_j)	1,066,530	.003	.028	.000	1.000	853,224	.003	.028	.000	1.000	213,306	.003	.028	.000	1.000
VSM cosine (key_i, $stem_j$)	1,066,530	.003	.027	.000	1.000	853,224	.003	.027	.000	1.000	213,306	.003	.026	.000	1.000
LSA cosine ($stem_i$, $stem_j$)	1,066,530	.012	.058	– .258	1.000	853,224	.012	.058	– .258	1.000	213,306	.012	.058	– .165	1.000
LSA cosine (key_i, key_j)	1,066,530	.006	.063	– .993	1.000	853,224	.006	.063	– .993	1.000	213,306	.006	.064	– .986	1.000
LSA cosine ($stem_i$, key_j)	1,066,530	.006	.048	– .989	1.000	853,224	.006	.048	– .989	1.000	213,306	.006	.047	– .984	1.000
LSA cosine (key_i, $stem_j$)	1,066,530	.006	.046	– .988	1.000	853,224	.006	.047	– .988	1.000	213,306	.006	.045	– .974	1.000
LDA cosine ($stem_i$, $stem_j$)	1,066,530	.051	.115	.009	1.000	853,224	.051	.114	.009	1.000	213,306	.051	.115	.009	1.000
LDA cosine (key_i, key_j)	1,066,530	.059	.121	.010	1.000	853,224	.059	.121	.010	1.000	213,306	.059	.122	.010	1.000
LDA cosine ($stem_i$, key_j)	1,066,530	.054	.116	.009	1.000	853,224	.054	.115	.010	1.000	213,306	.054	.117	.009	1.000
LDA cosine (key_i, $stem_j$)	1,066,530	.054	.114	.009	1.000	853,224	.054	.114	.009	1.000	213,306	.054	1.000	1.000	1.000
Average item length	1,066,530	11.826	2.133	5.000	30.500	853,224	11.827	2.133	5.000	30.500	213,306	11.824	2.133	5.000	30.000
Difference in difficulty	1,066,530	1.507	1.167	0.000	12.920	853,224	1.506	1.168	0.000	12.920	213,306	1.510	1.166	0.000	12.250
Content overlap	1,066,530	.075	.264	0	1	853,224	.075	.264	0	1	213,306	.075	.263	0	1

4.3.2 Application of Synthetic Minority Over-sampling on Training Dataset

The SMOTE technique was applied to the training dataset to counter the effect of data imbalance. There were initially 265 enemy item pairs and 852,959 non-enemy item pairs in the training dataset. The SMOTE technique generated 852,694 synthetic enemy item pairs based on attributes of the existing minority sample. The updated training dataset included 852,959 item pairs for each enemy class.

Figure 8 illustrates the distribution of two example cosines for both enemy classes, before and after the application of SMOTE. Each subplot is a scatter plot of the cosine between stems and the cosine between keys. The darker dots represent enemy item pairs, and the lighter dots represent non-enemy item pairs. The LSA plots have a different scale due to the negative cosines. As can be seen in the top row of subplots in Figure 6, the data imbalance issue was evident before the SMOTE application. The majority of lighter dots cluster around the lower left corner of the coordinates, indicating that the non-enemy item pairs typically had lower cosine values. Darker dots scatter sparsely in the plots due to the small number of enemy item pairs, but they mostly occupy the upper right corner representing higher values of both cosines. Comparing the pre-SMOTE cosine distribution across all three NLP models, the polarized pattern is more pronounced in the VSM and LSA model. The non-enemy pairs tend to have higher values on both cosine indices in the LSA plot, while the cosine values of non-enemy pairs seem to have a denser formation around the lower range (.00 to .20) in the VSM plot. For the LDA plot, the darker dots appear to be most scattered, which corresponds to the higher variation of cosine indices in Table 8. Although the majority of both cosine indices clustered around .00 in the LDA model, there was a substantial amount of cases where the cosine measures between keys were located around .20 - .30, while the cosines between stems did not show such cluster.

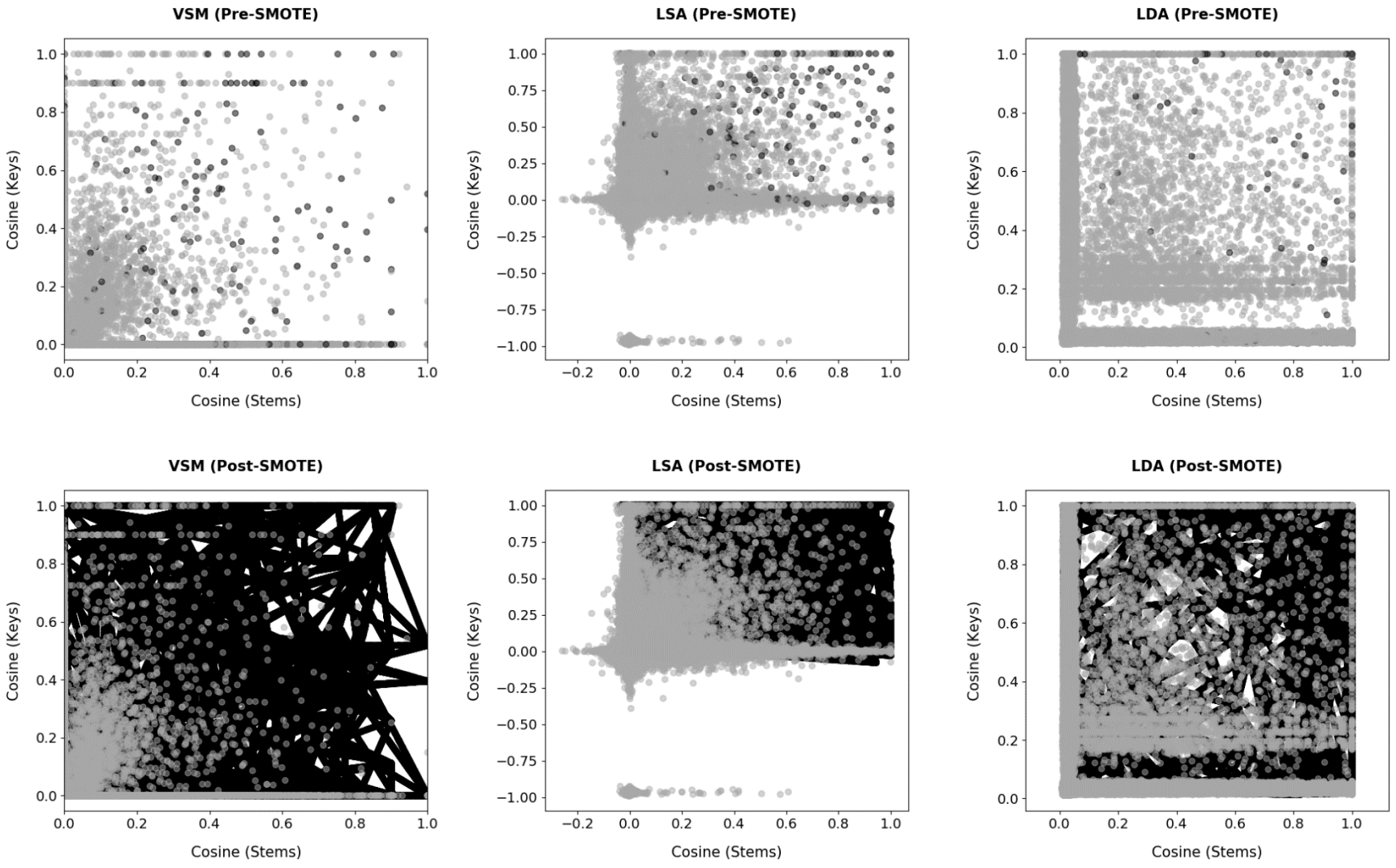

Figure 8　Scatter plots of cosine indices grouped by enemy status before and after SMOTE

The bottom row of Figure 8 shows the data distribution after SMOTE was applied. Synthetic enemy item pairs (indicated by the darker dots) was generated based on the existing enemy sample to produce a balanced dataset. After the SMOTE application, the polarized cosine values between enemy and non-enemy pairs become more evident, with enemy item pairs filling the upper right space associated with higher cosine values. This polarized pattern can be easily observed in the LSA plots. In the VSM and LDA plots, some enemy pairs have low cosine values, but non-enemy pairs still predominantly occupy the lower left of the coordinates.

TABLE X

DESCRIPTIVE STATISTICS OF VARIABLES

IN THE TRAINING DATASET AFTER SMOTE APPLICATION (BEFORE SME REVIEW)

Variables	*N*	Mean	*S.D.*	Min.	Max.
Enemy status	1,705,918	.500	.500	.000	1.000
VSM cosine ($stem_i$, $stem_j$)	1,705,918	.161	.229	.000	1.000
VSM cosine (key_i, key_j)	1,705,918	.139	.264	.000	1.000
VSM cosine ($stem_i$, key_j)	1,705,918	.033	.104	.000	1.000
VSM cosine (key_i, $stem_j$)	1,705,918	.049	.133	.000	1.000
LSA cosine ($stem_i$, $stem_j$)	1,705,918	.242	.308	−.258	1.000
LSA cosine (key_i, key_j)	1,705,918	.183	.315	−.993	1.000
LSA cosine ($stem_i$, key_j)	1,705,918	.053	.151	−.989	1.000
LSA cosine (key_i, $stem_j$)	1,705,918	.072	.179	−.988	1.000
LDA cosine ($stem_i$, $stem_j$)	1,705,918	.239	.316	.009	1.000
LDA cosine (key_i, key_j)	1,705,918	.224	.346	.010	1.000
LDA cosine ($stem_i$, key_j)	1,705,918	.086	.166	.010	1.000
LDA cosine (key_i, $stem_j$)	1,705,918	.104	.190	.009	1.000
Average item length	1,705,918	11.581	2.227	5.000	30.500
Difference in difficulty	1,705,918	1.211	1.022	0.000	12.920
Content overlap	1,705,918	.467	.499	0	1

Table 10 shows the descriptive statistics of variables in the training dataset after the SMOTE application. The balanced training dataset included 1,705,918 item pairs, with equal number of enemy and non-enemy item pairs. The addition of 852,694 synthetic enemy item pairs increased the means for all cosine indices and the proportion of between-item content overlap across the training dataset. The average difference in difficulty parameters was lowered by 0.30 logits. The average item length stayed around eight. All these changes in the basic descriptive statistics were expected from the SMOTE application.

4.3.3 Application of Logistic Regression Classifier and the Artificial Neural Network Classifier

The logistic regression and the Artificial Neural Network classifiers were both applied on the post-SMOTE training data to predict the enemy status of the item pairs in the test data. Table 11 showed the results from the logistic regression classification. The table presents the log odds coefficient for each predictor, its standard error in parentheses, and its significance level as asterisks. The results showed that the four cosine indices, the average item length, the difference in difficulty, and the content overlap indicator were all significant predictors of the item pair's enemy status ($p < .001$ for all predictors). The cosine between item stems was the strongest predictor across all NLP models. The effects of the cosine indices are more pronounced in the VSM and the LSA models than in the LDA model. Controlling for all other predictors, a .01 increase in the cosine between stems would increase the odds of the item pair's enemy status by 27% for the LSA model ($e^{.23697} = 1.267$) and by 16% for the VSM model ($e^{.15191} = 1.164$), whereas for the LDA model the same increase in the cosine between stems would only result in a 6% increase in the odds of enemy status ($e^{.05667} = 1.058$). The other cosine indices also showed strong influences on the prediction of enemy status, indicating significant positive associations between cosine indices and enemy status.

TABLE XI

RESULTS OF LOGISTIC REGRESSION (BEFORE SME REVIEW)

Variables	VSM	LSA	LDA
Cosine ($stem_i$, $stem_j$)	15.191***	23.697***	5.667***
	(0.042)	(0.070)	(0.018)
Cosine (key_i, key_j)	6.672***	10.186***	3.511***
	(0.034)	(0.062)	(0.018)
Cosine ($stem_i$, key_j)	5.858***	7.253***	1.326***
	(0.051)	(0.088)	(0.017)
Cosine (key_i, $stem_j$)	8.211***	12.721***	2.162***
	(0.047)	(0.081)	(0.001)
Average item length	−0.428***	−0.374***	−0.297***
	(0.001)	(0.001)	(0.001)
Difference in difficulty	−0.438***	−0.532***	−0.613***
	(0.004)	(0.004)	(0.003)
Content overlap	2.004***	2.165***	2.883***
	(0.009)	(0.008)	(0.006)
N	1,705,918	1,705,918	1,705,918
Adjusted *R*-squared	.811	.929	.660

Note. *** $p < .001$

The average item length had a negative association with the enemy status. With each unit of increase in item length, the odds of the item pair's enemy status would decrease by 35%–54% across the NLP models when all other predictors were controlled. On the other hand, the more different the item pairs are in terms of their difficulty levels, the less likely they would be predicted as enemies. Each unit of increase in the difference in difficulty levels was associated with of 35%–46% of decrease in the odds of the enemy status. The content overlap was also a strong predictor of the item pair's enemy status. Holding other variables constant, an item pair

from the same content area had 6–17 times higher odds of being enemies than an item pair from different content areas.

The *R*-squared statistics showed that the classification model with the LSA cosine indices as predictors explained the most variance in the enemy status of item pairs (93%), followed by the model utilizing the VSM cosine indices (81%), while the model using the LDA cosine indices explained the least proportion of the variance (66%).

The distributions of the predicted enemy probabilities across all classification models were presented in Table 12, grouped by the enemy status in the item bank. In general, flagged enemy item pairs had a much higher average predicted enemy probability than the non-enemy item pairs across all models.

