Journal of Applied Testing Technology, Vol 23(Special Issue 1), 30-40, 2022

Leveraging Machine Learning Technology to Improve Accuracy and Efficiency of Identification of Enemy Item Pairs

Ian Micir[1]*, Kimberly Swygert[1] and Jean D'Angelo[2]

[1]National Board of Medical Examiners, Philadelphia, PA 19104, USA; imicir@nbme.org, kswygert@nbme.org
[2]American Osteopathic Association, Chicago, IL 60611-2864, USA; jdangelo@osteopathic.org

Abstract

The interpretations of test scores in secure, high-stakes environments are dependent on several assumptions, one of which is that examinee responses to items are independent and no enemy items are included on the same forms. This paper documents the development and implementation of a C#-based application that uses Natural Language Processing (NLP) and Machine Learning (ML) techniques to produce prioritized predictions of item enemy statuses within a large item bank, which can then be followed by medical editor review of the prioritized predictions as part of an iterative process. An item bank of 4130 items from a large-scale healthcare specialist certification exam was used, in which it was assumed that many unidentified enemy pairs existed. For each pair of items, cosine similarities using TF-IDF weights were computed for the stem and answer text separately, with additional dichotomous classification variables added indicating content and existing enemy relationships. Each item pairs' existing enemy status (enemy or non-enemy) was the dependent variable for the supervised ML model, the coefficients of which were then used to generate probabilities that a given pair of items were enemies. Medical editors reviewed prioritized lists of the actual versus predicted enemy relationships in an iterative fashion. Of the 700 untagged enemy item pairs reviewed, 666 were confirmed and tagged by editors as enemies (95.1% accuracy). Thus, this application was successful in allowing editors to efficiently identify the most egregious uncoded enemy item pairs in a large item bank. The ultimate goal of this research is to inform discussion about the potential for NLP and ML applications to greatly improve accuracy and efficiency of human expert work in test construction.

Keywords: Item Banks, Item Enemies, Machine Learning, Natural Language Processing, TF-IDF Function

1. Introduction

The goal of any assessment is to measure the degree to which an examinee understands the requisite knowledge, skills or abilities in a given domain or set of domains. Within that context, each item on an exam form is an opportunity for the test developer to obtain a small amount of information from a test-taker, which, when summed across many items, contributes to an increasingly accurate and reliable measurement of overall ability level. On a standard Multiple-Choice-Question (MCQ) based assessment, the capability to sum these opportunities to a total score that is highly reliable and supported by validity evidence is dependent upon the items being independent of factors unrelated to the construct being measured and uncorrupted from various sources of construct-irrelevant variance. One common source of this type of corruption is item pre-knowledge, which is often equated to cheating in the test security literature. For this paper, however, the focus is on a specific type of non-security-related item pre-knowledge, known as enemy item relationships, that can exist within a test form as a potential and problematic source of construct-irrelevant variance.

*Author for correspondence

Enemy items are any pair of items that should not be placed on the same test form because of the potential to inflate a test-taker's score and create adverse effects for reliability, validity evidence and/or content sampling. Item pairs can be deemed enemies for a variety of reasons; the most common ones are a high degree of item text similarity between the two items; a similar assessment point, such that knowing the answer to one would assure the examinee would know the answer to another; or cluing (aka cuing), where one item actually contains information about the correct answer to the other item (Woo and Gorham, 2019; Lane, Raymond and Haladyna, 2016; NBME, 2021; Gierl, Lai, and Tanygin, 2021). Best practices for maintaining and reviewing item banks and ensuring accurate item coding or metadata should include a systematic review by content experts to ensure all enemy item pairs are identified. Following this step, enemies within an item bank can be "tagged" as such in the metadata, and any test construction algorithm used to create the forms can place a constraint on this metadata to avoid placing a pair of enemies on a test form. The complication arises, however, with the question of how to complete a comprehensive and systematic review to identify all enemies in an existing bank, especially large item banks that are continuously updated.

In a bank of 100 items, there are 4950 possible unique pairs of items, which might seem manageable for a manual review. Editors or subject matter experts can easily manually flag enemy items in a bank of this size by searching for other items in the same content areas or with similar metadata. However, in the world of larger-scale assessments, a bank of 100 items is unrealistic and as bank size increases, the number of possible unique pairs increases quadratically. In a 1000-item bank, which might be standard for banks used to produce multiple logistic short testing forms in educational scenarios, there are 499,500 pairs. In a 10,000-item bank – the minimum size required for any type of adaptive or on-the-fly testing or for developing lengthy linear forms with multiple content constraints that are used in high-stakes, high-security scenarios – the number swells to 49,995,000 possible pairings. In addition, while untagged item enemies might be identified via form review after logistic fixed forms are constructed, the lack of such review for any type of dynamic or adaptive test administration requires that the flagging of item enemies – and non-enemies – must be flawless for the entire bank prior to any exam administration. Thus, it very quickly becomes apparent that the crucial challenge with enemies is not the management of the known, but the identification of the unknown.

