



## **Finding Stolen Items and Improving Item Banks**

Kirk A. Becker, PhD

Shu-Chuan Kao, PhD

*Pearson VUE*

Paper presented at the 2009 annual meeting of the American Educational Research Association, San Diego, CA. Correspondence concerning this article should be addressed to Kirk Becker, Pearson, 1 N Dearborn, Suite 1050, Chicago, Illinois, 60602, or via e-mail: [kirk.becker@Pearson.com](mailto:kirk.becker@Pearson.com)

## Finding Stolen Items and Improving Item Banks

Kirk A. Becker, PhD

Shu-Chuan Kao, PhD

Pearson VUE

*Natural Language Processing (NLP) offers several methods for quantifying the similarity between written documents (Bates, 1995). Within the testing industry, these methods have been used for automatic Item generation (Deane & Sheehan, 2003) and automated essay scoring (Shermis & Burstein, 2003), and they have more recently been used to estimate item statistics (Hall, 2008; Belov & Knezevich, 2008). This paper presents initial research into the use of these methods for identification of stolen items, classification and referencing, and the identification of enemy and duplicate items. The results of these initial analyses show significant evidence for the utility of NLP methods for item bank management. While this paper presents a particular application of the cosine similarity index, more recent similarity indexes such as ROUGE (Lin and Hovy, 2003), or more advanced processing of item content may prove even more useful for these purposes.*

### Background

The field of Natural Language Processing (NLP) within computer science has developed methods for indexing, categorizing, summarizing, and interpreting large numbers of text documents. The testing industry has made use of these methods in the development of automated essay scoring engines (Shermis & Berstein, 2003). Recent research on item parameter estimation has also made use of NLP methods (Belov & Knezevich, 2008; Hall, 2008). However, outside of these areas, the testing industry has remained ignorant of the capabilities of NLP relative to our biggest assets—our item banks.

The testing industry works predominantly with large collections of text, and we have technology for organizing test items (content management tools). While we have moved away from shoe boxes for organizing test items, all the work done to review and evaluate item banks is performed by humans reading those items. This paper presents a proof-of-concept for the application of NLP techniques to several areas of test development and item bank management.

### Theoretical Framework

Beginning in the 1950s, a new discipline arose in computer science devoted to make computers understand natural language, a language spoken or written by humans for general-purpose communication. The goal of Natural Language Processing is to convert samples of human language into more formal representations that are easier for computer 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 understand” of the sentence, considered more feasible and useful. (For more detail on the history of NLP, see Bates, 1995.)

This research makes use of one method from NLP—the cosine similarity index—a measure of the similarity between two selections of text. The selections of text could be essays, paragraphs from textbooks, websites, or any other written material. This research analyzed individual items from publicly available as well as proprietary item banks. Details of how item text is processed and analyzed appear in the Methods section.

The analyses in this paper represent a proof-of-concept for the application of one method from NLP to several tasks related to item bank management. After investigating the applications of a similarity index for identifying stolen items, identifying duplicate items, and identifying or verifying item classifications, it also proposes future directions for this research.

Since items on continuous exams are typically stolen through memorization, stolen content would not be expected to match exactly the content in an item bank. Because the cosine similarity is robust to small differences in content, it can evaluate near-matches between potentially stolen test items and actual test

questions. Additionally, because the item bank is represented abstractly, as a matrix of numbers, it is potentially more secure than methods that require access to the full content of the test items.

Large item banks present logistical problems to the item development process that are difficult to solve. As item banks grow, test development processes become exponentially less efficient. The ability of item developers to identify duplicate content and answer cuing is especially problematic as the number of items in a content area increases above 50-100 items. The similarity index provides a method for evaluating whether newly written items are substantially the same as existing items (and therefore do not warrant pretesting). Additionally, the similarity index provides a possible method for flagging item pairs that should be considered enemies (either for content overlap, or for cuing).

Item classifications are typically assigned by item writers as they generate new test items. While it may be unusual to receive items without classifications, a measure of similarity between new items and existing items with known classifications serves as an additional check. When content classifications change (e.g., due to a job analysis) or new classifications are added (e.g., for cognitive complexity), there may be a need to reclassify large numbers of items. The relationship between item classifications and cosine similarity will be evaluated.

