Journal of Applied Testing Technology, Vol 23(Special Issue 1), 41-52, 2022

Identifying Enemy Item Pairs using Natural Language Processing

Kirk A. Becker[1*] and Shu-chuan Kao[2]

[1]Senior Research Scientist, Pearson VUE, United States; Kirk.Becker@Pearson.com
[2]Senior Manager, Measurement and Testing, Examinations, National Council of State Boards of Nursing, United States; skao@ncsbn.org

Abstract

Natural Language Processing (NLP) offers methods for understanding and quantifying the similarity between written documents. Within the testing industry these methods have been used for automatic item generation, automated scoring of text and speech, modeling item characteristics, automatic question answering, machine translation, and automated referencing. This paper presents research into the use of NLP for the identification of enemy and duplicate items to improve the maintenance of test item banks. Similar pairs of items can be identified using NLP, limiting the number of items content experts must review to identify enemy and duplicat items. Results from multiple testing programs show that previousely unidentified enemy pairs can be discovered with this method.

Keywords: Cosine Similarity, Enemy Items, Item Banking, Natural Language, Processing, Text Indexing

1. Historical Use of NLP in Testing

Beginning in the 1950s, a new discipline arose in computer science devoted to making computers understand natural language - language spoken or written by humans for general-purpose communication. The goal of NLP is to convert samples of human language into more formal representations that are easier for programs to manage. Since the 1980s, research interest began to focus on systems that could deal with written language in paragraphs instead of with typed interactions by computer users. At the same time, with the idea of relaxing the goal to process every word of the input as deeply as necessary to produce an understanding of the sentence as a whole, researchers started to accept the value of "partial understanding" of the sentence as a more feasible and useful goal for practical work. (For more detail on the history of NLP, see Bates, 1995.)

The field of NLP within computer science has developed methods for indexing, categorizing, summarizing, and interpreting large numbers of text documents. Given the fact that the testing industry works predominantly with large collections of text, which require people to write, reference, evaluate, classify, edit, score, and analyze information, NLP seems like an obvious choice for working with those data.

An early use of NLP in testing was developed to score essay exams and to overcome the drawbacks of human raters (such as fatigue, subjectivity, time, and cost) to score essay exams. Page (1966) developed the first computer software for essay scoring, focusing on the length of paragraphs, average sentence length, and the counts of textual units. Subsequently many engines were developed for the automated scoring of text responses including open-ended questions (Attali *et al.,* 2008; Sukkarieh & Stoyanchev, 2009), essays (Attali & Burstein, 2006; Burstein *et al.,* 1998; Chodorow & Burstein, 2004; Deane *et al.,* 2011; IntelliMetric Engineer, 1997; Landauer & Dumais, 1997; Landauer *et al.,* 1998; Page, 1994, 2003; Rudner *et al.,* 2006), and speech (Xi *et al.,* 2008). With the strength of NLP, these scoring engines are able to capture finer features, such as dimensions of content, organization, style, sentence structure, etc. There are still debates concerning the validity of the automated scoring

(Bennett, 2004; Cheville, 2004), which should lead to the generation of better scoring tools.

In addition to automated scoring, automated item generation has been an increasingly accepted research area for NLP. With the intention of generating items in an efficient way, researchers (Bejar, 1996; LaDuca, Staples, Templeton & Holzman, 1986; Embretson & Yang, 2007; Gierl & Lai, 2012) proposed item modeling as a construct-driven approach to test development. For large-scale testing, some item models are more statistics-driven (e.g., Glas & van der Linden, 2003) and others are more content-driven (e.g., Bejar *et al.,* 2003). Each item model provides templates that allow decomposition of knowledge or skills and identification of the key components that constitute meaningful new items. Later, Automatic Item Generation (AIG) was developed to produce items algorithmically using the item modeling techniques (Bejar *et al.,* 2003; Higgins, 2007; Irvine, & Kyllonen, 2002; Gierl *et al.,* 2008). Beyond using the template system and string manipulation, recent developments in automated item generation (Brown *et al.,* 2005; Deane & Sheehan, 2003; Hombo & Dresher, 2001, Hoshino & Nakagawa, 2005; Mitkov *et al.,* 2005; Shin *et al.,* 2019) have combined the strength of NLP to create items with more flexibility in linguistics and to predict items' statistical properties (Belov and Knezevich, 2008). It should be noted that no matter how powerful the AIG models are, human review is currently needed to assure the item quality in most instances.

1.1 Use of NLP in Item Bank Management

What is not found in the published NLP research is the application to item bank management. After large numbers of test items are created, item bank management becomes a challenge. Item bank maintenace, which includes identifing similar and duplicate items, categorizing new and operational items by the test blueprint, and evaluating items for content currency is labor-intensive and can be subjective. Large item banks present logistical problems to the item development process that are difficult to solve.

The authors are aware of conference presentations looking at methods for indentifying enemy items within item banks (Becker & Kao, 2009; Lai & Becker, 2010; Li & Shen, 2011; Peng *et al.,* 2018; Peng *et al.,* 2019; Fu & Han, 2022; Li, Hu, & Wilmurth, 2022). Following this application of text similarity to enemy detection, the automated item generation literature began to use text similarity to evaluate items created from item templates (Gierl & Lai, 2013). Additionally, two dissertations have explored the application of topics models (Weir, 2019) and latent semantic analysis (Peng, 2020) to enemy item identification.