2. The Rise of the Machines

As with many challenging big data scenarios, the automated enemy identification scenario for large item banks seems ripe for a solution based on elements of Artificial Intelligence (AI). The term AI predates the widespread adoption of other, more familiar forms of computer technology such as the internet, thanks to the seminal article by John McCarthy (1956) that defined it as "the science and engineering of making intelligent machines." The evolution of AI has not run smoothly since its beginning, with several "AI Winters" occurring due to lack of interest, lack of funding or insufficient computing power and data mining techniques to support research (Taulli, 2019). The current age, however, is seeing a boom in both the recognition and widespread use of the data mining, modeling and statistical analysis components that comprise AI.

The conception of a general AI, where machines can be programmed to learn and execute extremely complex tasks by thinking very much like humans, has leapt ahead in the public's imagination, helped by visionary forms of entertainment such as the films A.I. and Ex Machina (Semmler and Rose, 2017). However, the AI methodology that would be most useful for managing item banks is the less glamorous, but far more available, narrow AI, where the AI element is focused on a single, specific task (Kaplan and Haenlein, 2019). Test developers who choose this option should be guided by the same principles related to AI transparency, defensibility and accountability as any technology company that includes AI elements in their products (ATP, 2021). In addition, all test development tasks, even the parts of processes invisible to the test-taker, should contribute to the validity evidence for the use of the test scores as delineated in the Standards for Educational and Psychological Testing (AERA, 2018; ATP, 2021). These requirements are most easily supported with the use of a subtype of narrow AI that still relies heavily on "humans in the loop," where the use of AI methodology is reserved only for tasks where the computer can do something faster or more accurately than humans, and human expertise is retained

for oversight, judgments and decision-making steps that still require human intelligence. This subtype can be observed with technologies such as automated essay scoring in high-stakes assessment, where human experts, though known to be imperfect, are still used as a check on extreme automated scores (Wang and von Davier, 2014). Likewise, any AI process used to identify enemy items in a large item bank would benefit, for validity reasons, from using human experts to render the final decisions.

As AI is a collection of modeling and statistical analysis techniques, the next topic to consider is the specific subset of narrow AI methodologies that would be most useful in the context of enemy item identification. Given that most MCQs are heavily text-based, an algorithm related to Natural Language Processing (NLP) will obviously be useful. NLP is a subfield of AI in which computers process and "understand" human language (Jurafsky and Martin, 2009). Challenges in training computers to process and make judgments related to human language have been substantial, given that language can be ambiguous, change frequently, contain errors, be vague and have other inconsistencies that humans learn to decipher more readily than do computers (Taulli, 2019). However, progress in this field has moved from the Turing Test (1950), in which a computer's use of language was linked to its perceived ability to think, to the world of today, where humans routinely communicate with Alexa, Siri, chatbots and other computer programs that use narrow AI to decipher and respond to written or spoken human language (Cho *et al.*, 2019; Punjabi, Arsikere and Garimella, 2019).

For evaluating the language of test items, however, the situation is greatly simplified. Problematic scenarios such as inconsistent terminology, vague terms, grammatical errors and other issues common in regular speech are systematically removed from test items for high-stakes, standardized assessments (Paniagua, Swygert and Downing, 2019; NBME, 2021). Thus, test items are ideal testing grounds for NLP methods that can quickly parse standardized text, where straightforward comparisons of text similarity are easily computed and interpreted. Investigations into the potential for NLP applications for general management of large item banks have emerged in the last 15 years (Becker and Kao, 2009, 2013; Kao and Becker, 2010), with a handful of studies related specifically to the use of NLP methods for automated item enemy identification being published within the last two years (Peng, 2020; Gierl, Lai and Tanygin, 2021; Mao, Zhang and Clem, 2021).

This paper represents a substantial contribution to this brief but rapidly growing field, moving beyond the theoretical support for the use of NLP in enemy item pair management to the operational and verified use by humans in a pilot study applied to an existing item bank for a large-scale standardized assessment. This pilot study was conducted to demonstrate the utility of an application prototype that pairs NLP methodology with the Machine Learning (ML) algorithm of logistic regression to rapidly generate probabilities that item pairs are enemies, between all possible item pairs, in a large item bank. This prototype, informally known as Smokey, generates and prioritizes predicted enemy probabilities to allow medical editors to review outcomes alongside the actual item text and make the final judgments on whether enemy item pair flags should be added to or removed from the live item bank. This study also touches on the importance of developing AI applications that are intuitive and highly efficient, so that new software, using algorithms that might seem technologically daunting at first, can be introduced into well-established test development processes with minimal disruption and manageable training and change management procedures.