### **Methods**

Prior to calculating the cosine similarity, test items must be processed to create a matrix. Each row in the matrix represents a test item, while each column represents a word. The values for each cell of the matrix represent the count of the word within the item. Methods such as this that treat each text response as a collection of disassociated word variables are known as “bag of words” methods (Steyvers & Griffiths, 2004). Table 1 represents an entry for a test item:

Typical water pressure at a boiler at rest in a two-story house is ; 1-5 psi; 5-10 psi; 10-15 psi; 15-20 psi; 20-25 psi

Table 1. Example of Text Parsing

item id	a	At	boiler	house	hydronic	In	is	normal	operating	Pressure	psi	Rest	system	the	two-story	Typical	water
item 1	2	2	1	1	0	1	1	0	0	1	5	1	0	0	1	1	1

To cut down on unnecessary variance between processed items, stopwords were removed from test items, and all remaining words were stemmed. Stopwords are common articles, pronouns, adjectives, adverbs, and prepositions (e.g., “the,” “a,” “and,” etc.). Stemming is when words are converted to their “stems,” reducing different tenses and forms of a word to a common root. For example, a stemming program would convert the words “respect,” “respecting,” “respects,” and “respectful” to the same form.

For each item, a matrix was created for both the stem only, as well as the stem and the response options.

First, punctuation, numbers, and case were removed from test items. Next, words were removed based on a list of 180 stopwords. Finally, items were stemmed using the Porter stemmer (Porter, 1980).

The Porter stemmer provides a well-documented method for automatically removing suffixes from words. Porter’s algorithm makes use of an explicit list of suffixes and applies criteria to determine when they can be removed from a word in order 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., “wand” and “wander” should be separate words, but this

only matters if the data contain instances of both “wand” and “wander”). The Visual Basic implementation of the Porter stemmer was written by Navonil Mustafee while at Brunel University, and the source code is freely distributed (Porter Stemming Algorithm, n.d.).

All of stemmed words from the item bank (with stop words removed) form the dimensions of the item bank’s “semantic space.” The primary idea underpinning the semantic vector is that points can represent words and concepts in a mathematical multidimensional space. In this high dimensional vector space, the presentation is learned from the text in a way that concepts with similar or related meanings are near one another (Widdows & Ferraro, 2008). When applying the idea of semantic vector/space in testing, each item can then be represented as an N-dimensional vector within this space. Consequently, the comparison of items’ content can be achieved by evaluating items’ N-dimensional vectors.

Each dimension of the vector space corresponds with 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 zero. Vectors that contain the same words or content should be roughly parallel, while vectors that relate to different content should be oblique. The concept of angular distance, the size of the angle between the two directions originating from the origin and pointing towards two objects, is employed to signify the distance/similarity of two semantic vectors. The degree to which two vectors are parallel can be quantified through the cosine of the angle between the vectors. 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 distance/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 research precludes negative cosines, so the range of the cosine similarity index is 0 to 1. A

weighting scheme involving negative weights would produce vectors that could have negative cosines.

## Data

Multiple-choice items posted to a national message board under the claim of exposed content were used both to demonstrate the relationship between content and cosine similarity, as well as to evaluate the utility of the cosine similarity for finding stolen content.

Several live item banks used in the regulatory and certification industries were used to investigate the cosine similarity index for enemy item detection and item classifications.

## Results

### *Stolen Items*

Claiming that a national exam was compromised, a participant on a message board posted content from the supposedly exposed item pool. A total of 233 items were initially posted. The item bank in question contained approximately 1500 test items.

A content developer reviewed all 233 items relative to the format of the item bank and found that many of the items did not match the structure of the items in the item bank (e.g., number of distractors, true/false items, and fill in the blank). Additionally, keywords were manually extracted from each item and used to query the item bank. Finally, the cosine similarity was calculated between each of the 233 items and each of the 1500 items from the item bank. This analysis flagged 11 items based on a total item similarity (stem + options) greater than 0.8. Review of these items found them to deal with similar content but established that they were obviously not copies of the live test items.