1.2 Historical Methods for Detection of Enemy and Duplicate Items

In large-scale test- and item-development, enemy item sets arise due to constraints from inter-item dependencies (Yen, 1993; Veldkamp & van der Linden, 2000). For example, when an item requires the same or similar knowledge to another item already administered, that item is deemed unfit for use, as the results from that item will be dependent on the knowledge already demonstrated by the examinee. One item might also contain information pertinent to the answer of another item, referred to as cuing, which provides an advantage to test takers who receive both items. This dependence between a pair of items is known as an enemy item set. In the context of a large item bank, many sets of enemy items may exist for a variety of reasons. Pommerich and Seagall (2006) demonstrate the effect that these sorts of dependencies can have on IRT calibrations.

Enemy item sets have historically been identified during item writing, item review and test form assembly by Subject Matter Experts (SMEs). SMEs evaluate item and test forms to identify content overlap, cuing, or other content features they believe will lead to local item dependence and reduce construct representation. This is a manual and tedious process, especially for large computer-based test banks involving an enormous amount of text. Processing all of that text requires large numbers of people who write, read through, evaluate, classify, edit, score, and analyze (Downing & Haladyna, 2006; Haladyna, 1999; Roid & Haladyna, 1982). Not only is this process time-consuming and resource-intensive, but it is also subjective and error-prone. While making a judgement on how similar two test items are, multiple SMEs can introduce inconsistent results and rater bias. As item banks grow, identification of duplicate and enemy items becomes quadratically less efficient (the number of item pairs is N*(N-1)/2), where N is the number of items in the item bank (e.g., comparing item 1 to items 2-100, then item 2 to items 3-100, etc.). In practice a review of items for enemies does not typically involve comparing each item to every

other item, making the review somewhat faster but also less accurate. Based on interviews with test development professionals, reviewing 100 items for enemy relationships takes between 2 and 3 hours. In our experience, the ability of item developers to identify enemy items is especially problematic as the number of items in a content area increases above 50-100 items, although grouping items by content area or keywords allows for the review of longer test forms or item sets (enemy relationships frequently occur within content areas).

This paper presents a proof-of-concept for the application of NLP techniques to one aspect of item bank management: identifying duplicate and enemy items. Identifying enemy relationships is important for some test designs, and critical for others. Fixed-form tests benefit from accurate identification of enemy item pairs because one or multiple forms can be more appropriately created (either automatically or manually) when that information is available (Luecht, 1998; Swanson & Stocking, 1993; van der Linden, 1998). Even though fixed-form tests are typically reviewed prior to administration, human reviewers may not identify enemy pairs, or the identification of enemy pairs may delay the publication process. In situations where the specific items or sets of items are selected at the time of testing by an algorithm, (e.g., LOFT or CAT) the identification of enemy pairs is critical because test takers see different items and therefore the administration of enemy item pairs on a test differentially advantages or disadvantages some test takers.

1.3 Theoretical Framework for Similarity

Similarity is a sophisociated subset of NLP, a concept in linquistics and information theory. An intuitive definition for similarity is that the more objects A and B have in common, the greater similarity exists between A and B. Mathematical algorithms have been developed to provide easier and more precise of similarity. A number of indices have been developed that are applicable to use with test items. Manning and Schütze (1999) discuss 5 coefficients for comparing the similarity of two text documents (matching, Dice, overlap, Jaccard, and cosine) which provide options for weighting or penalizing different features of item pairs. The Dice coefficient, for example, decreases when the lengths of two documents (items) are very different.

The research presented in this paper makes use of the cosine similarity index, which is appropriate when the texts being compared may vary in length. While these methods allow for a level of evaluation that's especially valuable for large-scale programs, they will also prove guidance to programs with small item banks and low candidate volumes. At present, similarity procedures are not meant to replace humans, but to make them more efficient and accurate in the duplicate/enemy item identification process.

In large item banks the cosine similarity index provides a promising method for flagging item pairs that should be considered enemies either due to content overlap or for cuing. Additionally this application of the cosine similarity index also provides a method for evaluating whether newly written items are duplicates or substantially the same as existing items already in the item bank. This is a common problem when multiple item writers utilize the same reference textbooks.

The goal of this paper is to demonstrate the application of a relatively simple approach for the detection of enemy items. This method can be applied directly to a single pair of test items or a large item bank. No additional information is necessary besides the content of the test items. The cosine function is available in statistical platforms as well as Excel. While there are more advanced methods for parsing and comparing text, the methods included in this paper are both easily applied and produce actionable information. Detailed surveys of additional text similarity method are available in Gomaa & Fahmy (2013) and Wang & Dong (2020), which may be of interest for future research on this topic.

2. Methods

Prior to calculating the cosine similarity index, the text of a test item must be processed and parsed to create a document/word matrix. First the item content is exported from the item bank. Then, punctuation, numbers, and case are removed. While the authors made use of a Visual Basic program to complete this and subsequent text formatting, the steps could be easily accomplished with a variety of different software solutions (e.g., "replace' in Excel, "gsub" in R, or "numpy" in Python).