3. Methods

3.1 Data

An item bank of 4130 items (representing 8,526,385 possible pairs) from a large-scale healthcare specialist certification exam was used for this study. All items were single-best-answer multiple-choice questions with short vignettes, written and edited to specific standards in terms of phrasing and sentence structure (NBME, 2021). The items were developed to satisfy a detailed test construction blueprint, where the constructs of interest represented by the blueprint's content areas involved specialized knowledge in a healthcare field that contains both patient-facing and administrative tasks. Item structure ranged from short presentations of facts, with a lead-in asking the test-taker to recall definitions, to longer vignettes that required the test-takers to synthesize information, some of which included patient scenarios.

This item bank was chosen not only because the items were standardized and relatively short, but also because it was anticipated, due to previous methods used for bank assembly and review, that there were many enemy item pairs within the bank that were not tagged as enemies, despite potential substantial text overlap. All item pairs that were tagged as enemies had been reviewed and tagged by a human reviewer, but all remaining item pairs not tagged as enemies (the vast majority of the bank) might have never been reviewed or might have been reviewed and determined not to be enemies. Not only were there many item pairs not tagged as enemies, but an unknown number of those had never been evaluated as to whether they were enemies. The initial dataset included the full text of the item stem (including facts/vignette and lead-in) and full option set, including the key. Prior to implementing any NLP methodology, however, the text required preprocessing.

3.2 Text Preprocessing

Text preprocessing is an essential step in most NLP work. It involves the act of converting the much nuanced system of human grammar into a more standardized, computer-friendly version of itself. The steps include lowercasing, punctuation removal, stop-word removal (articles and other non-impact words like "if," "and," "to," etc.) and stemming. Lowercasing and punctuation removal are fairly straightforward, but the list of stop-words can vary greatly based on the overall contents of the text in a given analysis. The items for this pilot study were written in extremely standardized fashion, typically featuring the phrase, "Which of the following," in the lead-in. Thus, the word "following," which could be a meaningful word in other contexts, was added to the list of stop-words to be removed. This is a small but useful example of how even a preliminary step like text preprocessing is not one-size-fits-all; even the most routine elements of NLP may require iterations of revisions to better fit the data available or task at hand.

Stemming and lemmatization are similar steps, with stemming being the simplistic, less precise approach. In both processes, the end goal is to extract the root word of a given term, such that injects, injected, injection, injecting, etc. all register as the same root word, allowing the computer to recognize that they are the same. Stemming is a fairly simplistic approach that removes letters from a term until it reaches a root term. Lemmatization incorporates the broader context of a language to provide a simplified version of an actual term. Lemmatization would turn the "inject-" words above into "inject," whereas, stemming would result in "injec." Lemmatization produces a more intelligible output at the expense of speed, as it is more computationally taxing. If the data for this pilot study were dealing with social media text, or comparing the prose in novels written by different authors, lemmatization would be optimal and would most likely provide significantly better results. However, given the highly standardized terminology and sentence structure of the pilot test items, speed was the primary factor, and the Porter stemmer method (Porter, 1980) was used. Once items were transformed during preprocessing, further work could commence.

3.3 Item Text as Vectors in Space

Item text can be considered part of a semantic space in which words are vectors and the similarity of an item as an N-dimensional vector with another item can be represented mathematically via vector comparison (Jurafsky and Martin, 2009). To compare two items represented as vectors, the cosine can be used as a measure of distance between them, where a cosine value = 1 would indicate two items that are identical across all words retained after preprocessing, and a cosine value = 0 would indicate two items with no words in common. This cosine value is known as the cosine similarity index and is one of the best-established measures for evaluating semantic similarity (Kusner *et al.*, 2015; Sitikhu *et al.,* 2020). However, not all words are created equal for the purpose of judging semantic space similarity, and this is where a weighting measure such as the tf-idf (term frequency-inverse document frequency) weight can be useful.

The first step in estimating the tf-idf weight is to establish a vocabulary of all unique terms (post-preprocessing). The number of times a term is used within a document (in this case, an item) is the tf, and the idf represents a weight that assigns higher value to rare terms in the total vocabulary (in this case, all unique words from all items), based on the number of documents that contain one or more uses of the term. The use of the tf-idf weighting is crucial in assessment scenarios because, within the framework of highly standardized test items, there are often terms that are used across many items in the bank that are not removed during preprocessing due

to their content-relevant meaning. For example, if a test contains items that include patient vignettes, "man" might appear in roughly half of these items. While this term may be very relevant for a given item, it would not be of high value as an indicator of similarity between items in that bank. Conversely, highly specific medical terms that refer to diagnoses and symptoms are more likely to be rare, even on an exam focused on such constructs, and should be weighted more highly as an indicator of similarity. Using the item bank as the corpus, a tf-idf weighted cosine similarity can be calculated for any pair of items, with values ranging from 1 to 0; the higher the value of the cosine similarity, the more similar the items are, with rare terms weighted more heavily in quantifying that similarity than common terms.