TABLE XII

DISTRIBUTION OF PREDICTED ENEMY PROBABILITIES (BEFORE SME REVIEW)

Model	*N*	Mean	*S.D.*	Min.	Max.
Enemy					
VSM Logistic	62	.938	.197	.028	1.000
LSA Logistic	62	.939	.194	.042	1.000
LDA Logistic	62	.920	.164	.050	1.000
VSM ANN	62	.929	.230	.005	1.000
LSA ANN	62	.930	.227	.000	1.000
LDA ANN	62	.917	.165	.047	1.000
Non-Enemy					
VSM Logistic	213,244	.065	.134	.001	1.000
LSA Logistic	213,244	.059	.138	.000	1.000
LDA Logistic	213,244	.128	.190	.001	1.000
VSM ANN	213,244	.052	.155	.005	1.000
LSA ANN	213,244	.054	.151	.000	1.000
LDA ANN	213,244	.130	.185	.032	1.000

4.3.4 ROC Curve and Precision-Recall Curve Analyses

Altogether, this study explored six classification models which consisted of the combination of two classifiers and three sets of cosine indices from the NLP approaches. The four cosine indices from each NLP model, along with the average item length, the difference in difficulty parameters, and the indicator for content overlap were used to predict the enemy status of item pairs. For each of the NLP × Classifier combinations, the classification model was fitted on the post-SMOTE balanced data, and the parameters estimated from the training data were used to make predictions for the enemy probability for each item pair in the test data.

An ROC curve analysis was first performed to assess the global performance of each classification model and to identify the range of optimal cutoff point. To plot the ROC curve, 213,306 (equivalent to the sample size of the test dataset) probability cutoffs with equal intervals between .00 and 1.00 (range of probability threshold) were identified. The FPR and TPR were computed at each of these probability cutoffs. Figure 9 shows the resulting ROC curves by plotting the FPR against the TPR at each of the probability cutoffs. An ROC curve shows the change in both rates when the probability threshold is gradually relaxed from 1.00 to .00. A higher probability threshold implies a stricter classification criterion (i.e. an item pair is only classified as enemies if the predicted probability is .95 or higher), and vice versa. As the probability threshold relaxes, the TPR and FPR are expected to increase as more cases are classified as positive.

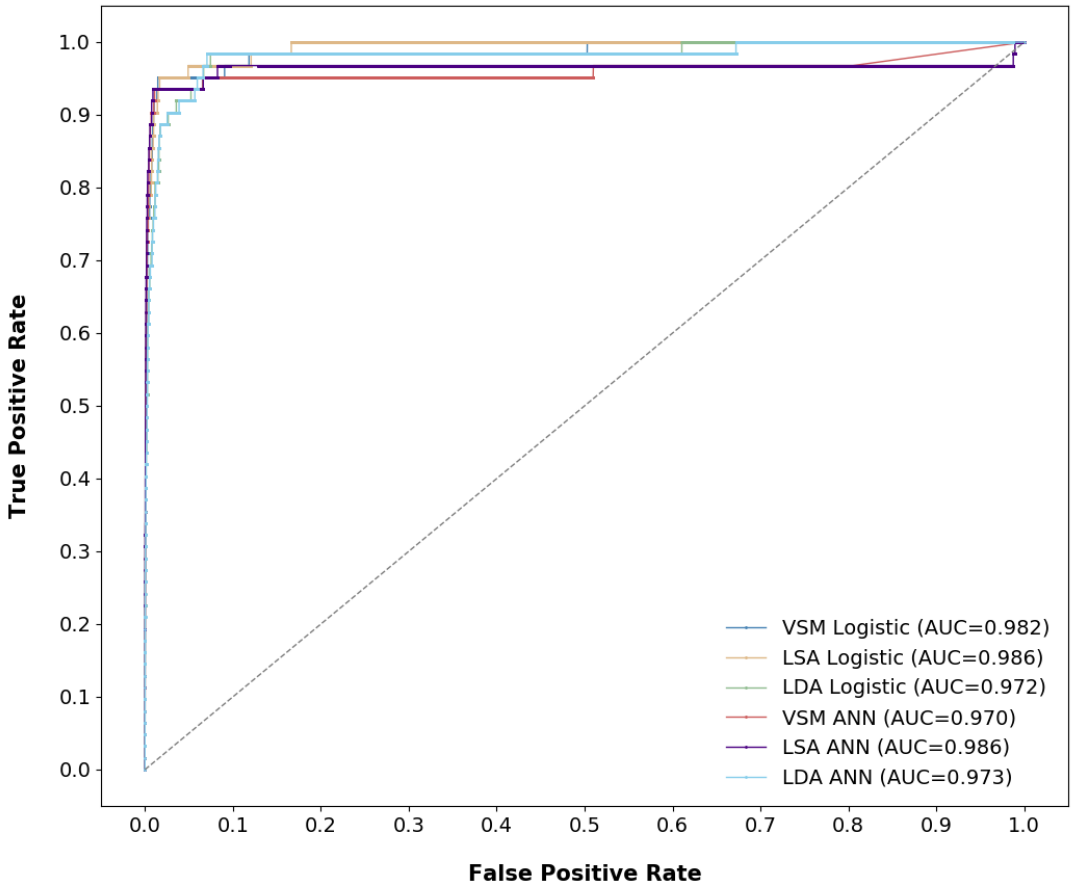

Figure 9 ROC Curves Before SME Review

The ROC curves in Figure 9 show the rate of increase of both TPR and FPR as the cutoff threshold decreases. The general trends of the ROC curves were very similar across all models. The FPR increased slowly in the beginning, while the TPR quickly rose up to above .85 as the probability cutoff gradually relaxed, which suggests that most of the enemy item pairs in the item bank were correctly classified by the model at a higher probability cutoff. The inflection point on the ROC curve, which typically suggests the optimal balance point between the TPR and FPR, occurred when the TPR is above .85 and the FPR was below .10 across all models.

The two classification models using LDA cosine indices appeared to have a relatively lower TPR when the FPR was held constant across the models. This under-performance was also reflected by the lower scores on the Area Under the Curve. The area under the ROC curve ranged from .970 to .986 across all classification models. Within the same classifier, the model utilizing LSA cosine indices showed slightly better global performance (AUC=.986) across all models. However, the differences were very small ($\leq$.01).

I next turned to the Precision-Recall Curves in Figure 10 for additional information on the model performance. As expected, the curves present a negative association between the recall and precision as the cutoff threshold decreased from 1.00 to .00. However, the pattern of the curves did not point to an obvious reflection point which would reveal the location of optimal cutoff that balances the recall and the precision rates. The change rates for the recall and the precision appeared to be consistent across the cutoffs. Therefore, the Precision-Recall Curve analysis was not informative in providing guidance for the cutoff selection in this case. Due to the large difference between the numbers of enemy and non-enemy item pairs, the number of False Positive cases were expected to be substantially higher than the number of True Positive cases. A low precision rate was commonly observed in classification study with class imbalance. The low precision rate also made it difficult to gaze at the pattern of the Precision-Recall Curve when the class imbalance was extreme.

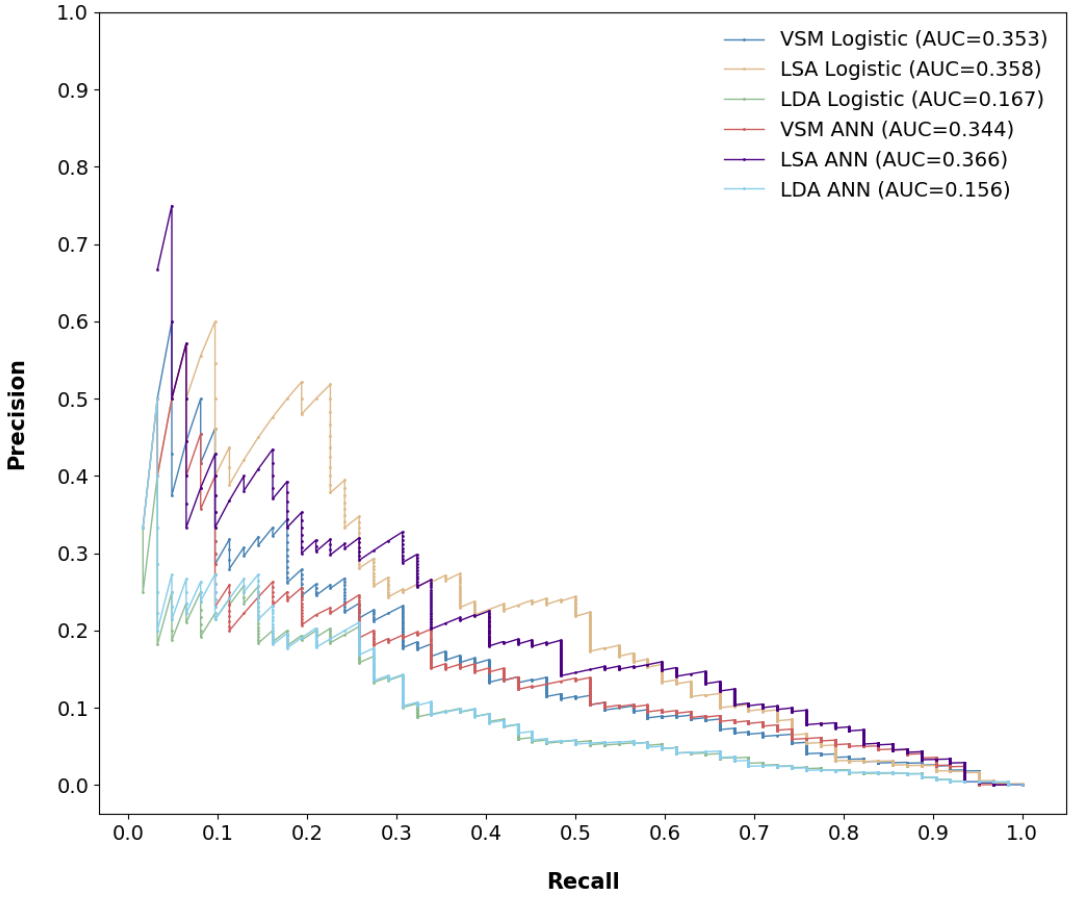

Figure 10 Precision-Recall Curves Before SME Review

The area under the Precision-Recall Curve across the models was lower than .50, indicating weak discriminating power in terms of the Precision and Recall rates. The models using the LSA cosine indices consistently showed better performance within the same classifier (AUG=.358 with logistic regression classifier; AUG=.366 with ANN classifier). Models using the VSM cosine indices followed closely behind (AUG=.353 with logistic regression classifier; AUG=.344 with ANN classifier). The LDA models fell short in terms of the area under the Precision-Recall Curve measure, with less than half of the value compared to other models.

The ROC curve analysis provided general guidance for the range of the optimal cutoff. Since the Precision-Recall Curves showed an unclear pattern of results and were not very informative in revealing an inflection point, I used the ROC curve analysis to select the probability cutoffs for further evaluation. To identify a reasonable range of probability cutoffs, I first set the target values for each of the classification metrics based on the location of the inflection points on the ROC plot. The target of TPR was set at .85, where the ROC curve started to reach the inflection point, and 85% of the enemy item pairs were detected by the model. In terms of FPR, due to the large number of True Negative cases included in the calculation of FPR, it became more sensible to set the target for the number of False Positive cases rather than for the FPR. After consulting with the SMEs, the maximum number of False Positive cases was set at 4,000, which corresponded to an FPR of around .02. The corresponding values of the probability cutoff can be obtained at these target values of TPR and FPR.

TABLE XIII

PROBABILITY CUTOFFS AT THE TARGET VALUES OF TPR AND FPR BEFORE SME REVIEW

Variables	Cutoff (FPR = .020)	Cutoff (TPR = .850)
VSM Logistic	.456	.931
LSA Logistic	.521	.952
LDA Logistic	.753	.858
VSM ANN	.629	.960
LSA ANN	.644	.963
LDA ANN	.745	.855
Average	.625	.920

Table 13 shows the probability thresholds at the target values of the TPR and the FPR. The middle column shows the cutoff values corresponding to an FPR of .020 for each model. The model will produce 4,000 or fewer False Positive item pairs when the probability cutoff is set above this cutoff threshold. Similarly, the last column shows the cutoff values corresponding to a TPR of .850. A model with probability cutoff at or below this value will correctly identify 85% of the enemy item pairs. Based on this table, the average cutoff that limited the FPR at or below 2% was around .60, and the average cutoff that achieved a TPR at or above 85% was around .90. This range of cutoff provided guidance on the location of the optimal cutoff threshold that met both the target values of TPR and FPR (i.e. FPR $\leq$ .02; TPR $\geq$ .85). Therefore, it is sensible to focus model evaluation within the cutoff range of .60 to .90. Due to the scope of this research, this study only examined the model performance at four probability cutoffs within this range – .60, .70, .80, and .90 – in hope of revealing the approximate location of the optimal cutoff.

4.3.5 Evaluation of the Classification Performance Metrics

Having settled on the four probability cutoffs within the optimal range, the confusion matrix was derived at each of the cutoff points, based on which multiple performance metrics were calculated across all classification models. Table 14 presents the classification results across all models at each of the cutoffs. The numbers of item pairs identified as True Positives, False Negatives, False Positives, and True Negatives are shown in the table, followed by various resulting performance metrics: recall, precision, specificity, accuracy, and F_1 score.

In general, most of the item pairs in both enemy classes were correctly classified across all models and cutoffs. Out of the 62 enemy item pairs in the test dataset, 47 to 59 were correctly classified. This large proportion of enemy item detection was reflected by the high recall rate,

ranging from 75.8% to 95.2%. Most of the non-enemy item pairs in the item bank were also correctly detected by the model, which was indicated by the high specificity rate at or above 95% across the table. The False Positive cases, though sometimes quite large in practical terms, remained at under 5% of the non-enemy item pairs. This high degree of correct classification also led to the high accuracy rate across the models and cutoffs.

The classifications exhibited excellent discriminating ability for both enemy classes based on the high degree of correct classification and low misclassification, and the class imbalance still transferred to the distributions of some performance metrics. The difference in size between the True Positive cases and False Positive cases caused the precision rate to be low across the conditions, with a range of .005 to .027. As a result, F_1 score also turned out to be low because of the large difference between the recall and the precision rates.