2.1 Stop words

Next, stop words, which are common articles, pronouns, adjectives, adverbs, and prepositions (e.g., "the," "a," "and," etc.) are removed. For this research a list of 180 basic stop

words were used. These words are commonly used in a language, but not to contribute to the meaning of the text and will therefore inflate the similarity between unrelated test items. The list of stop words can be customized based on an SME review of results (e.g., identification of words from item templates, or words which are common in a given content domain, such as "agent" in the insurance domain). Certain weighting schemes, such as *tf idf* (Larkey & Croft, 2003), provide lower weights for words that appear in a large number of items and may supplement or replace this step.

2.2 Stemming

All words remaining after stop word removal are transformed via a process called "stemming". Stemming is the action of reducing word tenses and forms to their common root. For example, a stemming program would convert the words "respect," "respecting," "respects," and "respectful" to "respect". Stemming cuts down on unnecessary word variation, making all subsequent processes more efficient. We used the Visual Basic implementation of the Porter stemmer written by Navonil Mustafee while at Brunel University (Porter Stemming Algorithm, n.d.). It provides a well-documented method for automatically removing suffixes from words. Porter's algorithm (Porter, 1980) makes use of an explicit list of suffixes and applies criteria to determine when they can be removed from a word to leave a valid stem. The accuracy of the Porter stemmer, like that of all stemming programs, is less than 100%. For example, the Porter stemmer treats the "er" in "wander" as a suffix, even though it is part of the stem. This is only a problem when the stemmer treats unrelated words as identical due to the stemming process (e.g., "abrasion" and "abrasive" should be separate words), and the item bank contains instances of both words. An alternative to stemming is lemmatization, which groups together multiple inflections of a word and serves a similar purpose to stemming.

2.3 Semantic Space

All stemmed words from a set of test items form the dimensions of the "semantic space." The primary idea underpinning the semantic vector is that words and concepts can be represented by points in a mathematical, multidimensional space. Each dimension of the vector space corresponds to a stemmed word. The representation of an item is the count of each stemmed word contained in the item. For dimensions representing stemmed words not contained in an item, the vector has a value of zero. In this high-dimensional vector space, the spatial representation is derived from the text in such a way that concepts with similar or related meanings are near one another (Widdows & Ferraro, 2008). Vectors that contain the same words or content should be roughly parallel, while vectors that relate to different content should be oblique. When applying the idea of semantic vector space in testing, each item can be represented as an N-dimensional vector within this space, where N is the number of unique stemmed words (excluding stop words). Consequently, the comparison of item content can be achieved by comparing items' N-dimensional vectors.

2.4 Angular Distance

The concept of angular distance is employed to signify the similarity or dissimilarity of two semantic vectors. Angular distance is the size of the angle between two semantic vectors originating from the origin and pointing towards two points. The degree to which two vectors are parallel can be quantified through the cosine of the angle between the vectors.

$$\cos \alpha = \frac{\overline{a} \cdot \overline{b}}{||a||\ ||b||} \tag{1}$$

If the angle is 0 degrees (for perfectly parallel vectors), the cosine of the angle is 1. If the angle between vectors is 90 degrees, the cosine of the angle is 0. In short, the smaller the angle, the greater the similarity and the higher the cosine similarity will be. While it is possible for angles to have negative cosines, the manner in which semantic vectors are defined for this analysis precludes negative cosines, resulting in a range of 0 to 1 for the cosine similarity index.

2.5 Example Matrix and Items

Table 1 shows an example of text that has been parsed. Each row represents a test item while each column represents a word. The values in each cell of the matrix represent the count of the word within the item. Table 1 shows an example for two sample test items:

(I1) "Which Europeans were the first to establish a colony in the area that is now New York State?"
(I2) "New Amsterdam was the capital of a Dutch settlement in the area that is now:"

The word "Europeans" appears once in item I1 and never in item I2, while "Amsterdam" appears once in item I2 and never in item I1. This method treats each text response as a collection of disassociated word variables or "bag of words" (Steyvers and Griffiths, 2004).

Table 1. Example of text parsing after stemming and stopping

item	Amsterdam	Area	Capital	Colony	Dutch	Establish	Europeans	First	New	Now	of	Settlement	State	York
I1	0	1	0	1	0	1	1	1	1	1	0	0	1	1
I2	1	1	1	0	1	0	0	0	1	1	1	1	0	0

Figure 1 provides example cosine similarity indices for two related test items and two unrelated test items. The *total similarity* is calculated based on words contained in the stem and the options, the *stem similarity* is calculated based on words in the item stems only, and the *stem+key similarity* is based on words in the stem and words in the key (non-keyed options are excluded). The related items both contain words such as "snowmobile," "manufacture," "thief," "river," "falls," "minnesota," "arctic," and "cat," which leads to relatively high similarity values. The unrelated items have no words in common after stop words such as "which," "of," "the," "is," "to,"" were," "in," and "that" are removed, resulting in similarity values (cosines) of zero. The stem and the key together produce a higher similarity value than the total item due to the exclusion of distractors, which contain unique words ("Polaris," "Ski-doo," "ice cream," etc.).