### *Enemy Items*

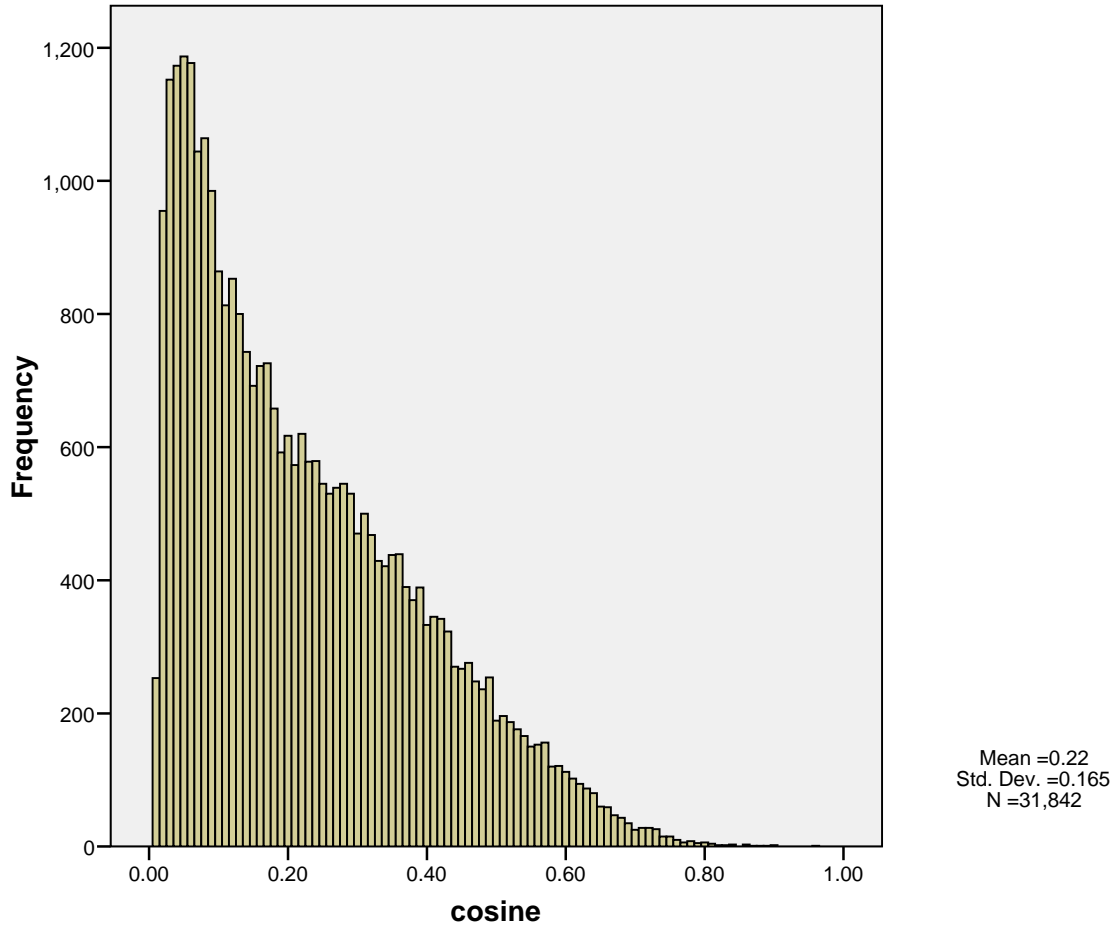
A total of 257 items with enemy relationships were selected from an item bank. Of these 257 items, 156 had an enemy pair within the selected items, resulting in 92 enemy pairs out of approximately 33,000 item pairs.

The frequency of different ranges of cosine similarity values for item pairs is presented in Table 2, along with the number of enemy pairs within that range. Only values greater than 0 are included. Figure 1 provides the histogram for these results. All but 2 enemy items are above the mean (0.22) of the cosine similarity.

Table 2. Frequency of Similarity Indexes and Identified Enemy Pairs

Index Range	Frequency	Percent	Cumulative Percent	Enemies
.01-.09	8990	28.2	28.2	2 enemy pairs
.1-.19	7463	23.4	51.7	1 enemy pairs
.2-.29	5656	17.8	69.4	9 enemy pairs
.3-.39	4314	13.5	83.0	14 enemy pairs
.4-.49	2894	9.1	92.1	16 enemy pairs
.5-.59	1614	5.1	97.1	13 enemy pairs
.6-.69	719	2.3	99.4	22 enemy pairs
.7-.79	166	0.5	99.9	9 enemy pairs
.8-.89	23	0.1	100.0	6 enemy pairs
.9-.99	3	0.0	100.0	
Total	31842	100		92 enemy pairs

Figure 1. Distribution of Similarity Index



*Duplicate Items*

Several of the 233 items referenced in relation to stolen items were slight variations of each other, and because that content was publicly posted, those items were used for demonstration purposes. This section is more anecdotal, showing the content of items with high and low cosine similarities. Future research will investigate the identification of item copies or item variants that are

already known, and it will describe comparisons in time and accuracy between human and computer reviews of 50-100 item banks. The total and stem similarities, as well as content of three items from this set of items, are listed below.