It was not surprising to observe small differences in the specificity rates and the accuracy rates across models due to the skewed distribution of enemy classes. By contrast, recall, precision, and F_1 score were the metrics that would be most fluctuated by an improvement in classification in this context.

Comparison across the NLP approaches

After observing the general patterns of the classification results, the model performance was compared within each condition of the designed (NLP × Classifier × Cutoff). When comparing across the NLP approaches, the LDA models stood out with a substantial number of False Positive cases. At a lower cutoff, LDA models produced more than twice the number of FP cases (as many as 10,913) as other models. When the cutoff increased, this difference narrowed down to about 10% of the FP cases, but the LDA models still produced over 257 more FP item pairs than other NLP models at the cutoff of .90. The LDA models also showed some degree of

under-performance in terms of enemy item detection ability. The LDA models identified fewer enemy associations out of the 62 enemy item pairs, showing a lower recall rate across conditions. This lack of identification became more evident as cutoff increases to .80 or above, with a 10%-20% lower recall rate. The lower number of True Positive cases and higher number of the False Positive cases translated to lower precision rates for the LDA models. Similarly, the higher proportion of misclassification in both enemy classes resulted in worse performance in terms of specificity, accuracy and F_1 score compared to other NLP models.

Comparing between the VSM and the LSA models, the enemy item recall rate remained identical until the cutoff was increased to .90 where the LSA identified one more enemy association. However, the difference in recall rate was small (1%). In terms of the classification performance of non-enemy item pairs, the VSM models show better performance with more correctly classified non-enemy item pairs and fewer false positive cases. The LSA models produced 222-489 more False Positive cases than the VSM models across the classifiers and cutoffs. Although the differences in the precision and specificity rates between the two NLP approaches were below 0.3%, the additional False Positive cases could potentially place extra burden on the SMEs review. The difference in accuracy metric was also small (between 0.1% to 0.3%). On the other hand, these two NLP models showed larger differences in terms of F_1 score favoring the VSM models, though the differences remained under 0.5%.

Comparison across the Classifiers

Based on Table 14, the performance of the two classifiers appears to interact with the NLP approaches and cutoffs. For the VSM models, the logistic regression classifier performed better than the ANN classifier at the cutoffs of .60 and .70, with higher values across all performance metrics. However, at the cutoffs of .80 and .90, the recall rates of ANN classifier

surpassed that of the logistic regression models, though the differences were minor (one additional enemy item pair correctly classified). For the non-enemy item pairs classification, the logistic regression classifier achieved better classification results for the VSM models, which were indicated by fewer False Positive item pairs and better specificity rate across all cutoffs. The precision, accuracy, and F_1 score were consistently higher with the logistic regression, though this advantage diminished when the cutoff reached .90.

The performance comparison between the two classifiers showed a different pattern for the LSA models. The logistic regression classifier produced better results than the ANN classifier on all performance metrics at the cutoffs of .60 and .70. However, when the cutoff was increased to .80, the ANN classifier started to gain a slight advantage in the recall rate by correctly identifying one more enemy item pair, with the rest of the metrics still out-performed by the logistic regression classifier. At the cutoff of .90, the performance was completely reversed, with the ANN performing better on all the metrics.

Comparing the two classifiers within the LDA models, the ANN classifier produced fewer false positive items across all cutoffs, showing overall better classification results for the non-enemy class. For enemy class identification, the recall rate remained identical between the two classifiers until the cutoff reached .90, where the logistic regression identified 5% more enemy item pairs than the ANN classifier. As a result, the precision rate and F_1 score for the ANN classifier suffered slightly at the cutoff of .90.

Comparison across the Cutoffs

Comparing the model performance across the cutoffs, the recall rate was relatively stable when the cutoff was between .60 and .80. At least 55 out of the 62 enemy item pairs in the test dataset were detected by the models as enemies at the cutoff of .80, corresponding to a minimum

recall rate of 89%. The most dramatic change in the recall rate occurred when the cutoff was increased from .80 to .90 for the LDA models, as the LDA Logistic model suffered an 8% drop and the LDA ANN model a 13% drop. Each increase in the cutoff threshold led to a substantial decrease in the number of False Positive cases in practical terms. The number of False Positive cases was decreased by approximately 80% for the LDA models and by at least 50% for the VSM and the LSA models across the cutoffs. Although this difference in the non-enemy classification was not evident in the change of specificity due to the large base of non-enemy item pairs, it is much more sensible to choose a higher cutoff if the false positive items were to be reviewed by human SMEs. As discussed previously, the precision rate and the F_1 score are more sensitive to the improvement in the model performance when class imbalance exists. As can be observed from the table, both metrics more than doubled as the cutoff increased from .60 to .90.

In summary, the overall performance across the models showed that the LDA models were less efficient in classifying both enemy and non-enemy item pairs under all conditions. The performance metrics for both the VSM and the LSA models improved greatly at minimal cost of the recall rate when the threshold was set at a higher value. The LSA ANN model at the cutoff of .90 appeared to be the best performing model, as all performance metrics were maximized at no cost of recall. The VSM ANN model was also a competing option with equivalent performance metrics except a 3% lower recall rate. Although the LSA Logistic model produced slightly inferior classification metrics, a logistic regression classification took significantly less time and computing power to converge than an ANN model. Therefore, the LSA Logistic may also be a sensible choice when computing time and hardware limitation are considered.

TABLE XIV

CLASSFICATION RESULTS BEFORE SME REVIEW

Model	TP	FN	FP	TN	Recall	Precision	Specificity	Accuracy	F_1
Cutoff = .60									
VSM Logistic	59	3	4,113	209,131	.952	.014	.981	.981	.028
LSA Logistic	59	3	4,622	208,622	.952	.013	.978	.978	.026
LDA Logistic	57	5	10,913	202,331	.919	.005	.949	.949	.010
VSM ANN	58	4	5,774	207,470	.935	.010	.973	.973	.020
LSA ANN	58	4	6,067	207,177	.935	.009	.972	.972	.018
LDA ANN	57	5	10,774	202,470	.919	.005	.949	.949	.010
Cutoff = .70									
VSM Logistic	59	3	3,422	209,822	.952	.017	.984	.984	.033
LSA Logistic	59	3	3,829	209,415	.952	.015	.982	.982	.030
LDA Logistic	56	6	6,373	206,871	.903	.009	.970	.970	.018
VSM ANN	58	4	4,447	208,797	.935	.013	.979	.979	.026
LSA ANN	58	4	4,534	208,710	.935	.013	.979	.979	.026
LDA ANN	56	6	6,181	207,063	.903	.009	.971	.971	.018
Cutoff = .80									
VSM Logistic	57	5	2,751	210,493	.919	.020	.987	.987	.039
LSA Logistic	57	5	3,046	210,198	.919	.018	.986	.986	.035
LDA Logistic	55	7	4,495	208,749	.887	.012	.979	.979	.024
VSM ANN	58	4	3,307	209,937	.935	.017	.984	.984	.033
LSA ANN	58	4	3,257	209,987	.935	.017	.985	.985	.033
LDA ANN	55	7	4,327	208,917	.887	.013	.980	.980	.026
Cutoff = .90									
VSM Logistic	55	7	2,018	211,226	.887	.027	.991	.991	.052
LSA Logistic	56	6	2,240	211,004	.903	.024	.989	.989	.047
LDA Logistic	50	12	2,497	210,747	.806	.020	.988	.988	.039
VSM ANN	56	6	2,033	211,211	.903	.027	.990	.990	.052
LSA ANN	58	4	2,088	211,156	.935	.027	.990	.990	.052
LDA ANN	47	15	2,370	210,874	.758	.019	.989	.989	.037

4.4 SME Review

As discussed in the previous chapter, the enemy association between the item pairs was not completely identified due to the sheer volume of the item bank. Only a small portion of the item pairs were reviewed by the SMEs during the item review and test form building process. Among the limited number of item pairs reviewed, only those that were confirmed to be enemy item pairs were recorded in the item bank. Therefore, it is likely that some enemy items in the item bank were never assessed by SMEs, and their enemy relationships were never recorded. In the initial classification, these unidentified enemy items were used to train the classifiers as though they are not enemies. To address this issue, an SME review was conducted to assess the high probability enemy item pairs classified as enemies by the models but not recorded in the item bank as enemy item pairs (i.e. False Positive item pairs).

A subset of item pairs was identified for the SME review. Based on the model training results in the initial classification, predictions were made on the entire dataset (including training dataset and test dataset) using all models. All false positive item pairs above the cutoff of .60 were then merged across the models to form the subset of item pairs for SME review. Table 15 shows the numbers of True Positive, False Negative, False Positive, and True Negative cases when the predictions were made on the whole dataset using a cutoff of .60. The models utilizing the LDA cosine indices produced the largest number of False Positive cases (54,581 and 53,760 item pairs, respectively). All the False Positive item pairs across the models were merged for the SME review. All together, there were 83,939 FP item pairs at the cutoff of .60 across the models. For each item pair in this subset, the predicted probabilities across the models were averaged to generate an overall enemy probability. These FP item pairs were then sorted by this overall

enemy probability in a descending order. Within the descending order, the false positive item pairs were grouped into sets of 20 item pairs.

TABLE XV

CLASSFICATION RESULTS ON THE WHOLE DATASET (CUTOFF=.60) BEFORE SME REVIEW

Model	TP	FN	FP	TN
VSM Logistic	302	25	20,971	1,045,232
LSA Logistic	302	25	23,236	1,042,967
LDA Logistic	287	40	54,581	1,011,622
VSM ANN	307	20	28,942	1,037,261
LSA ANN	307	20	30,497	1,035,706
LDA ANN	287	40	53,760	1,012,443

The two SMEs conducted the review from higher probability sets to lower probability sets and determined true enemy relationships for each item pairs. For every 20 items, the SMEs recorded the number of items confirmed to be true enemies. The SMEs terminated the review when they encountered only one true enemy item pair (less than 10%) within a set of 20 item pairs. During the review process, the two SMEs compared their enemy relationship decisions every 200 item pairs to check for agreement. When their decisions did not agree for any item pair, the SME resolved the disagreement through discussion and finalized the enemy status for these item pairs. As a result, the SMEs completed the review of a total of 1,040 FP item pairs in three consecutive days, among which 469 (45%) were confirmed to be true enemy item pairs. The average value of the overall enemy probabilities of these confirmed enemy item pairs was .95, with a minimum enemy probability of .92. In other words, the SMEs encountered the

first set of 20 item pairs that contained less than 10% true enemy item pairs below the overall probability of .92. The SME review was therefore terminated at this probability cutoff.

Within these newly confirmed enemy item pairs, the number of uniquely identified item pairs for each model was examined. We found that the numbers of unique identifications from the same NLP method (but different classifiers) are relatively similar, with variation of between 2 to 5 item pairs. The majority (80%) of these new enemy item pairs were flagged across all or two (LSA or LDA) of the NLP approaches. The LSA Logistic and LSA ANN models uniquely identified 23 and 25 of these true enemy pairs, respectively. This number was smaller for the VSM models (15 by VSM Logistic model; 16 by VSM ANN model). The LDA models uniquely identified one true item pair across the two classifiers.

4.5 Results from the Second Round of Classification

Based on the SMEs feedbacks, the 469 confirmed enemy relationships were updated in the classification dataset. The updated dataset included 796 enemy item pairs and 1,065,734 non-enemy item pairs.

Another iteration of classification was then conducted, in which the models were retrained to produce the updated results. The re-run of classification followed the same procedures as the previous round. The data were split into 80% training dataset and 20% test dataset, resulting in 647 enemy item pairs and 852,577 non-enemy item pairs within the training dataset, and 149 enemy pairs and 213,157 non-enemy pairs within the test dataset. The descriptive statistics of all variables in the whole dataset, the training dataset, and the test dataset for the second round of classification are presented in Appendix A. Except for the update on the dependent variable (enemy status), no changes were made to other variables in the dataset, as the cosine similarity indices and the item meta data remained the same. The slight variation in the

descriptive statistics of predictors in the training and test datasets was attributed to the random assignment during the data splitting. Although the SME review more than doubled the enemy item pairs, the mean of the updated enemy status variable still showed that the updated dataset included less than 0.1% enemy item pairs. The standard deviation of enemy status increased due to the addition of enemy item pairs.

Likewise, the SMOTE technique was applied to the updated dataset to generate 851,930 synthetic enemy item pairs. The final training dataset after the SMOTE application included a total of 1,705,154 item pairs (852,577 in each enemy class). Appendix B shows the descriptive statistics of variables in this training dataset after the SMOTE application. As expected, the addition of synthetic enemy item pairs increased the means for all cosine indices and the proportion of between-item content overlap across the training dataset. The average difference in difficulty parameters was lowered by .25 logits. The average item length remained around 8.

4.5.1 Updated ROC Curve and Precision-Recall Curve Analyses

The models were retrained on the updated training dataset, and predictions were made on the test dataset. Results from the logistic regression classifier are shown in Appendix C, in which all predictors remained significant and an increase in *R*-squared statistics was observed. The predicted probabilities across all models grouped by enemy status are presented in Appendix D. Figure 11 shows the updated ROC curves from the second round of classification. Compared to the ROC curves from the previous round, some improvements on the model performance can be identified. First, the figure suggests a greater coverage of the area under the ROC curves, especially for the VSM and the LSA models. A closer comparison at the global measures confirmed that the AUG improved by at least .01 for all models. Although the .01 improvement in the AUG measure may not seem substantial, a sharper reflection on the curves became

evident. As the probability threshold gradually relaxed, the TPR rapidly increased to as high as 90% at a relatively strict cutoff threshold. The inflection points on the curves were more identifiable for the VSM and LSA models, which occurred when the TPR was above .90 and the FPR below .05.

Another visible change in the ROC curves is that the difference in performance between the classifiers was minimized, which was indicated by the narrower distance between curves within the same NLP approach across cutoffs. For models within the LSA or the LDA approach, the difference in the AUG measure between the logistic regression and the ANN classifiers was negligible ($\leq .01$), as the curves overlapped with each other. The difference in AUG were larger (.08) for the two models within the VSM approach, with the logistic regression classifier model outperforming the ANN classifier model.