Related Test items	Total similarity: 0.65 Stem similarity: 0.67 Stem + Key similarity: 0.82
Which snowmobile manufacturer is based in Thief River Falls, Minnesota? (Stem) **A. Arctic Cat (Key)** B. Polaris C. Ski-doo D. Yamaha	Arctic Cat, located in Thief River Falls Minnesota, manufactures: (Stem) A. heated cat houses B. ice cream. **C. snowmobiles. (Key)** D. winter camping gear.
Unrelated Test items	Total similarity: 0.00 Stem similarity: 0.00 Stem + Key similarity: 0.00
Which of the following is Kant's most important contribution to moral philosophy? (Stem) A. The hedonic calculus B. The categorical imperative (Key) C. The theory of virtues D. The doctrine of the mean	Which Europeans were the first to establish a colony in the area that is now New York State? (Stem) A. French **B. Dutch (Key)** C. Spanish D. English

Figure 1. Sample test items data.

Several item banks are used to demonstrate the application of cosine similarity to enemy item identification. The first is an item bank developed for regulatory licensure consisting of 451 items classified into 4 different content areas (the content areas had between 68 and 177 items). There were 185 enemy pairs previously identified by SMEs. For a separate organization, two different item banks from a medical certification program were also analyzed, program 1 with 2266 items and 355 enemy pairs, and program 2 with 2174 items and 4157 enemy pairs. Program 2 is making use of automatic item generation, which accounts for the large difference in number of existing enemy pairs. Finally, from a third professional licensure organization 5 item banks (with 1864, 1508, 2728, 2039, and 2191 items) were analyzed and results reviewed.

The enemy pairs identified through previous SME review of the regulatory item bank provides a benchmark of the relationship between enemy status and cosine similarity. The frequency of different ranges of cosine

similarity values for enemy pairs and for all item pairs in the regulatory item bank are presented in Table 2. The table contains a count of item pairs not currently identified as enemies and enemy pairs. Additionally, four other summary statistics are provided:

- Cumulative Non-Enemy is a count of the number of item pairs at or above a given range, showing the number of pairs that would be selected if that value were used to identify item pairs for review.
- The Cumulative Percent of Enemies is the percentage of currently identified enemies at or above a given value.
- The % of Enemies in Range is the count of currently identified enemy pairs divided by the total number of items at a specific range, which may estimate the frequency of unidentified enemy pairs within that range.
- Finally, the Cumulative % of Enemies at or Above Range is the cumulative number of enemies divided by the cumulative number of item pairs, which may provide a rough estimate of the percent of cumulative non-enemies in the pool that would be identified as enemies once reviewed.

The median similarity index between enemy pairs in the regulatory data is .57 (*ranging from .11 to 1*), while the median for non-enemy pairs is .18 (*ranging from 0 to .89*). In this item bank a total similarity greater than .6 would identify 46% of the known enemy pairs while requiring review of 920 pairs (9% of all pairs). At the high end (>.9), the cosine similarity index correctly identified 9 out of 9 enemy item pairs. Lowering the similarity threshold will increase the percentage of enemy pairs found, but also increase the number of pairs for SMEs to review. While Table 2 shows the descriptive statistics for an item bank with some known enemy items, there are also unidentified enemy enemy pairs in the item bank. The goal in selecting a similarity threshold value is to maximize the likelihood of finding enemies while limiting the number of items to review – in this case, a value of 0.65 or 0.7 would likely be appropriate. It may further be possible to limit the number of item pairs to review by restricting reviews to item pairs within content areas of the test, or to otherwise include content information in the pair selection process.

Table 2. Existing enemy pairs for regulatory program

Similarity Range	# Non-enemies	# Enemies	Cumulative Non-Enemies	Enemies as Percent of Range	Cumulative Percent of Known Enemies	Cumulative % of Enemies at or Above Range
1	0	2	0	100%	1%	100%
.9-99	0	7	0	100%	5%	100%
.8-.89	16	10	16	38%	10%	54%
.7-.79	157	48	173	23%	36%	28%
.6-.69	662	18	835	3%	46%	9%
.5-.59	2,400	31	3235	1%	63%	3%
.4-.49	6,595	28	9830	0%	78%	1%
.3-.39	13,647	23	23477	0%	90%	1%
.2-.29	20,820	13	44297	0%	97%	0%
.1-.19	25,432	5	69729	0%	100%	0%
.01-.09	19,986	0	89715	0%	100%	0%
0	11,125	0	100840	0%	100%	0%
Total	100,840	185				

For the professional programs two additional steps were taken. First, cosine similarity was calculated separately for different components of the test items. Cosines for the full item, stem only, key only, key plus stem, and non-keyed options were calculated. Second, these separate cosine values, along with information about content area, were entered into a linear regression to produce a weighted combination that best predicted existing enemy relationships. For these 2 programs we have feedback from the review of item pairs by subject matter experts.