**Similarity:** 0.98 (stem similarity 1)

**Item 1:** The three major components of a fossil fuel forced air furnace are ; heat exchanger burner blower; return air combustion air and vent system; plenum blower register; heating ducts hot air vent and plenum box

**Item 2:** Three major components of a fossil fueled forced air furnace are ; plenum blower register; return air combustion air vent system; heat exchanger burner blower; heating ducts hot air venting system plenum

**Similarity:** 0.9 (stem similarity .73)

**Item 1:** In a hydronic system the normal operating pressure is ; 30-40 psi; 12-25 psi; 3-5 psi; 22-25 psi

**Item 2:** In a steam heating system a normal operating pressure is ;30 psi; 12-20 psi; 3-5 psi; 22-25 psi

**Similarity:** 0.81 (Stem similarity .17)

**Item 1:** Typical water pressure at a boiler at rest in a two-story house is ; 1-5 psi; 5-10 psi; 10-15 psi; 15-20 psi; 20-25 psi

**Item 2:** In a hydronic system the normal operating pressure is ; 30-40 psi; 12-25 psi; 3-5 psi; 22-25 psi;

### *Item Classifications*

Cosine similarities were computed for all pairs of items in a national item bank with four major content areas. The average cosine similarity between each item and the items from the four content areas was determined. A predicted classification was calculated based on the highest of the four average cosines. For example, if item 1 had average cosines of .12, .06, .04, and .08 with content

areas 1, 2, 3, and 4, it would be classified into content area 1. These predicted classifications were then compared with the item classifications in the item bank, and 503 out of 676 (74.4%) matched. These results are reported in Table 3.

Table 3. Predicted and True Classifications for National Item Bank.

Predicted Classifications	"True" Classifications				Grand Total
	01	02	03	04	
01	195	38	3	33	269
02	29	190	6	29	254
03	1	25	77	9	112
04	0	0	0	41	41
Grand Total	225	253	86	112	676

### Discussion/Future Directions

Automating routine and repetitive methods leads to higher accuracy and more time for creative work. One of the main purposes of this paper is to expose the testing industry to the potential offered by NLP for evaluating item banks. Methods such as cosine similarity are widely available, and they could be implemented without the need for a high level of sophistication. These methods show promising results when applied to several areas of item bank analysis, and we would encourage other programs to replicate our results.

The proposed procedure can be applied for searching possibly compromised items to enhance item pool security. Comparing the items in the operational pools against those on the shared Internet websites or in training materials from a coaching school can identify compromised items, and further investigation can proceed accordingly. Using a comparison method based on semantic vectors provides additional security, in that actual item content is not required for the comparisons (although it is required for the initial processing). With the proposed

procedure, identifying stolen items can be less labor-intensive and time-consuming. More importantly, the objective evidence, the statistics of content similarity between item pairs, can be provided if legal action should take place.

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 definitely support the use of these methods for identifying duplicate content. It would make sense for testing programs to evaluate their newly written items relative to their banks to determine if they are developing unique content or rehashing existing test items.

Results from the evaluation of enemy items are promising (enemy pairs have higher cosine similarity than items in general), but they are not high enough to significantly limit the content a person would need to review. Review of characteristics of enemy items, and of characteristics of non-enemy items with high cosine similarities, will determine how to better identify potential enemies.

While the cosine similarity index has shown promising properties, NLP has produced other measures that are coming into wider use. Additionally, processes for representing meaning rather than specific words (e.g., WordNet or LSA) might better identify items with similar meaning, and they might not flag items with similar structure. Further research is needed in this area.

Initial results using the cosine similarity to classify items look good. Future research will involve classifying items into subcategories as well as major categories, and reviewing content for items that appear more similar to content areas to which they are not assigned.

### References

- 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.
- Belov, D. I., & Knezevich, L. (2008). *Predicting item difficulty with semantic similarity measures*. Paper presented at annual meeting of the National Council on Measurement in Education, New York, NY.
- Deane, P. & Sheehan, K. (2003). *Automatic item generation via frame semantics: Natural language generation of math word problems*. Princeton, NJ: ETS.
- Hall, E. (2008). *Exploring the use of item bank information to improve IRT item parameter estimation*. Paper presented at annual meeting of the National Council on Measurement in Education, New York, NY.
- Lin, Chin-Yew and E.H. Hovy (2003). Automatic evaluation of summaries using N-gram co-occurrence statistics. In *Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May 27 - June 1, 2003.
- Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130-137.
- Porter Stemming Algorithm (n.d.). Retrieved March 6, 2005, from <http://www.tartarus.org/~martin/PorterStemmer>
- Shermis, M. D., & Burstein, J. C. (2003). *Automated essay scoring: A cross-disciplinary perspective*. Mahwah, NJ: Lawrence Erlbaum.

Steyvers, M., & Griffiths, T. (2004). Probabilistic topics models. Manuscript submitted for publication.

Widdows, D. & Ferraro, K. (2008). Semantic vectors: A scalable open source package and online technology management application. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation, Vol.(issue)*, pages.