With the narrowing of performance across the classifiers, the difference in the NLP approaches became more evident. The LDA models showed a .20 under-performance of the AUG measure, which was also reflected by the shape of the curves and the relatively ambiguous inflection points.

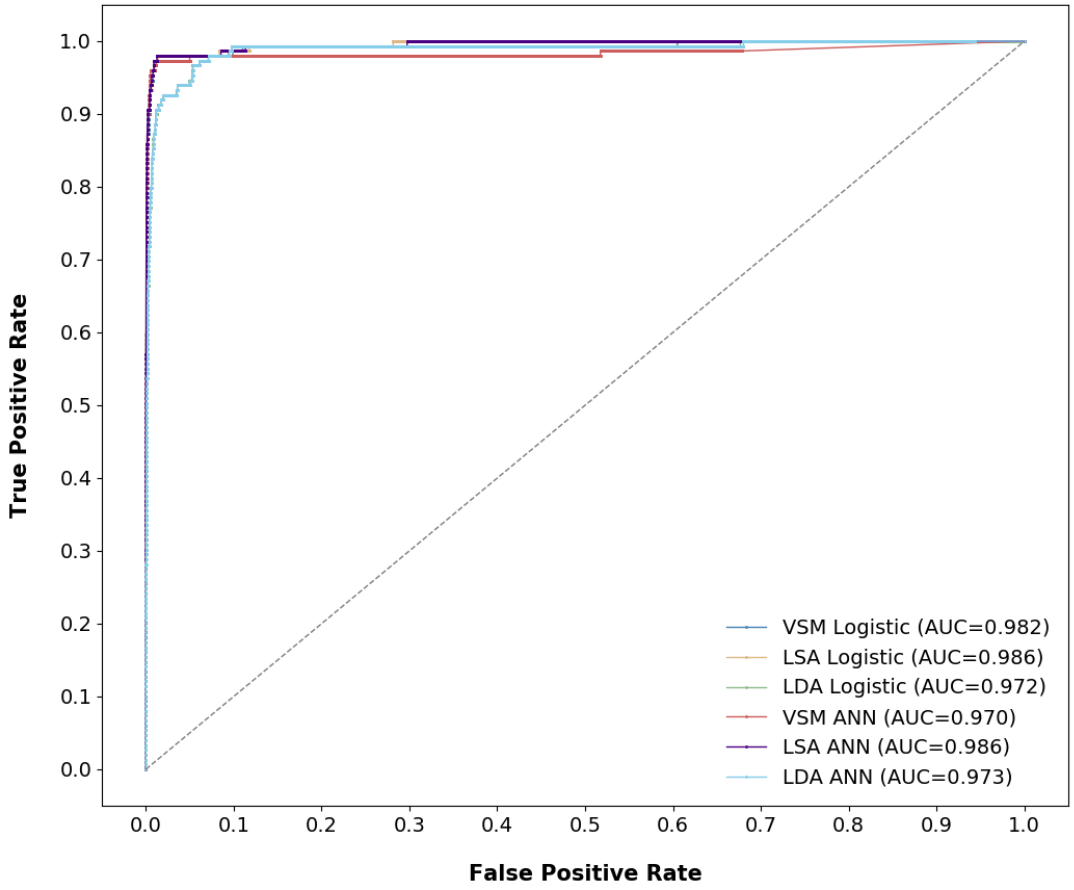

Figure 11 Updated ROC Curves After SME Review

Next, the Precision-Recall Curves are presented in Figure 12. Compared to the same figure from the previous round, the updated Precision-Recall Curves show significant changes in terms of global performance. The area under the Precision-Recall Curve measures for VSM and LSA models all rose above .50, showing great improvement in the discriminating ability of the classifications. Although there was a significant increase in the AUG measures for the LDA models as well, the AUG measures remained under .50, which indicated weaker discriminating ability of the classification models.

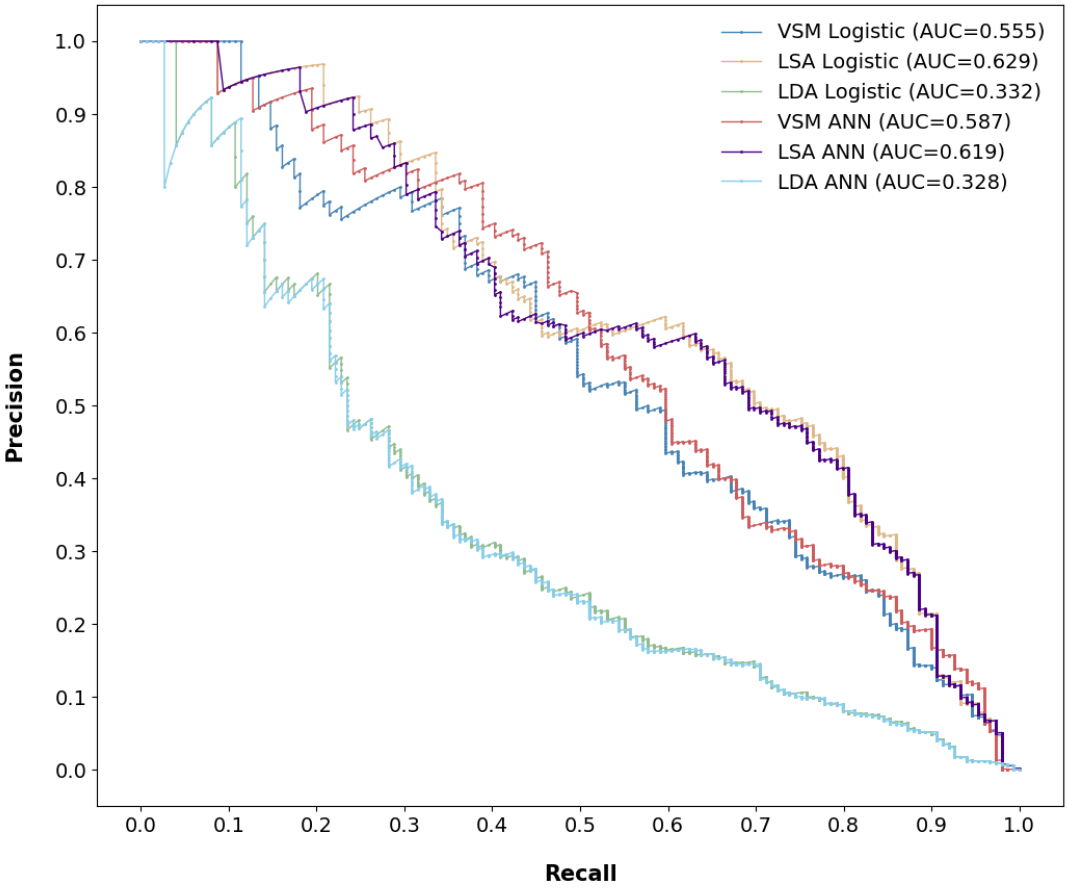

Figure 12 Updated Precision-Recall Curves After SME Review

Similar to the findings from the ROC curves, the difference in model performance between the two classifiers became smaller, as the distance between the curves within the same NLP approach tightened across the cutoffs. On the other hand, the difference between the NLP approaches became more pronounced in this figure. We can see a clear separation of the LDA models from other models, indicating a substantial degree of under-performance compared to the VSM and LSA models. Across the probability cutoffs, the LDA models exhibited much lower precision when controlling for the recall rate.

I then examined the cutoffs at the target values of TPR and FPR (see Table 16). The target values were re-assessed due to the changes in the model performance. Judging from the updated ROC curves, the inflection point occurred when the TPR reached above .90 and the FPR below .05. In addition, the FPR is required to stay below .018 in order to control the False Positive cases under 4,000 cases. Table 19 shows the corresponding cutoff thresholds across the models at the FPR of .018 and the TPR of .900. We can see a downshift in the cutoff thresholds at the target FPR value, indicating that the updated classifications produced fewer False Positive cases at the same cutoff. The cutoffs that achieved the TPR of 90% was located around .90 to .95 across the models, with an average value of .934.

TABLE XVI
PROBABILITY CUTOFFS AT THE TARGET VALUES OF TPR AND FPR AFTER SME REVIEW

Variables	Cutoff (FPR = .018)	Cutoff (TPR = .900)
VSM Logistic	.295	.960
LSA Logistic	.303	.984
LDA Logistic	.744	.838
VSM ANN	.396	.963
LSA ANN	.217	.973
LDA ANN	.796	.884
Average	.459	.934

4.5.2 Updated Classification Performance Metrics

For the purpose of comparison, the classification results at the same four cutoffs (.60, .70, .80, and .90) are presented in Table 17. The updated results consistently show a high degree of correct identification in both enemy classes, characterized by high recall, specificity, and accuracy rates. Compared to the classification results in the first round, the updated results showed substantial improvement in the recall, precision, and F_1 score across all conditions. Overall, the updated classifications were able to correctly detect 1%-12% more true enemy item pairs, with the largest improvement in recall occurring at the cutoff of .90.

As expected, the improvement in the specificity rates was less evident due to the large number of True Negative cases. However, the number of False Positive item pairs were significantly reduced in practical terms. For both the VSM and the LSA models, the numbers of False Positive item pairs were nearly halved compared to the previous round. Driving this change was the decreased number of non-enemy item pairs with the addition of enemy

associations, as well as the improvement in the correct classification of non-enemy item pairs. Although the FP item pairs were greatly reduced in the updated classification results, new item pairs have surfaced as potential enemy pairs which were not previously predicted in the initial round of classification. Further examination revealed that there were 23-45 such predicted enemy item pairs revealed across the models in the updated classification results. When the item bank tagging is updated to reflect more accurate enemy relationships after each iteration of enemy item pair review, it is expected that more of such new FP item pairs will be unveiled during the earlier iterations as the model updates the dependent variable and recalibrates.

Comparison across the NLP approaches

Comparing across the models utilizing different NLP approaches, the VSM and the LSA models continued to out-perform the LDA models with better performance metrics across all conditions. The LSA models produced the highest recall rates for classifying enemy item pairs and, under most circumstances, achieved the best specificity rate across the NLP models. The only exceptions occurred when the LSA approach was paired with a logistic regression classifier at the cutoffs of .60 and .70, in which cases the VSM models yielded fewer False Positive cases and higher specificity. Comparing the classification results between the two rounds of classification, a major improvement in non-enemy item pairs classification for the LSA models was observed. As we recall, the LSA models consistently produced more False Positive cases and lower specificity rates than the VSM models in the first round of classification, while in the second round of classification, the LSA models exhibited fewer misclassifications of non-enemy class in most conditions.

Comparison across the Classifiers

The two classifiers continued to show different performance patterns across the three NLP approaches. However, the patterns changed in the updated classification results. This was characterized by an increase in performance of the ANN classifier for the LSA models and, by contrast, a decrease in performance of the ANN classifier for the VSM and LDA models.

When pairing with the VSM or the LDA approach, the logistic regression classifier outperformed the ANN classifier on all performance metrics. However, this effect was reversed when pairing with the LSA approach, where the ANN classifier consistently yielded better classification results across all cutoffs.

Comparison across the Cutoffs

The pattern of performance change across the cutoffs was similar to results from the previous round of classifications. The recall rates across the models was relatively stable at the lower cutoffs, but started to separate at the higher cutoffs. The recall rates of the VSM ANN model and the LDA ANN model suffered at the cutoffs of .60 and .80.

Similarly, there was a substantial decrease in the number of False Positive cases in practical terms each time the cutoff was raised. The number of False Positive cases was reduced greatly across the cutoffs (an approximately 80% drop for the LDA models; at least a 50% drop for the VSM and the LSA models). The updated results continue to show superior classification performance at a higher cutoff for the VSM and LSA models, with a significant increase in the precision and the F_1 score (approximately 50%) at a minimal cost of recall rate (1%-5%).

As the cutoff increased, the difference in performance across the models becomes more pronounced. The LDA models consistently showed underperformance across the cutoffs. The differences between the VSM and LSA models were negligible (under 1%) at the lower cutoffs of .60 and .70. However, the advantage of the LSA ANN model emerged as the cutoff increased.

At the cutoff of .90, the LSA ANN model yielded the best classification results with a 3%-9% higher recall rate, a 2%-9% higher precision rate, and a 5% to 15% higher F_1 score.

The updated results continued to favor the LSA ANN model as the best performing model, with all performance metrics moderately higher than the other models at all cutoffs. At the cutoffs of .90, the LSA ANN model showed superior detection of enemy item pairs (96% identified) and produced as few as 877 False Positive item pairs. The VSM ANN model was no longer a competing model due to its less ideal recall rate at the same cutoff. Instead, the LSA Logistic model and the VSM Logistic model turned out to be the runner-up options.