Tables 3 and 4 provide the summary of existing enemy and non-enemy pairs for the three medical

programs. Non-enemy item pairs were selected from each item bank based on the total number of pairs and the cumulative percent of enemies for the regression probability. For program 1 this value was .05, resulting in 781 non-enemy pairs to review. For program 2 the value used was .1 resulting in 1179 non-enemy pairs to review. In operational settings decisions such as this will be based on resources available and characteristics of the item bank. The item pair review for program 1 identified 370 additional enemy pairs (47% of item pairs reviewed). The item pair review of program 2 identified 560 additional enemy pairs (47% of item pairs reviewed). The percent of enemies found within the item pairs reviewed for program 1 is similar to the cumulative percent of enemies above .05 (45%), while the cumulative percent for program 2 is higher (89%) likely due to the inclusion of AIG items.

Table 3. Existing enemy pairs for medical program 1

Similarity Range	# Non-enemies	# Enemies	% of Enemies in Range	Cumulative Non-Enemy	Cumulative Enemy	Cumulative Percent of Enemies	Cumulative % of Enemies at or Above Range
>.4	71	43	38%	71	43	12%	38%
.3 to .4	44	20	31%	115	63	18%	35%
.25 to .3	26	9	26%	141	72	20%	34%
.2 to .25	52	14	21%	193	86	24%	31%
.15 to .2	74	10	12%	267	96	27%	26%
.1 to .15	146	23	14%	413	119	34%	22%
.05 to .1	368	41	10%	781	160	45%	17%
.04 to .05	166	9	5%	947	169	48%	15%
.03 to .04	227	9	4%	1174	178	50%	13%
.02 to .03	448	27	6%	1622	205	58%	11%
01 to .02	1207	28	2%	2829	233	66%	8%
0 to .01	1337860	122	0%	1340689	355	100%	0%

Table 4. Existing enemy pairs for medical program 3

Similarity Range	# Non-enemies	# Enemies	% of Enemies in Range	Cumulative Non-Enemy	Cumulative Enemy	Cumulative Percent of Enemies	Cumulative % of Enemies at or Above Range
>.4	297	3578	92%	297	3578	86%	92%
.3 to .4	125	28	18%	422	3606	87%	90%
.25 to .3	138	37	21%	560	3643	88%	87%
.2 to .25	149	16	10%	709	3659	88%	84%
.15 to .2	175	21	11%	884	3680	89%	81%
.1 to .15	295	32	10%	1179	3712	89%	76%
.05 to .1	663	57	8%	1842	3769	91%	67%
.04 to .05	273	13	5%	2115	3782	91%	64%
.03 to .04	445	21	5%	2560	3803	91%	60%
.02 to .03	975	27	3%	3535	3830	92%	52%
01 to .02	2367	48	2%	5902	3878	93%	40%
0 to .01	2556186	279	0%	2562088	4157	100%	0%

For the last 5 item banks the logistic regression approach was also used. In this case, not only were the pre-existing non-enemy item pairs with high similarity flagged for review (See column Flag high in Table 6), but the existing enemy pairs with low similarity were also flagged for review (See column Flag Low in Table 6). Table 6 shows a summary of the results for these 5 programs. These results are more variable than the medical certification program results, with between 16% and 74% of flagged item pairs marked as enemies during the review process. These enemy pairs were missed by the existing process for handling item enemies. Similarly, low similarity item pairs previously identified as enemies had that status removed between 47% and 98% of the time. For all item banks over 100 previously unidentified enemy pairs were found through this process. Enemy pairs identified typically represent content overlap, although some pairs have duplicate content.

Table 5. Professional licensure SME review of flagged items

	Total Items	Pre-Existing Enemies	Potential Enemies Flagged	New Flagged Enemies	% Newly Flagged Enemies	Potential Non-Enemies Flagged	Non-Enemies Removed	% Confirmed Non-Enemies
Program 1	1864	734	858	133	16%	46	45	98%
Program 2	1508	155	402	138	34%	30	25	83%
Program 3	2728	740	550	270	49%	102	84	82%
Program 4	2039	988	408	300	74%	120	55	46%
Program 5	2191	1347	908	548	60%	232	108	47%

3. Discussion and Future Directions

One of the main purposes of this paper is to draw testing practictioners' attention to the potential advantages offered by NLP for evaluating item bank health and improving test development efficiency. Automating routine and repetitive tasks leads to higher accuracy and effeciency. The literature cited in this paper covers several approaches for identifying similar item pairs, including cosine similarity, topics models, program-specific ontologies, and machine learning. The calculation of cosine similarity is an accessible process that can identify a limited number of item pairs with a relatively high probability of an enemy relationship. This process also allows for fine-tuning and modification over time to improve accuracy (e.g., revising the stop word list, or incorporating logistic regression into the flagging process). Given the high cost of item development, methods for identifying duplicate content within large item banks will help focus resources on unique items rather than on common variants. Our results support the use of these methods for identifying content overlap.

When using text similarity to identify item pairs for human review, the information contained in tables 2-4 can be helpful for determining a threshold for selecting item pairs. Criteria such as enemies within range >10% and cumulative non-enemies at a manageable level (e.g., <500 pairs) can provide substantial value to an item bank while requiring only a few hours of review. In practice a cosine >0.7 for the combined text of stem and key tends to be a good starting value. Keeping track of previously reviewed items will also effectively reduce the workload for future review, as "not enemy" in most item banking systems could mean either not reviewed or reviewed and judged to be non-enemy pairs.