TABLE XVII

CLASSFICATION RESULTS AFTER SME REVIEW

Model	TP	FN	FP	TN	Recall	Precision	Specificity	Accuracy	F_1
Cutoff = .60									
VSM Logistic	144	5	2,217	210,940	.966	.061	.990	.990	.115
LSA Logistic	145	4	2,270	210,887	.973	.060	.989	.989	.113
LDA Logistic	140	9	8,367	204,790	.940	.016	.961	.961	.032
VSM ANN	143	6	2,883	210,274	.960	.047	.986	.986	.090
LSA ANN	145	4	1,849	211,308	.973	.073	.991	.991	.135
LDA ANN	141	8	11,148	202,009	.946	.012	.948	.948	.025
Cutoff = .70									
VSM Logistic	143	6	1,862	211,295	.960	.071	.991	.991	.133
LSA Logistic	143	6	1,890	211,267	.960	.070	.991	.991	.131
LDA Logistic	138	11	4,411	208,746	.926	.030	.979	.979	.059
VSM ANN	142	7	2,418	210,739	.953	.055	.989	.989	.105
LSA ANN	144	5	1,528	211,629	.966	.086	.993	.993	.158
LDA ANN	138	11	5,725	207,432	.926	.024	.973	.973	.046
Cutoff = .80									
VSM Logistic	142	7	1,512	211,645	.953	.086	.993	.993	.158
LSA Logistic	143	6	1,499	211,658	.960	.087	.993	.993	.160
LDA Logistic	141	8	3,095	210,062	.946	.044	.985	.985	.083
VSM ANN	136	13	1,826	211,331	.913	.069	.991	.991	.129
LSA ANN	143	6	1,221	211,936	.960	.105	.994	.994	.189
LDA ANN	136	13	3,774	209,383	.913	.035	.982	.982	.067
Cutoff = .90									
VSM Logistic	139	10	1,135	212,022	.933	.109	.995	.995	.195
LSA Logistic	139	10	1,053	212,104	.933	.117	.995	.995	.207
LDA Logistic	129	20	1,821	211,336	.866	.066	.991	.991	.123
VSM ANN	135	14	1,160	211,997	.906	.104	.995	.994	.187
LSA ANN	143	6	877	212,280	.960	.140	.996	.996	.245
LDA ANN	130	19	2,203	210,954	.872	.056	.990	.990	.105

5. DISCUSSION

This chapter summarizes the findings from this study and the practical implications. I first discuss the findings by addressing each research question. I also summarize the practical implications of the findings for automatic enemy item detection and make recommendations to testing organizations employing the automatic detection procedure. I then point out the strengths and limitations of this study. Finally, I offer suggestions on other relevant research directions of automatic enemy item detection that future studies may pursue.

5.1 Summary of Findings by Research Questions

5.1.1 Research Question 1

Do natural language processing techniques adequately capture item similarity?

This study compared the four cosine similarity indices produced by each NLP approach between the enemy and non-enemy item pairs. The results showed that the cosine indices for the enemy item pairs were consistently and significantly higher than those found in the non-enemy item pairs within each NLP approach, both before and after the SME review. Figure 8 further confirmed that the distributions of cosine indices showed distinct patterns across the NLP techniques, with the enemy item pairs consistently having higher cosines than the non-enemy item pairs.

The logistic regression results showed that all predictors, including the four cosine indices produced by the NLP models, were strong significant predictors of the enemy status between item pairs ($p < .001$). The logistic regression coefficients indicated that the changes in the cosine indices have the most impact on the probability of enemy status, compared to other predictors which included the average item length, the difference in item difficulty parameters,

and a dummy indicator of content overlap. The classification models utilizing the cosine indices from the VSM and the LSA approaches were able to account for over 80% of the variance in the item pairs' enemy status (81% and 92%, respectively). By contrast, the model with the LDA cosine indices did not explain as much variance in the enemy status (66%).

SMEs helped review the concepts/topics extracted from the LSA and the LDA approaches. Their feedback showed that both NLP approaches have the ability to extract sensible concepts/topics that reflect the underlying semantic structure of natural language. The SMEs were able to assign meaningful labels to the concepts/topics according to their associated terms, which implies that the extracted concepts/topics are comparable to the human perception of conceptual similarities. However, the SMEs also noted some differences between the LSA concepts and the LDA topics. The concepts extracted from the LSA analysis appeared to be conceptually distinct from one another, as each concept was associated with a different content area or procedure that is likely to be assessed in a nurse aide licensure test. In comparison, the topics extracted from the LDA approach seemed to be more general and ambiguous. The SMEs sometimes had difficulty assigning a substantive label to some of the concepts. Some topics seemed to fall into the same content area, while others had conceptually different terms.

This study also examined the classification results before and after the SME review of false positive item pairs. The classification metrics showed that, in general, the accuracy of the classification models for the three NLP approaches were high (at or above .95) across the cutoffs and classifiers. A large proportion of enemy and non-enemy item pairs were correctly identified by all NLP models. When comparing the three NLP approaches, the LSA and the VSM models consistently outperformed the LDA models on all classification metrics. Before the SME review of FP item pairs, both the VSM and LSA models produced competing recall rates of enemy item

pairs, and the VSM models produced fewer FP item pairs than the LSA model. However, among the FP item pairs predicted across all models, more item pairs were confirmed to be true enemies from the LSA models, which also had the largest number of uniquely identified true enemy item pairs. This indicates that, although the LSA approach produced more FP item pairs initially, it was able to detect true enemy pairs that were not flagged by the VSM and LDA models. After the SME review and the enemy status update, the performance of the LSA models improved greatly and produced better recall rates and fewer false positive item pairs than the VSM models under most conditions.

In contrast to the previous study in which the LDA approach achieved satisfactory classification results (above .90 recall rates both before and after the review) and was sensitive to the enemy status update after the SME review (Weir et al., 2019), the under-performance of LDA model found in this study is thought-provoking. Although it may not be possible to confirm the root cause of such difference through study replication due to test security reasons, a number of reasons can be speculated. First, the item characteristics between the two studies differed. The average item length after the NLP transformation was 35 in Weir's (2019) study, while it was approximately 12 in the current study. Traditionally, the LDA techniques were developed and applied to longer documents (i.e. paragraphs, articles, and movie reviews). The length of the document could affect the stability of the LDA model. It can be challenging to achieve stable estimation of topic-document distribution and topic-term distribution with shorter documents. Because the item keys were considered separate documents in this study and they typically contained fewer terms especially after the NLP transformation, the performance of the LDA model may suffer from the short length, making it difficult to estimate the document-topic distributions for these documents. This may explain the floor effect of the LDA cosine measures

between item keys observed in Figure 8. Moreover, the breadth and depth of item content varied across the studies. In the current study, the job description of nurse aides determined that there is a limited number of tasks/procedures a nurse aide is qualified to perform. On the other hand, the other study included test items that assessed more complex medical knowledge which involved comprehension of medical procedures and diagnostic processes. In addition, the test items employed a simpler structure and mainly required basic level of cognitive processing. The lack of diversity in the item content and cognitive level may undermine the performance of the LDA.

Regarding research question 1, all three of the NLP techniques have shown promising ability in capturing the item similarity. The overall classification performance of the LDA models were less efficient in classifying both enemy and non-enemy item pairs under all conditions. The LSA models were more sensitive to the update/correction of enemy status, as the performance of LSA models improved and surpassed the VSM models after the SME feedbacks were incorporated to update the enemy status. The LSA models turned out to be the most efficient in capturing item similarity and classifying enemy item pairs.

5.1.2 Research Question 2

Compared to a logistic regression classifier, does the Artificial Neural Network classifier improve the accuracy of classifying enemy item pairs?

The results of the study showed that the performance of the two classifiers interacted with the NLP approaches and cutoffs. The update of enemy status based on the SME feedbacks also had a significant impact on the performance of the classifiers.

Before the SME review of false positive item pairs and the update of enemy status, the patterns of performance for the two classifiers differed for each NLP approach. For the VSM and LDA models, the logistic regression classifier performed better at lower probability cutoffs, but

the ANN classifier gained more advantage when the probability cutoff was increased. For the LSA models, the ANN classifier consistently yielded better classification accuracy than the logistic regression classifier. However, the recall rate suffered for the ANN classifier at the cutoff of .90.

After the SME feedback was incorporated into the updated dataset, the ANN classifier, when combined with the LSA approach, consistently outperformed the logistic regression classifier on the recall rates. The LSA ANN classifier yielded substantially fewer false positive item pairs than the LSA logistic regression classifier. However, the same performance pattern was not observed when combined with the VSM or the LDA models: the ANN classifier's ability to identify non-enemy item pairs was less ideal when compared with the logistic regression classifier, which was reflected by higher numbers of false positive items and lower specificity rates of the VSM ANN and the LDA ANN models.

5.1.3 Research Question 3

What probability cutoff is considered optimal for classifying a sufficient number of existing enemy item pairs while keeping the number of falsely classified item pairs manageable?

As the probability cutoff increased, fewer enemy item pairs were classified by the model due to stricter classification criterion. As a result, the recall rate inevitably suffered, whereas the precision rate improved due to a reduced number of false positive item pairs. Comparing the classification performance across the cutoffs (both before and after the SME review), the performance metrics of all models produced the best results at the cutoff of .90 except for the recall rates. A closer examination of recall rates showed that, when the cutoff increased from .60 to .90, the recall rates suffered by 0-11% across the models. For the LSA and VSM models which consistently produced better performance metrics, the reduction in recall rates was smaller

(between 0% to 5%). Even at the cutoff of .90, the models were able to achieve high levels of recall: all models correctly identified more than 75% of enemy item pairs before the SME review was conducted, and the recall rates were improved to above 86% after the SME feedback of enemy status was incorporated into the classification. More specifically, the LSA models and the VSM models yielded higher recall rates (89% - 96%) than the LDA models (76% - 87%). This level of recall was found to be higher than or comparable to the recall rates from previous studies (Peng et al., 2018; Weir et al., 2018; Peng et al., 2019; Weir, 2019).

The number of false positive item pairs was more than halved as the cutoff increased to .90, which, in substantive terms, reduced the burden of SME review by over a thousand item pairs when comparing with the results at the cutoff of .60. The resulting number of item pairs that required review was manageable (approximately 2000-2500 for the first round of classification and 900-2000 after the SME review), given that the SMEs completed the review of approximately 1,000 item pairs within three days.

The accuracy rates of all models peaked at the cutoff of .90, indicating the fewest cases of enemy status misclassification. In addition, the F_1 score, which is an indication of the tradeoff between the recall and the precision rates, also yielded the best values when the cutoff was set at .90.

To summarize, the models consistently produced optimal results at the cutoff of .90. Over 76% of the enemy item pairs were classified at this cutoff and the recall rate was improved to above 86% after the SME review and update of item bank enemy status. The number of false positive item pairs that required further SME review were also manageable at this cutoff.

5.1.4 Research Question 4

Does the automatic enemy item detection procedure help reveal more true enemy items previously not identified in the item pool? Does retraining the model, using the input from SME review of false positive item pairs, help improve the accuracy of classifying enemy item pairs?

The automatic detection procedure has shown promising ability to predict item pairs' enemy status. The first round of classification revealed high probability enemy item pairs that had not been flagged in the item bank. Guided by this result, the SMEs reviewed approximately 1,000 item pairs in a reasonable time frame, and a high percentage (45%) of the reviewed pairs were successfully confirmed as enemies.

The feedback of the SME review was then incorporated to update the enemy status of item pairs which, in return, improved the accuracy of the dependent variable used for the classification model. As a result, all the performance metrics were markedly improved across the models in the re-trained classification, especially in the precision rate, the accuracy rate, and the F_1 score. The number of false positive item pairs was also remarkably reduced, easing the burden of further SME review. This iterative process helped strengthen the enemy relationship monitoring of the item bank and efficiently reduced the time and resources allocated to enemy item pair detection.

Given the successful identification of additional enemy item pairs, a supplemental analysis was conducted in which SMEs also reviewed the FN item pairs in Table 15 which were predicted to have enemy probabilities below .60 but were tagged as enemy item pairs in the item pool. Although these item pair were highly likely to have been reviewed during previous test assembly processes, human errors could occur during manual management of the item bank, and the definition of enemy items may also shift overtime which calls for re-evaluation of existing

enemy associations. Upon gathering the FN item pairs across all models in the entire item bank, a total of 53 unique FN item pairs were sent to further review in which two SMEs reviewed all these item pairs and verified the enemy status of each pair. The review result confirmed that one out of these FN items pair was in fact a non-enemy item pairs. The SME review determined that the contents of the two items within this pair did not warranty the enemy relationship – one item pertaining to addressing the emotional need of the patient and the other related to the nurse aide's response to the emergency call light. The SMEs suspected that this item pair was likely assigned an enemy association due to operational errors made during the item bank management process. Furthermore, this item pair was identified as non-enemy pair by the LSA Logistic model and the LSA ANN model but not detected by the rest of the models. Upon closer inspection, these two items have two overlapping words, which possibly infatuated the similarity indices produced by the VSM and the LDA models. This additional analysis showed that not only did the automatic enemy detection process reveal additional enemy relationships within the item pool, it can also help verify the existing enemy relationships and identify incorrectly flagged enemy relationships.

5.2 Implications

The following section discusses the implications of this study as well as the operational recommendations implied by this study.

The results of this study showed robustness of the proposed enemy detection process utilizing Natural Language Processing techniques and classification approaches for the automatic identification of enemy relationships in the item bank. The topic modeling NLP techniques successfully extracted meaningful concepts/topics from the item text. The feedback from the SME review implied that the LSA approach was able to extract more refined and distinct concepts compared to the LDA approach. The relatively poor LDA model's performance was

also reflected in the classification results, where the classification metrics consistently favored the VSM and the LSA models. The VSM Logistic model showed advantages in the precision, accuracy, and F_1 score in the initial round of classification, while the LSA ANN model achieved better performance when the enemy relationships were more accurately reflected in the updated classification dataset. The models yielded the optimal classification results at the cutoff of .90, with a high proportion of enemy item pairs successfully detected and a manageable number of false positive item pairs for review. The SME review of the false positive item pairs confirmed that approximately 45% of the reviewed pairs to be true enemies, which helped identify additional enemy associations in the item bank. After the new enemy relationships were updated in the dataset, the performance metrics of all models were significantly improved, and the numbers of misclassified item pairs were markedly reduced.

Although this automatic process is not intended to replace the human SME review, it will narrow down the number of item pairs to be reviewed and provide a helpful starting point for the enemy relationship screening. This process will minimize the repetitive manual efforts needed for item bank maintenance. In addition, this study implies that the effectiveness of this automatic process will continuously improve throughout the iterative process of human SME review and re-classification. As more and more enemy associations are identified from the review of false positive item pairs, the accuracy of enemy relationship flagging in the item bank will be enhanced. Item pairs with confirmed enemy status can also be marked and excluded from future reviews.