It would make sense for testing programs to evaluate their newly written items relative to their existing banks to determine if they are developing unique content or rehashing existing test items. While the test items analyzed for this research are multiple-choice in format, any item format consisting of text can be accomodated (e.g., multiple-select, constructed response prompts, etc.). Test items with substantial non-text content such as audio, video, graphics, or interactive elements will not lend themseves to these methods.

Additional research is needed to identify other characteristics of enemy item pairs that may be algorithmically identified. Electronic databases such as WordNet (Fellbaum, 1998) and UMLS (National Library of Medicine, 2021) provide access to term similarity metrics that may improve our ability to identify different

ways of asking the same question. Researchers (Li *et al.*, 2012; Belov & Kary, 2012) have used those databases with both stand-alone test items and test reading passages to accurately identify enemy relationships. Both Lai & Becker (2010) and Li *et al.*, (2012) made use of modeling methods including artificial neural networks and logistic regression to incorporate item similarity measures and other item meta data (item classifications, item type, etc.) as part of the enemy classification process. Other research might explore the use of item-bank specific stop words, removal of words associated with item templates, other similarity indices (Manning & Schütz, 1999; Lin & Hovy, 2003), or methods such as *tf idf* weighting (Larkey & Croft, 2003) that increase the similarity measure of documents with unusual terms compared with common terms. Other comparisons, such as the similarity between keys, options, and stems, as well as other NLP indices, may help to identify additional enemy pairs not flagged by the total item similarity (Lai and Becker, 2010; Li & Shen 2011). The cosine similarity index is not currently used to replace human reviewers for identifying enemy items but to provide a useful supplemental process that increases the identification of enemy items. Supplemental methods for identifying enemy items with lower cosine similarities are needed.

In addition to the topics covered in this article, other applications of NLP to test development are also needed. Processes to automatically generate multiple-choice items from source material have been developed (Mitkov *et al.*, 2005) and from Transformer architecture (Khan *et al.*, 2021) have also been demonstrated. Classifying items into content areas or cognitive levels (Becker & Kao, 2009), key varification through automatic question answering (Le An Ha & Yaneva, 2019; Minaee & Liu, 2017), and item parameter estimation (Benedetto *et al.*, 2020; Le An Ha *et al.*, 2019) will also prove useful to the test development process.

4. Conflict of Interest Disclosure

There were no external sponsors of this research.

5. References

Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater V.2.0. *Journal of Technology, Learning, and Assessment*, 4(3), 1-34.

Attali, Y, Powers, D., Freedman, N, Harrison, M., & Obetz, S. (2008). Automated Scoring of Short-Answer Open-Ended GRE Subject Test Items, ETS Research Report No. RR-08-20. Princeton, NJ: ETS https://doi.org/10.1002/j.2333-8504.2008.tb02106.x

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. https://doi.org/10.1073/pnas.92.22.9977 PMid:7479812 PMCid:PMC40721

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

Bejar, I. I. (1996). Generative response modeling: Leveraging the computer as a test delivery medium. Princeton, NJ: ETS. https://doi.org/10.1002/j.2333-8504.1996.tb01691.x

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), 3-28. https://doi.org/10.1002/j.2333-8504.2002.tb01890.x

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

Belov, D. I., & Kary, D. (2012, April). A heuristic-based approach for computing semantic similarity between single-topic texts. Paper presented at the annual meeting of the National Council on Measurement in Education, Vancouver, BC.

Benedetto, L., Cappelli, A., Turrin, R., Cremonesi, P. (2020). Introducing a Framework to Assess Newly Created Questions with Natural Language Processing. In: Bittencourt, I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds) Artificial Intelligence in Education. AIED 2020. *Lecture Notes in Computer Science*. Springer, Cham. pp. 43–54. https://doi.org/10.1007/978-3-030-52237-7_4 PMCid:PMC7334176

Bennett, R. E. (2004). Moving the Field Forward: Some Thoughts on Validity and Automated Scoring, ETS Research Memorandum No. RM-04-01 Princeton, NJ: ETS.

Brown, J., Firshkoff, G., & Eskenazi, M. (2005). Automatic question generation for vocabulary. assessment. Proceedings of HLT/EMNLP. Vancuver, Canada. https://doi.org/10.3115/1220575.1220678

Burstein, J., Braden-Harder, L., Chodorow, M., Hua, S., Kaplan, B., Kukich, K., Lu, C., Nolan, J., Rock, D., & Wolff, S. (1998). Computer analysis of essay content for automated score prediction: A prototype automated scoring system for GMAT analytical writing assessment, ETS Research

Rep. No. RR-98-15. Princeton, NJ: ETS. https://doi.org/10.1002/j.2333-8504.1998.tb01764.x

Cheville, J. (2004). Automated scoring technologies and the rising influence of error. *The English Journal*, 93(4), 47-52. https://doi.org/10.2307/4128980

Chodorow, M., & Burstein, J. (2004). Beyond essay length: Evaluating e-rater's performance on TOEFL essays, TOEFL Research Rep. No. RR-73. Princeton, NJ: ETS. https://doi.org/10.1002/j.2333-8504.2004.tb01931.x

Deane, P., & Sheehan, K. (2003). Automatic item generation via frame semantics: Natural language generation of math word problems. Princeton, NJ: ETS.