This automatic detection process not only allows screening of enemy item pairs within the entire item bank, but it also offers operational flexibility to control the time and resources allocated for this task. For example, this process can be performed on smaller item pools for

faster iteration time. SMEs can modify the stopping rules of the false positive item pair review process to focus on item pairs with higher estimated enemy probabilities. The stopping point where the SME review concluded indicates the probability cutoff below which fewer enemy item pairs were encountered. It can be used as an informative reference cutoff for the next iteration of enemy search and SME review. In addition, as time is of the essence in the enemy item pair identification, practitioners need to consider the computing time and hardware limitation when selecting the model. The topic modeling approaches, especially the LDA method, which employs MCMC sampling, involve additional analytical steps compared to the VSM models and thus require longer time to converge. Moreover, the machine learning classifier requires more computing power than the logistic regression classifier due to the gradient descent and backpropagation procedures. The entire automatic detection procedure, when using logistic regression as the classifier, typically finishes within 15 minutes. When performing models with the ANN classifier, this study utilized a GPU accelerated server, and the computing time of the models took approximately 30-50 minutes.

Although this study was focused on the application of the automatic detection process in the nurse aide licensure examination, it can be applied to a variety of other testing contexts. For large scale testing programs having employed fully adaptive computerized testing, there is constant demand to monitor and screen for enemy items as required by continuous and frequent testing. The automatic process will provide human reviewers with helpful indices of enemy relationships and greatly reduce the manual effort needed for this onerous task. For testing programs utilizing fixed-form assessments, equivalent test forms are usually generated through automatic test assembly algorithms and the test forms need to be screened for enemy item pairs before the test publication. More often than not, enemy items would be identified within the

same test forms which would call for the replacement of items. Multiple iterations of such form review would usually be carried out until the test forms could be verified to be free of enemy items. With the help of this automatic enemy item detection approach, the enemy items can be screened and identified in advance, which reduces the likelihood of selecting enemy item sets to the same test form. Depending on the availability of the item-meta data, the model can be modified to include additional predictors to improve the enemy classification. Some item banks include various methods for the categorization of item content which may further inform the enemy relationships. The automatic approach can also be applied to pretest items. Even though the difficulty parameters are not available for pretest items, the NLP techniques can still analyze the item text and serve the purpose of item similarity screening.

The updated classification results indicated that the model performance will be improved as more iterations of this automatic detection procedures are conducted in the item bank. It is expected that the enemy status would be stabilized over time when the enemy relationships in the item bank are exhaustively scrutinized and verified throughout the iterations. As a result, there would be fewer FP item pairs to review after each iteration. However, it is necessary to routinely conduct such iterations of the automatic enemy detection procedure in real-world testing context. As more test items are added periodically to the item bank, the text corpus is constantly updating, and the topics themselves are changing. Moreover, the topics and standards of care evolve over time due to advancements in the field. Re-training the automatic detection model on a routine basis accounts for such shifts and ensures that measures of between-item similarities are accurate snapshots of the testing period. Since large testing programs typically rotate different item pools for test security purposes, it is prudent to conduct this automatic procedure before each test publication to ensure that enemy relationships are scrutinized within the item pool. It is

important that the enemy relationships within the item bank are constantly monitored and screened, as having enemy item sets on the same test diminishes the measurement precision and test validity. The thorough identification of enemy relationships also improves the test form assembly process for testing organizations, as much time and resources are devoted to finding, tagging, and replacing enemy item sets after the test forms are assembled. Furthermore, the enemy relationship identification would ensure that the examinees are not administered enemy item sets on the same test, which avoids the confusion of receiving duplicative items on the same test and potential interference in the examinee's ability estimation.

5.3 Strength and Limitations

A major strength of this study is that it examined and compared the effectiveness of the automatic enemy detection process across various NLP approaches and classifiers based on a single, entire operational item bank. Previous studies have been focused on one or two of these methods and utilized different item pool data, whereas this study allows direct comparisons of performance to be made between various models across a wide range of possible cutoffs, which provides useful insights to practitioners on the model selection. Additionally, the item sample of this study includes approximately 1,500 operational items. Compared to the previous studies (Becker & Kao, 2009; Peng et al., 2018; Weir et al., 2018; Peng et al., 2019), this research offers a larger database that ensures adequate training of the topic modeling NLP approaches and the ANN machine learning method.

However, certain limitations of the study must be noted. First, when resampling was applied to address the imbalanced data issue using a traditional frequentist approach SMOTE, the variance of the data were artificially altered and modeled, which might lead to questions about the validity of the classification results. Therefore, the implications of the recommended cutoffs

from this study are not applicable to studies that do not implement the SMOTE adjustment. Some alternative methods could be applied without altering the dataset. For example, a Bayesian approach could be an alternative method to address the data imbalance issue by applying a balancing informative prior and weighting the data in favor of the smaller subset so that the influence of imbalance to the overall prediction could be minimized. Second, this study used the point estimates to calculate the cosine similarity indices without taking the associated error terms into account. This might have introduced bias in the classification results and caused additional item pairs to be misclassified. Third, this study assumed stop words could limit the false positive cases but did not further investigate the validity of this assumption. The results could vary to some degree if a different list of stop words were used, and the extent of how this would introduce error is unknown to this study. In addition, given the large size of the item bank, the true number of concepts/topics is unknown. While this study used the proportion of variance explained and singular value for determining the number of concepts in the LSA model and perplexity score for the LDA model, there are various other methods to select the ideal number of latent concepts/topics. In addition, due to the scope of the research, only four probability cutoffs were examined with an increment of .10 to provide an approximate location of the ideal cutoff. Finding the optimal cutoff requires a more exhaustive search of probability thresholds, especially at the higher end of the scale (between .90 and 1.00).

The SME review of the FP item pairs conducted in this study has focused on the highest enemy probability item pairs, and the review was terminated when fewer enemy pairs were encountered. This stopping rule was chosen mainly due to operational reasons, because it appeared to be the most cost-effective approach to identify as many enemy pairs as possible while limiting the number of item pairs the SMEs need to review. However, it is likely that true

enemy item pairs still exist below the stopping point. This automatic approach will inevitably miss some true enemy item pairs and some bias may have been introduced due to this stopping rule. When time and resources allow, human reviewers are encouraged to apply more exhaustive review rules which will increase the number of enemy item pairs being identified through each iteration of SME review. As aforementioned, the accuracy of the model is bound to improve over time when more enemy relationships are unveiled and tagged in the item bank.

Another potential limitation lies within the generalizability of this study. Due to the diversity of testing programs, the item type mixture and availability of item-meta data differ drastically. For instance, some of the predictors used in this study may not be readily available in a different item pool. A number of factors may have an impact on the classification performance of the automatic approach, which include, but are not limited to, item type, item length, size of item pool for training, model selection of the NLP methods, content area and cognitive levels involved within the items. This study used a heuristic approach where the enemy probabilities across all models were averaged and then sorted for the SME review. If a single model was applied, the stopping point of the review may have been different. Moreover, due to the confidential nature of test items, the majority of this type of studies cannot be replicated by researchers outside of the testing organization.

5.4 Future Studies

This study has revealed other directions of future research. First, the research design described in this study can be expanded to include additional classifiers. For example, *k*-nearest neighbor classifier and random forest classifier have also been applied for item text classification in other studies. Second, the results from this study suggested that the models achieved better performance at the higher probability cutoffs. Future research could explore additional

probability cutoffs between .90 and 1.00 to further refine the optimal cutoff. Third, while there is no best way to determine the number of latent concepts/topics for the models, future studies could explore other methods, such as using the coherence score, to select the ideal number of concepts/topics. Furthermore, future research could examine the refinement of stop words and their effect on the classification performance, as this topic is not well documented in the literature.

The approaches used in this study could also prove useful for other tasks related to test development and item bank maintenance. For example, the NLP methods can be applied to extract topics from the item bank for analysis of content coverage. This information can be used to identify content areas requiring item replenishment and provide targeted guidance for item writing. Moreover, the measurement of between-item similarity can help testing organizations investigate allegedly leaked or stolen test items by comparing them to the item bank. The NLP approaches could also help with the quality control of the newly developed items. It can be used to analyze various linguistic features and syntax structure of the newly developed items and compare them with items with known quality.

In addition, future research could explore the possibility of incorporating online lexical databases or ontologies in computing the similarity indices. The strength of Natural Language Processing lies within the ability to process and analyze enormous amount of data, and the model performance benefits from additional input of items, text, and contents. Some existing lexical databases or ontologies offer pre-trained word embeddings that mathematically describe the lexical proximity between words/terms, which are generated from analyzing large corpus not limited to the sample item data used in this study. Utilizing these pre-trained word embeddings could help enhance the efficiency and performance of the automatic enemy detection process.

While this study used multiple-choice items with a simpler, shorter item structure that do not require complex cognitive abilities, future research could explore the generalizability of this automatic approach when applied to different item pools. With the evolution of modern testing, many testing programs are starting to include a diverse mixture of item types. For example, a number of licensure examinations and educational assessments already incorporated more sophisticated items which assess the examinees' ability to analyze, diagnose and generalize. These types of items may describe a more detailed and complex problem (i.e. patient vignette, clinical indices) and require multiple higher-level cognitive processes to arrive at the solution. They can also be presented in a variety of formats (i.e. matrix response items, drag and drop items, short answer items) and even be organized into testlets with a series of context dependent items. Future investigation into how this automatic detection process can be expanded to address different item types will contribute to the refinement of the methods.

5.5 Conclusion

This study helps to show the robustness of the Natural Language Processing techniques in automatic identification of enemy item pairs. The findings from this study may assist testing organizations in making decisions regarding how they screen for enemy item pairs and which model and cutoff to use for this process. The most significant difference between this research and the previous studies is that this study systematically investigated the classification performance from the numerous conditions across the NLP techniques, classifiers, and probability cutoffs. More specifically, it examined the ability of topic modeling NLP approaches in capturing conceptual similarity between item pairs and the potential of using a machine learning classifier for the enemy item pair classification. The results showed that the LSA approach was able to extract refined distinct concepts from the item bank that emulated human

perceptions of conceptual similarities. The classification results from the numerous conditions indicated that the LSA models and the VSM models consistently outperformed the LDA models and yielded optimal results at the cutoff of .90.

With the input of the SMEs, the automatic detection process helped identify additional enemy relationships previously untagged in the item bank. This process also greatly reduced the time and manual labor needed for enemy relationship monitoring and offered flexibility for SME review. After the SME feedback was incorporated into the item bank enemy flagging, the performance of the re-trained models was significantly enhanced based on all classification metrics. The LSA ANN model yielded the best results after the model retraining. However, the LSA Logistic model and the VSM Logistic model still produced compelling performance with less computing time. Therefore, I recommend that testing organizations focus the resources for enemy identification on the highest probable enemy item pairs predicted by the models and choose the logistic regression classifier if the available computing power is limited, as the enemy status review and update are more likely to have the most impact on the enemy relationship identification in the item bank. Finally, this study helped reveal many promising future research directions.

Within the parameters of this study, we can conclude thus far that the proposed automatic enemy detection process is effective in identifying enemy item pairs in the item bank, and it helps reduce the time and resources required by this daunting task. The findings of this study offer practical implications to testing organizations in item bank monitoring and maintenance. The improvement in enemy item identification is crucial for ensuring the measurement precision and test validity, especially for high-stake licensure examinations.

While this research showed promising results from the applications of the Natural Language Processing techniques for automatic enemy item pair detection, there are still many areas that need to be explored by future studies, particularly in the contexts in which topic modeling and machine learning approaches have been newly implemented.

APPENDICES

APPENDIX A

DESCRIPTIVE STATISTICS OF VARIABLES IN THE SECOND ROUND OF CLASSIFICATION

	Whole Dataset					Training Dataset					Test Dataset				
Variables	*N*	Mean	*S.D.*	Min.	Max.	*N*	Mean	*S.D.*	Min.	Max.	*N*	Mean	*S.D.*	Min.	Max.
Enemy status	1,066,530	.001	.027	.000	1.000	853,224	.001	.028	.000	1.000	213,306	.001	.026	.000	1.000
VSM cosine ($stem_i$, $stem_j$)	1,066,530	.007	.036	.000	1.000	853,224	.007	.036	.000	1.000	213,306	.007	.036	.000	1.000
VSM cosine (key_i, key_j)	1,066,530	.004	.036	.000	1.000	853,224	.004	.036	.000	1.000	213,306	.004	.037	.000	1.000
VSM cosine ($stem_i$, key_j)	1,066,530	.003	.028	.000	1.000	853,224	.003	.028	.000	1.000	213,306	.003	.028	.000	1.000
VSM cosine (key_i, $stem_j$)	1,066,530	.003	.027	.000	1.000	853,224	.003	.027	.000	1.000	213,306	.003	.026	.000	1.000
LSA cosine ($stem_i$, $stem_j$)	1,066,530	.012	.058	− .258	1.000	853,224	.012	.058	− .258	1.000	213,306	.012	.058	− .165	1.000
LSA cosine (key_i, key_j)	1,066,530	.006	.063	− .993	1.000	853,224	.006	.063	− .993	1.000	213,306	.006	.064	− .986	1.000
LSA cosine ($stem_i$, key_j)	1,066,530	.006	.048	− .989	1.000	853,224	.006	.048	− .989	1.000	213,306	.006	.047	− .984	1.000
LSA cosine (key_i, $stem_j$)	1,066,530	.006	.046	− .988	1.000	853,224	.006	.047	− .988	1.000	213,306	.006	.045	− .974	1.000
LDA cosine ($stem_i$, $stem_j$)	1,066,530	.051	.115	.009	1.000	853,224	.051	.114	.009	1.000	213,306	.051	.115	.009	1.000
LDA cosine (key_i, key_j)	1,066,530	.059	.121	.010	1.000	853,224	.059	.121	.010	1.000	213,306	.059	.122	.010	1.000
LDA cosine ($stem_i$, key_j)	1,066,530	.054	.116	.009	1.000	853,224	.054	.115	.010	1.000	213,306	.054	.117	.009	1.000
LDA cosine (key_i, $stem_j$)	1,066,530	.054	.114	.009	1.000	853,224	.054	.114	.009	1.000	213,306	.054	.113	.009	1.000
Average item length	1,066,530	11.826	2.133	5.000	30.500	853,224	11.827	2.133	5.000	30.500	213,306	11.824	2.133	5.000	30.000
Difference in difficulty	1,066,530	1.507	1.167	0.000	12.920	853,224	1.506	1.168	0.000	12.920	213,306	1.510	1.166	0.000	12.250
Content overlap	1,066,530	.075	.264	0	1	853,224	.075	.264	0	1	213,306	.075	.263	0	1