Deane, P., Quinlan T., & Kostin, I. (2011). Automated Scoring Within a Developmental, Cognitive Model of Writing Proficiency, ETS Research Report No. RR-11-16. Princeton, NJ: ETS. https://doi.org/10.1002/j.2333-8504.2011.tb02252.x

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

Embretson, S. E., & Yang, X. (2007). Automatic item generation and cognitive psychology. In C. R. Rao & S. Sinharay (Eds.) Handbook of Statistics: Psychometrics, 26, 747-768. North Holland, UK: Elsevier. https://doi.org/10.1016/S0169-7161(06)26023-1

Fellbaum, C. (1998). Wordnet: An electronic lexical database. Cambridge, MA: MIT Press. https://doi.org/10.7551/mitpress/7287.001.0001

Fu, Y., & Han, K. (2022, April). Enemy item identification for different item types. Paper presented at the annual meeting of the National Council on Measurement in Education, San Diego, CA.

Gierl, M. J., & Lai, H. (2012, April). Methods for evaluating the item model structure used in automated item generation. Paper presented at the annual meeting of the National Council on Measurement in Education, Vancouver, BC.

Gierl, M. J., & Lai, H. (2013). Instructional topics in educational measurement (ITEMS) module: Using automated processes to generate test items. *Educational Measurement: Issues and Practice*, 32(3), 36-50. https://doi.org/10.1111/emip.12018

Gierl, M. J., Zhou, J., & Alves, C. (2008). Developing a Taxonomy of Item Model Types to Promote Assessment Engineering. *The Journal of Technology, Learning and Assessment*, **7**(2). Retrieved from https://ejournals.bc.edu/index.php/jtla/article/view/1629

Glas, C. A. W., & van der Linden, W. J. (2003). Computerized Adaptive Testing with item cloning. *Applied Psychological Measurement*, 27(4), 247-261. https://doi.org/10.1177/0146621603027004001

Gomaa, W.H., & Fahmy, A. A. (2013). A survey of text similarity approaches. *International Journal of Computer Applications,* 68, 13-18. https://doi.org/10.5120/11638-7118

Haladyna, T. M. (1999). Developing and validating multiple-choice test items. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Higgins, D. (2007). Item Distiller: Text Retrieval for Computer-Assisted Test Item Creation, ETS Research Memorandum No. RM-07-05. Princeton, NJ: ETS.

Hombo, C. M., & Dresher, A. R. (2001). A simulation study of the impact of automatic item generation under NAEP-like data conditions. Paper presented at the annual meeting of the National Council of Educational Measurement, Seattle.

Hoshino, A., & Nakagawa, H. (2005). Real-time multiple choice question generation for language testing: a preliminary study. Proceedings of the Second Workshop on Building Educational Applications using Natural Language Processing, 17-20. Ann Arbor, US. https://doi.org/10.3115/1609829.1609832

IntelliMetric Engineer [computer software]. (1997). Yardley, PA: Vantage Technilogies.

Irvine, S. H., & Kyllonen, P. (Eds.) (2002). Item Generation for test development. Mahwah, NJ: Lawrence Earlbaum Associates, Inc.

Khan, S. M., Hamer, J., & Almeida, T. (2021). Generate: A NLG system for educational content creation. Proceedings of the 14th International Conference on Educational Data Mining (EDM 2021).

LaDuca, A., Staples, W. I., Templeton, B., & Holzman, G. B. (1986). Item modeling procedure for constructing content-equivalent multiple-choice questions. *Medical Education*, 20(1), 52-56. https://doi.org/10.1111/j.1365-2923.1986.tb01042.x PMid:3951382

Lai, H., & Becker, K. A. (2010, May). Detecting enemy item pairs using artificial neural networks. Poster presented at annual meeting of the 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. https://doi.org/10.1037/0033-295X.104.2.211

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduciton to latent sematic analysis. *Discourse Processes*, 25, 259-284. https://doi.org/10.1080/01638539809545028

Lai, H., & Becker, K. A. (May, 2010). Detecting enemy item pairs using artificial neural networks. Poster presented at annual meeting of the National Council on Measurement in Education, Denver, CO.

Larkey, L. S., & Croft, W. B. (2003). A text categorization approach to automated essay grading. In M. D. Shermis & J. C. Burstein (Eds.), Automated essay scoring: A cross-disciplinary perspective (pp. 55-70). Mahwah, NJ: Lawrence Erlbaum Associates, Inc

Le An Ha & Yaneva, V. (2019, September). Automatic Question Answering for Medical MCQs: Can It Go Further than Information Retrieval? Proceedings of the 12th Conference on Recent Advances in Natural Language Processing (RANLP 2019), Varna, Bulgaria.

Le An Ha, Yaneva, V., Baldwin, P. and Mee, J. (2019). Predicting the Difficulty of Multiple Choice Questions in a High-stakes Medical Exam. Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications (BEA), held in conjunction with ACL 2019, Florence, Italy, 2 August, 2019. https://doi.org/10.18653/v1/W19-4402

Li, X., Hu, A., & Wilmurth, G. (2022, April). Enemy item detection using word embedding. Paper presented at the annual meeting of the National Council on Measurement in Education, San Diego, CA.