APPENDIX B

DESCRIPTIVE STATISTICS OF VARIABLES IN THE TRAINING DATASET AFTER SMOTE APPLICATION (AFTER SME REVIEW)

Variables	*N*	Mean	*S.D.*	Min.	Max.
Enemy status	1,705,154	.500	.500	.000	1.000
VSM cosine ($stem_i$, $stem_j$)	1,705,154	.182	.244	.000	1.000
VSM cosine (key_i, key_j)	1,705,154	.180	.299	.000	1.000
VSM cosine ($stem_i$, key_j)	1,705,154	.037	.102	.000	1.000
VSM cosine (key_i, $stem_j$)	1,705,154	.044	.118	.000	1.000
LSA cosine ($stem_i$, $stem_j$)	1,705,154	.269	.312	− .258	1.000
LSA cosine (key_i, key_j)	1,705,154	.232	.352	− .993	1.000
LSA cosine ($stem_i$, key_j)	1,705,154	.061	.154	− .989	1.000
LSA cosine (key_i, $stem_j$)	1,705,154	.070	.170	− .988	1.000
LDA cosine ($stem_i$, $stem_j$)	1,705,154	.254	.328	.009	1.000
LDA cosine (key_i, key_j)	1,705,154	.242	.346	.010	1.000
LDA cosine ($stem_i$, key_j)	1,705,154	.090	.168	.010	1.000
LDA cosine (key_i, $stem_j$)	1,705,154	.103	.184	.009	1.000
Average item length	1,705,154	12.817	2.290	2.000	24.500
Difference in difficulty	1,705,154	1.264	1.041	0.000	12.920
Content overlap	1,705,154	.508	.500	0	1

APPENDIX C

RESULTS OF LOGISTIC REGRESSION (AFTER SME REVIEW)

Variables	VSM	LSA	LDA
Cosine ($stem_i$, $stem_j$)	14.458***	23.192***	5.689***
	(0.050)	(0.082)	(0.018)
Cosine (key_i, key_j)	7.174***	11.369***	4.395***
	(0.041)	(0.072)	(0.016)
Cosine ($stem_i$, key_j)	4.410***	5.533***	0.609***
	(0.068)	(0.116)	(0.025)
Cosine (key_i, $stem_j$)	7.662***	11.510***	2.161***
	(0.057)	(0.092)	(0.021)
Average item length	−0.584***	−0.374***	−0.409***
	(0.002)	(0.001)	(0.001)
Difference in difficulty	−0.899***	−0.532***	−0.877***
	(0.007)	(0.004)	(0.004)
Content overlap	2.882***	2.165***	4.355***
	(0.014)	(0.008)	(0.008)
N	1,705,154	1,705,154	1,705,154
Adjusted *R*-squared	.913	.935	.779

Note. *** $p < .001$

APPENDIX D

DISTRIBUTION OF PREDICTED ENEMY PROBABILITIES (AFTER SME REVIEW)

Model	*N*	Mean	*S.D.*	Min.	Max.
Enemy					
VSM Logistic	149	.962	.154	.004	1.000
LSA Logistic	149	.964	.151	.006	1.000
LDA Logistic	149	.937	.155	.009	1.000
VSM ANN	149	.964	.161	.001	1.000
LSA ANN	149	.958	.160	.004	1.000
LDA ANN	149	.946	.144	.010	1.000
Non-Enemy					
VSM Logistic	213,157	.028	.103	.000	1.000
LSA Logistic	213,157	.024	.100	.000	1.000
LDA Logistic	213,157	.073	.177	.000	1.000
VSM ANN	213,157	.023	.108	.001	1.000
LSA ANN	213,157	.019	.090	.001	1.000
LDA ANN	213,157	.080	.190	.004	1.000

REFERENCES

Ackerman, T. A., & Spray, J. A. (1986). *A general model for item dependency.* Paper presented at the Annual Meeting of American Educational Research Association, San Francisco, CA.

Aldabe, I., & Maritxalar, M. (2014). Semantic similarity measures for the generation of science tests in Basque. *IEEE Transactions on Learning Technologies*, *7*(4), 357–375.

Alpaydin, E. (2014). *Introduction to machine learning* (3rd Ed.). Cambridge, Massachusetts: The MIT Press.

Ando, R. K. (2000). Latent semantic space: Iterative scaling improves precision of inter-document similarity measurement. *Proceedings of the 23rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2000)*, Athens, Greece, 216–223.

Aswani Kumar, C., Radvansky, M., & Annapurna, J. (2012). Analysis of a vector space model, Latent Semantic Indexing and formal concept analysis for information retrieval. *Cybernetics and Information Technologies, 12*(1), 34-48.

Bates, M. (1995). Models of natural language understanding. *Proceedings of the National Academy of Sciences of the United States of America, 92*(22), 9977–9982.

Becker, K. A., & Olsen, J. B. (2012). *Generating enhanced item writing materials with natural language processing*. Paper presented at the Annual Meeting of National Council on Measurement in Education, Vancouver, BC, Canada.

Becker, K. A., & Kao, S. (2009). *Finding stolen items and improving item banks.* Paper presented at the Annual Meeting of American Educational Research Association, San Diego, CA.

Bejar, I. I. (1996). *Generative response modeling: Leveraging the computer as a test delivery Medium.* Princeton, NJ: Educational Testing Service.

Bejar, I. I., Lawless, R. R., Morley, M. E., Wagner, M. E., Bennett, R. E., & Revuelta, J. (2003). A feasibility study of on-the-fly item generation in adaptive testing. *The Journal of Technology, Learning and Assessment, 2*(3). https://ejournals.bc.edu/index.php/jtla/article/view/1663

Belarouci, S., & Chikh, M. A. (2017). Medical imbalanced data classification. *Advances in Science, Technology and Engineering Systems Journal, 2*(3), 116-124.

Belov, D. I., & Knezevich, L. (2008). *Predicting item difficulty with semantic similarity measures.* Paper presented at the Annual Meeting of the National Council on Measurement in Education, New York, NY.

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research, 3,* 1137–1155.

Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM: Review, 37*(4), 573-595.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. Sebastopol, CA: O'Reilly.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research, 3,* 993–1022.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain*. New York: David McKay Company.

Bradford, R. B. (2008). An empirical study of required dimensionality for large-scale latent semantic indexing applications. *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, 153-162.

Brill, E., & Pop, M. (1999). Unsupervised learning of disambiguation rules for part-of-speech tagging. In S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann, & D. Yarowsky (Eds.), *Natural Language Processing Using Very Large Corpora* (pp. 27-42), Dordrecht: Kluwer Academic Publishers.

Bullinaria, J., & Levy, J. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods, 39*(3), 510–526.

Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M., Braden-Harder, L., & Harris, M. D. (1998). Automated scoring using a hybrid feature identification technique. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, 1,* 206-210.

Burstein, J., Claudia, L., & Richard, S. (2001). Automated evaluation of essays and short answers. *Proceedings of 5th Computer Assisted Assessment Conference,* 41-45.

Burstein, J. (2003). The E-rater scoring engine: Automated essay scoring with natural language processing. In M. D. Shermis & J. Burstein (Eds.), *Automated essay scoring: A cross-disciplinary perspective* (pp. 113-121). Mahwah, NJ: Lawrence Erlbaum Associates.

Cangelosi, R., & Goriely, A. (2007). Component retention in principal component analysis with application to cDNA microarray data. *Biology Direct, 2*(1), 2-2.

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research, 1*(2), 245-276.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research, 16,* 321–357.

Chomsky, N. (1957). *Syntactic structures*. Oxford, England: Mouton.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Oxford, England: M.I.T. Press.

Chung, G. K. W. K., & O'Neil, H. F. (1997). *Methodological approaches to online scoring of essays*. Los Angeles, CA: Center for the Study of Evaluation, National Center for Research on Evaluation, Standards, and Student Testing, Graduate School of Education & Information Studies, University of California, Los Angeles.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing. *Proceedings of the 25th International Conference on Machine Learning, 20*(1), 160–167.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems, 2*(4), 303-314.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

Downing, S. M., & Haladyna, T. M. (2006). *Handbook of test development.* Mahwah, NJ: Lawrence Erlbaum Associates.

Drasgow, F., Luecht, R., & Bennett, R. (2006). Technology and testing. In R. L. Brennan (Ed.), *Educational measurement* (4th Ed.) (pp. 471-516). Westport, CT: Praeger.

Dreyfus, S. E. (1990). Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure. *Journal of Guidance, Control, and Dynamics, 13*(5), 926-928.

Du, K., & Swamy, M. (2019). *Neural networks and statistical learning* (2nd Ed.). London, UK: Springer.

Dumais, S. T. (1994). Latent Semantic Indexing (LSI) and TREC-2. *Proceedings of Text Retrieval Conference,* 105–115.

Eliason, S. R. (1993). *Maximum likelihood estimation: Logic and practice.* Newbury Park: Sage.

Embretson, S. E., & Reise, S. (2000). *Item Response Theory for psychologists*. Mahwah, NJ: Lawrence Erlbaum Associates.

Embretson, S. E. (2002). Generating abstract reasoning items with cognitive theory. In S. H. Irvine, & P. C. Kyllonen (Eds.), *Item generation for test development* (pp. 219-250).

Mahwah, NJ: Lawrence Erlbaum Associates.

Embretson, S. E., & Yang, X. (2007). Automatic item generation and cognitive psychology. In C. R. Rao, & S. Sinharay (Eds.), *Handbook of Statistics: Psychometrics, Volume 26* (pp. 747-768). North Holland, UK: Elsevier.

Embretson, S. E., & Kingston, N. M. (2018). Automatic Item Generation: A more efficient process for developing mathematics achievement items. *Journal of Educational Measurement, 55*(1), 112-131.

Fernandez-Beltran, R., & Pla, F. (2018). Prior-Based Probabilistic Latent Semantic Analysis for multimedia retrieval. *Multimedia Tools and Applications, 77*(13), 16771-16793.

Fillmore, C. J. (1968). The case for case. In E. Bach, & R. T. Harms (Eds.), *Universals in linguistic theory* (pp. 1-88). New York, NY: Holt, Rinehart, and Winston.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks, 2*, 183-192.

Furnas, G. W., Deerwester, S., Dumais, S. T., Landauer, T. K., Harshman, R. A., Streeter, L. A., & Lochbaum, K. E. (1988). Information retrieval using a Singular Value Decomposition model of latent semantic structure. *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 465-480.

Foltz, P.W., Laham, D., and Landauer, T. K. (1999). The Intelligent Essay Assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, *1*(2). http://imej.wfu.edu/articles/1999/2/04/ index.asp.

Gierl, M. J., & Haladyna, T. M. (2013). *Automatic item generation: Theory and practice*. New York, NY: Routledge.

Gibbons, R. D., Weiss, D. J., Frank, E., & Kupfer, D. (2016). Computerized adaptive diagnosis and testing of mental health disorders. *Annual Review of Clinical Psychology, 12*(1), 83-104.

Gierl, M. J., Zhou, J., & Alves, C. (2008). Developing a taxonomy of item model types to promote assessment engineering. *Journal of Technology, Learning, and Assessment, 72*(2), 1-52.

Gierl, M. J., & Lai, H. (2013). Using automated processes to generate test items. *Educational Measurement: Issues and Practice, 32*, 36-50.

Gierl, M. J., & Lai, H. (2015). Using automated processes to generate test items and their associated solutions and rationales to support formative feedback. *Interaction Design and*

Architecture, 25, 9-20.

Gorin, A., & Mammone, R. J. (1994). Introduction to the special issue on neural networks for speech processing. *IEEE Transactions on Speech and Audio Processing, 2*(1), 113-114.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States, 101*(14), 5228.

Han, W., Huang, Z., Li, S., & Jia, Y. (2019). Distribution-sensitive unbalanced data oversampling method for medical diagnosis. *Journal of Medical Systems, 43*(2), 1-10.

Haykin, S. S. (1999). *Neural networks: A comprehensive foundation* (2nd Ed.). Upper Saddle River, NJ: Prentice Hall.

Hardle, W., & Simar, L. (2012). *Applied multivariate statistical analysis* (3rd Ed.). New York: Springer.

Harris, Z. S. (1954). Distributional structure. *Word, 10*(2-3), 146–62.

Harris, Z. S. (1968). *Mathematical structures of language*. New York: Interscience Publishers.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd Ed.). New York: Springer.

Heaton, J. (2008) *Introduction to Neural Networks with Java* (2nd Ed.). Chesterfield, MO: Heaton Research Inc..

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science, 349*(6245), 261-266.

Hofmann, T. (1999). Probabilistic Latent Semantic Indexing. *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, Berkeley, CA, 50–57.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feed-forward networks are universal approximators. *Neural Networks, 2*, 359-366.

Hornik, K. (1991). Approximation capabilities of multilayer feed-forward networks. *Neural Networks, 4*, 251-257.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (3rd Ed.). Hoboken, New Jersey: Wiley.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(1), 4-37.

Jolliffe, I. T. (2002). *Principal Component Analysis*. New York, NY: Springer.

Jurafsky, D., & Martin, J. H. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, N.J: Prentice Hall.

Kane, M. T. (2006). Validation. In R. L. Brennan (Ed.), *Educational measurement* (4th Ed.) (pp. 17-64). Westport, CT: Praeger.

Ke, J., & Liu, X. (2008). Empirical analysis of optimal hidden neurons in neural network modeling for stock prediction. *Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, *2*, 828–832.

Kingma, D. P., & Ba, J. (2015). *Adam: A method for stochastic optimization*. Paper presented at the 3rd International Conference for Learning Representations, San Diego, CA.