Li, F., & Shen, L. (2011, February). Detecting duplicate items by semantic similarity measures. Paper presented at the annual meeting of the Association of Test Publishers, Phoenix, AZ.

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

Lin, C-Y & Hovy. E. H. (2003). Automatic evaluation of summaries using N-gram co-occurrence statistics. Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003), pp. 71-78. https://doi.org/10.3115/1073445.1073465

Luecht, R. M. (1998). Computer-assisted test assembly using optimization heuristics. *Applied Psychological Measurement*, 22, 224-236. https://doi.org/10.1177/01466216980223003

Minaee, S. & Liu, Z. (2017). Automatic question-answering using a deep similarity neural network. 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2017, pp. 923-927. https://doi.org/10.1109/GlobalSIP.2017.8309095

Manning, C. D., & Schütz, H. (1999). Foundations of statistical natural language processing. The Cambridge, MA: The MIT Press.

Mitkov, R., Ha, A. A., & Karamanis, N. (2005). A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(2), 177-194. https://doi.org/10.1017/S1351324906004177

National Library of Medicine (2009). UMLS Reference Manual. Bethesda, MD: Author.

National Library of Medicine (2021). UMLS(R) Reference Manual. National Institutes of Health. https://www.ncbi.nlm.nih.gov/books/NBK9676/pdf/Bookshelf_NBK9676.pdf

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

Page, E. B. (1994). Computer Grading of Student Prose, Using Modern Concepts and Software. *Journal of Experimental Education*, 62, 127-142. https://doi.org/10.1080/00220973.1994.9943835

Page, E. B. (2003). Project essay grade: PEG. In M. Shermis & J. Burstein (Eds.), Automated essay scoring: A cross-disciplinary perspective. Hillsdale, NJ: Lawrence Erlbaum Associates.

Peng, F. (2020). Automatic enemy item detection using natural language processing. (Unpublished doctoral dissertation). The University of Illinois at Chicago, Chicago, IL.

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., & Seagall, D. O. (2006, April). Local dependence in an operational CAT: Diagnosis and implications. Paper presented at annual meeting of the National Council on Measurement in Education, San Francisco, CA.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130-137. https://doi.org/10.1108/eb046814

Porter Stemming Algorithm (n.d.). Retrieved March 6, 2005, from http://www.tartarus .org ~martin/PorterStemmer

Roid, G. H., & Haladyna, T. M. (1982). A technology for test-item writing. New York: Academic Press.

Rudner, L. M., Garcia, V., & Welch, C. (2006). An evaluation of the Intellimetric essay scoring system. *Journal of Technology, Learning and Assessment*, 4(4). Available from http://escholarship.bc.edu/jtla/

Shin, J., Guo, Q., & Gierl, M. J. (2019). Multiple-choice item distractor development using topic modeling approaches. *Frontiers in Psychology*, 10, 825. https://doi.org/10.3389/fpsyg.2019.00825 PMid:31133911 PMCid:PMC6524712

Steyvers, M., & Griffiths, T. (2004). Probabilistic topics models. In T. Landauer, D. McNamara, S. Dennis, & W. Kintsch (Eds.), *Handbook of latent semantic analysis* (pp. 427-448). Lawrence Erlbaum Associates, Mahway, New Jersey.

Sukkarieh, J. A., & Stoyanchev, S. (2009). Automating Model Building in c-rater. Paper in Proceedings of Text Infer: The ACL/IJCNLP 2009 Workshop on Applied Textual Inference, pp. 61-69. https://doi.org/10.3115/1708141.1708153

Swanson, L., & Stocking, M. L. (1993). A method and heuristic for solving very large item selection problems. *Applied Psychological Measurement*, 17, 151-166. https://doi.org/10.1177/014662169301700205

van der Linden, W. J. (1998). Optimal assembly of psychological and educational tests. *Applied Psychological Measurement*, 22, 195-211. https://doi.org/10.1177/01466216980223001

Veldkamp, B., & van der Linden, W. (2000). Designing Item Pools for Computerized Adaptive Testing in W. van der Linden and C. Glas, (Eds.) *Computerized Adaptive Testing: Theory and Practice*. Netherlands: Springer. pp. 149-162. https://doi.org/10.1007/0-306-47531-6_8

Wang, J. & Dong, Y. (2020). Measurement of Text Similarity: A Survey. *Information*, 11*(9)*, 421. https://doi.org/10.3390/info11090421

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

Widdows, D., & Ferraro, K. (2008). Semantic vectors: A scalable open source package and online technology management application. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odjik, S. Piperidis, & D. Tapias (Eds.) *Proceedings of the Sixth International Conference on Language Resources and Evaluation*. pp. 1183-1190.

Xi, X., Higgins, D., Zechner, K., & Williamson, D. (2008). Automated Scoring of Spontaneous Speech Using SpeechRater v1.0. ETS Research Report No. RR-08-62. https://doi.org/10.1002/j.2333-8504.2008.tb02148.x

Yen, W. (1993). Scaling Performance Assessments: Strategies for Managing Local Item Dependence. *Journal of Educational Measurement*, 30(3), 187-213. https://doi.org/10.1111/j.1745-3984.1993.tb00423.x