Komori, O., & Eguchi, S. (2019). *Statistical methods for imbalanced data in ecological and biological studies*. Minatoku, Tokyo, Japan: Springer Japan.

Kukich, K. (2000). Beyond automated essay scoring. In M. A. Hearst (Ed.), *The debate on automated essay grading, IEEE Intelligent Systems and their Applications* (pp. 27-31), *15*(5), 22-37.

Kurdi, M. Z. (2016). *Natural language processing and computational linguistics*. Hoboken, NJ: John Wiley and Sons.

Lai, H., & Becker, K. A. (2010). *Using Artificial Neural Network for enemy item detection.* Paper presented at the Annual Meeting of National Council on Measurement in Education, Denver, CO.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review, 104*(2), 211-240.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to Latent Semantic Analysis. *Discourse Processes, 25*, 259-284.

Landauer, T. K., Laham, D., & Foltz, P. W. (2003). Automated essay scoring and annotation of essays with the Intelligent Essay Assessor. In M. D. Shermis, & J. C. Burstein (Eds.), *Automated essay scoring: A cross-disciplinary perspective* (pp. 87-112). Mahwah, NJ: Lawrence Erlbaum Associates.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by nonnegative matrix factorization. *Nature, 401*, 788–791.

Lesk, M. E. (1969). Word-word associations in document retrieval systems. *American*

Documentation, 20(1), 27-38.

Levy, J. P., & Bullinaria, J. A. (2001). Learning lexical properties from word usage patterns: Which context words should be used? *Proceedings of the 6th Neural Computation and Psychology Workshop, 273–282*.

Li, F., Shen, L., & Bodett, S. (2012). *Can enemy items be automatically identified?* Paper presented at the Annual Meeting of the National Council on Research in Education, Vancouver, BC, Canada.

Liddy, E. D. (2009). Natural language processing for information retrieval. In M. J. Bates, & M. N. Maack (Eds.), *Encyclopedia of library and information science* (3rd Ed.). Boca Raton, FL: CRC Press.

Liu, M., Rus, V., & Liu, L. (2018). Automatic Chinese multiple choice question generation using mixed similarity strategy. *IEEE Transactions on Learning Technologies*, *11*(2), 193-202.

Lowe, W. (2001) Towards a theory of semantic space. *Proceedings of the 23rd Annual Conference of the Cognitive Science Society,* 576–581.

Malhotra, R., & Kamal, S. (2019). An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data. *Neurocomputing*, *343*, 120-140.

Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology, 5*(4), 115-133.

McLeod, J., Butterbaugh, D., Masters, J., & Schaper, E. (2015). *Predicting item difficulty by analysis of language features*. Paper presented at the Annual Meeting of National Council on Measurement in Education, Chicago, IL.

MetaMetrics Inc. (2020) *About Lexile measures for reading*, Retrieved September 27, 2020, from https://lexile.com/educators/understanding-lexile-measures/about-lexile-measures-for-reading/

Mitkov, R., & Ha, L. A. (2003). Computer-Aided generation of multiple-choice tests. *Proceedings of the HLT/NAACL 2003 Workshop on Building Educational Applications using Natural Language Processing,* Edmonton, Canada, 17-22.

Mitkov, R., Ha, L. A., Varga, A., & Rello, L. (2009). Semantic similarity of distractors in multiple-choice tests: Extrinsic evaluation. *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, Athens, Greece, 49–56.

Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). Efficient estimation of word representations in vector space. *Proceedings of the International Conference on Learning Representations*, 1–12.

Muckle, T., & Becker, K. A. (2018). *Impact of enemy items and repeat-test masking on computerized adaptive testing*. Paper presented at the Annual Meeting of National Council on Measurement in Education, New York, NY.

Myers, M. (2003). What can computers and AES contribute to a K-12 writing program? In M. D. Shermis, & J. C. Burstein (Eds.), *Automated essay scoring: A cross-disciplinary perspective* (pp. 3-18). Mahwah, NJ: Lawrence Erlbaum Associates.

Pado, S., & Lapata, M. (2004). Dependency-Based construction of semantic space models. *Computational Linguistics, 33*, 161–199.

Page, E. B. (1966). The imminence of grading essays by computer. *Phi Delta Kappan, 47*(5), 238-243.

Page, E. B. (2003). Project Essay Grade: PEG. In M. D. Shermis, & J. C. Burstein (Eds.), *Automated essay scoring: A cross-disciplinary perspective* (pp. 39-50). Mahwah, NJ: Lawrence Erlbaum Associates.

Pampel, F. C. (2000). *Logistic regression: A primer*. Thousand Oaks, CA: Sage.

Pantel, P., & Lin, D. (2002). Discovering word senses from text. *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Edmonton, AB, Canada, 613-619.

Peng, F., Xiao, L., Qian, H., & Woo, Ada. (2018). *Automatic detection of enemy item pairs using Latent Semantic Analysis.* Paper presented at the Annual Meeting of National Council on Measurement in Education, New York, NY.

Peng, F., Swygert, K. A., & Micir, I. (2019). *Automatic enemy item detection using natural language processing*. Paper presented at the 2019 Annual Meeting of National Council on Measurement in Education, Toronto, ON, Canada.

Pommerich, M., & Segall, D. O. (2008). Local dependence in an operational CAT: Diagnosis and implications. *Journal of Educational Measurement, 45*(3), 201–223.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., & Welling, M. (2008). Fast collapsed Gibbs sampling for Latent Dirichlet Allocation, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, 569–577.

Quillan, R. (1963). *A notation for representing conceptual information: An application to semantics and mechanical English paraphrasing.* Santa Monica, CA: Systems

Development Corp.

Ruder, L. & Gagne, P. (2001). An overview of three approaches to scoring written essays by computer. *Practical Assessment, Research & Evaluation, 7*(26). https://pareonline.net/getvn.asp?v=7&n=26

Ruge, G. (1992). Experiments on linguistically-based term associations. *Information Processing and Management, 28*(3), 317-332.

Rumelhart, D. E., Hinton, G. E., & Williams, R.J. (1986). Learning internal representation by error propagation. In D. E. Rumelhart, & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (pp. 318–362). Cambridge, Massachusetts: The MIT Press.

Salton, G. (1968). *Automatic information organization and retrieval*. New York: McGraw-Hill.

Salton, G., Wong, A., & Yang, C. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613-620.

Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Reading, MA: Addison-Wesley.

Selmic, R. R., & Lewis, F. L. (2002). Neural network approximation of piecewise continuous functions: Application to friction compensation. *IEEE Transactions on Neural Networks, 13*(3), 745-751.

Sheehan, K. M., Kostin, I., & Persky, H. (2006*). Predicting item difficulty as a function of inferential processing requirements: An examination of the reading skills underlying performance on the NAEP Grade 8 Reading Assessment*. Paper presented at the Annual Meeting of the National Council on Measurement in Education, San Francisco, CA.

Shermis, M. D., & Burstein, J. (Eds.). (2003). *Automated essay scoring: A cross-disciplinary perspective*. Mahwah, NJ: Lawrence Erlbaum Associates.

Shermis, M. D., & Burstein, J. (2013). *Handbook on automated essay evaluation: Current applications and new directions*. New York, NY: Routledge.

Shin, J., Guo, Q., & Gierl, M. J. (2019). Multiple-Choice item distractor development using topic modeling approaches. *Frontiers in Psychology, 10*, 825-825.

Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. In T. K. Landauer, D. S. McNamara, S. Dennis, & W. Kintsch (Eds.), *Handbook of latent semantic analysis* (pp. 427–448). Lawrence Erlbaum Associates Publishers.

Susanti, Y., Iida, R., Tokunaga, T. (2015). Automatic generation of English vocabulary tests, *Proceedings of the 7th International Conference on Computer Supported Education,*

Setubal: INSTICC, 77–87.

Tang, J., Meng, Z., Nguyen, X., Mei, Q., & Zhang, M. (2014). *Understanding the limiting factors of topic modeling via posterior contraction analysis.* Proceedings of the 31st International Conference on Machine Learning (ICML 2014), Beijing, China, 190-198.

Turney, P. D. (2006). Similarity of semantic relations. *Computational Linguistics, 32*(3), 379-416.

Van der Linden, W. J., & Glas, C. A. W. (2010). *Elements of adaptive testing.* New York, NY: Springer.

VeldKamp, B. P., & Van der Linden, W. J. (2010). Designing item pools for computerized adaptive testing. In W. J. Van der Linden, & C. A. W. Glas (Eds.), *Elements of adaptive testing* (pp. 149-166). New York, NY: Springer.

Von Davier, M. (2018). Automated item generation with Recurrent Neural Networks. *Psychometrika, 83*(4), 847-857.

Wei, W., Li, J., Cao, L., Ou, Y., & Chen, J. (2013). Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web, 16*(4), 449-475.

Weir, J. B., Dallas, A. & Goodman, J. (2018). *Enemy item detection with natural language processing: Latent Dirichlet Allocation.* Paper presented at the Annual Meeting of National Council on Measurement in Education, New York, NY.

Weir, J. B. (2019). *Enemy item detection using data mining methods*. (Unpublished doctoral dissertation). The University of North Carolina at Greensboro, Greensboro, NC.

Weizenbaum, J. (1966). ELIZA: A computer program for the study of natural language communication between man and machine. *Communications of the Association for Computing Machinery, 9*(1), 36-45.

Werbos, P. J. (1994). *The roots of backpropagation: From ordered derivatives to neural networks and political forecasting.* New York: J. Wiley & Sons.

Wilks, Y. (1972). *Grammar, meaning and the machine analysis of language.* London: Routledge and Kegan Paul.

Winograd, T. (1971). *Procedures as a representation for data in a computer program for understanding natural language.* M.I.T. Artificial Intelligence Laboratory Project MAC-TR-84.

Woo, A., & Gorham, J. L. (2010). Understanding the impact of enemy items on test validity and measurement precision. *Journal of Clear Exam Review, 21*(1), 15-17.

Woods, K., Doss, C., Bowyer, K., Solka, J., Priebe, C., & Kegelmeyer, W. (1993). Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography, *International Journal of Pattern Recognition and Artificial Intelligence, 7*(6), 1417–1436.

Woods, W. A. (1970). Transition network grammars for natural language analysis. *Communications of the Association for Computing Machinery*, *13*(10), 591-606.

Yen, W. M. (1984). Effects of local item dependence on the fit and equating performance of the three-parameter logistic model. *Applied Psychological Measurement*, *8*(2), 125–145.

Zhang, G. P. (2000). Neural networks for classification: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, 30*(4), 451-462.

Zhao, W., Chen, J. J., Perkins, R., Liu, Z., Ge, W., Ding, Y., & Zou, W. (2015). A heuristic approach to determine an appropriate number of topics in topic modeling. *BMC Bioinformatics*, *16*(13), 1-10.

VITA

Fang Peng

Education

2020 (Expected)	Ph.D. in Educational Psychology University of Illinois at Chicago, IL, USA
2013	M.A. in Language Testing and Assessment Tsinghua University, Beijing, China
2010	B.A. in English Language and Literature Tsinghua University, Beijing, China

Research

Aug. 2014 – Jan. 2020	Graduate Research Assistant *University of Illinois at Chicago*
Jun. – Jul. 2018	Psychometric Intern *National Board of Medical Examiners (NBME)*
Jun. – Jul. 2017	Psychometric Intern *National Council of State Boards of Nursing (NCSBN)*

Publication

Gordon, R. A., & Peng, F. (2020). Evidence Regarding Domains of the CLASS PreK in Head Start Classrooms. *Early Childhood Research Quarterly, 53, 23-39.*

Fujimoto, K. A., Gordon, R. A., & Peng, F. (2018). Examining the Category Functioning of the ECERS-R across Eight Datasets. *AERA Open, 4*(1), 1-16.

Peng, F. (2013). Considerations in Developing an Integrated Framework for EAP Writing, *Foreign Language Teaching and Research, 27*(3), 114-136.

Presentations

Peng, F. "Detection of Enemy Item Pairs Using Natural Language Processing." Upcoming presentation at the 2019 Annual Meeting of the National Council on Measurement in Education (April 4-9, 2019, Toronto, Ontario, Canada)

Peng, F. "Automatic Detection of Enemy Item Pairs Using Latent Semantic Analysis." Presented at the 2018 Annual Meeting of the National Council on Measurement in Education (April 14-16, 2018, New York, NY)

Peng, F., & Becker, K.A. “Recovery of Theta from Early Misfit in Multistage Adaptive Testing.” Presented at the 2018 Annual Meeting of the American Educational Research Association (April 13-17, 2018, New York, NY)

Peng, F. “Automatic Detection of Enemy Items in the NNAAP Operational Item Pool.” Presented at the 2017 NCSBN Joint Research Committee Meeting (August 25, 2017, Chicago, IL)

Gordon, R. A., Hofer, K.G., Peng, F., Gaur, D., & Lambouths, D. “ECERS-R Quality and Children’s Vocabulary: Examining Variation by Geography and Demographics Using Large Scale Meta-Synthesis.” Presented in the Society for Research in Child Development 2017 Biennial Meeting (April 6-8, 2017, Austin, TX)

Gordon, R. A., Hofer, K.G., Peng, F., Crabbe, R., & Lambouths, D. “Assuring quality preschool: Where are we and where do we need to go?” Presented in the American Educational Research Assocation Presidential Session on Universal Preschool: What Have We Learned, and What Does It Mean for Practice and Policy? (Chair: Rachel A. Gordon, Discussant: Libby Doggett). (April 6, 2014, Philadelphia, PA)

Peng, F. (2012). “An Analysis of Language Anxiety, Self-esteem, Risk-taking, and Sociability in the University EFL Classroom.” Presented in the 7th International Symposium on Teaching English at Tertiary Level (October 14, 2011, Hong Kong, China)

Peng, F. (2011). “The Use of an Automated Writing Evaluation Program from EFL Learners’ Perspective.” Presented in the the 6th International Symposium on Teaching English at Tertiary Level (October 16, 2010, Beijing, China)