3.4 Application Development for NLP and ML Elements

Smokey was designed in C# with a streamlined graphic user interface that allowed medical editors to select the specific item bank of 4130 items for this study, along with item information from that bank as follows: Stem text, correct option text and selected content codes. While the set of specific content codes and their descriptions that were used for this pilot study cannot be published here, they can be considered representative of the data that would be commonly used for coding medical factual or vignette-based items.

After the dataset was assembled, the preprocessing routine was conducted. Once the final set of text segments was available for use, Smokey was used to run two sets of analyses: 1. A set of NLP estimations (the computation of the tf-idf weighted cosine similarity between each pair of items) and 2. An ML analysis using the similarity indices and item content coding to predict a final similarity indication. These are described below.

NLP Calculations

The Smokey processing began with iteration through each unique pair of items where, for each pair, an array was generated for each item of length *n*, where *n* was the number of unique terms used across either of the items remaining after preprocessing. Each position in the array was then filled in with the tf-idf value of the corresponding term relative to the inverse of that item's term frequencies. For the purposes of this study, two tf-idf cosine similarities were generated between each pair of items: one for the item stem text (CS_IS), and one for the correct option text (CS_OT).

In the next step of processing, Smokey generated a dichotomous variable that represented agreement of the primary content codes between each pair of items, referred to here as variable CC1. For example, if a pair of items had been assigned the same content code for the primary content category, CC1 was assigned a 1 for that specific pair of items; if the codes were different, CC1 was assigned a 0. Finally, Smokey used the existing item enemy flags in the bank to generate a dichotomous enemy variable, EV, for each pair of items. If two items were already coded as enemies of one another in the bank, EV was set to 1; if there was no such code in the bank for this pair, EV was set to 0. As noted above, the vast majority of the item pairs in the bank had an EV = 0, but it was anticipated that many of these were genuine enemy pairs that had not been tagged during earlier manual reviews.

ML Estimations

The model chosen for this step was the iterative reweighted least squares logistic regression within the Accord Framework (an open-source .NET library for Machine Learning). A logistic regression was chosen, as the model was to be trained on data where the outcome variable was binary. The independent variables for each item pair were CS_IS, CS_OT and the dichotomous content code agreement value, CC1. The dependent value was EV, the dichotomous enemy indicator as reflected by the existing bank coding.

In ML scenarios with regression models, the dataset is often split into training and test datasets. The split is typically 80%-20% training to test but can be 50%-50%. However, in this scenario, the fact that it was expected that many of the item pairs were incorrectly tagged as non-enemies was a concern. It was not a given that any random subset of the bank would produce enough examples of accurately-tagged enemies and non-enemies to be useful as a training set. For this and other practical reasons, the decision was made to use the entire dataset for the training of the logistic model, to train Smokey on the level of semantic similarity to be found with enemy item pairs, and then to repeat the same regression over multiple runs, with the enemy item tags in the bank updated between each run with the enemy item pairs verified by editors during the previous iteration(s). This method was sufficient for the pilot at hand because the

pilot was intended to test whether Smokey could generate output that was easily interpretable to editors and whether editors could in turn efficiently make decisions to update the bank between iterations of regressions. In addition, the goal was to generate regression coefficients that were fit exactly to this one item bank at this point in time, with no assumptions made that the regression coefficients would be generalizable to any other item bank. The initial version of Smokey did not include a method to record the coefficient values for each variable. Smokey works with live (as opposed to static) data pulled directly from databases at runtime; because the data have since changed considerably, we are unable to retrieve the point-in-time coefficients. This functionality has since been added to Smokey and will be made available should we publish follow-up studies using more clinically complex vignettes and diverse metadata.

Thus, to start, the entire bank was used to estimate coefficients for the CS_IS, CS_OT and CC1 values, where those were generated using the existing EV values as the dependent variable. That dependent variable EV was then removed from the equation and the coefficients were applied to the same three predictor variables for the entire bank to generate, for each item pair, a Smokey value for an estimated EV outcome, representing the probability on a 0 to 1 scale that the two items were enemies. As the ultimate goal was to support editors in making binary enemy/non-enemy decisions, the decision was made to choose a low baseline threshold to classify an item pair as enemies. All pairs with probabilities > 0.05 were classified as predicted enemies, while those below the threshold were predicted to be non-enemies (empirical results from earlier test runs of the application suggested that this threshold would be sufficient for identifying a high proportion of the untagged enemies). The results for each pair of items could be classified into one of the four categories shown in Table 1. Because the 0.05 threshold was somewhat arbitrary, we note below for our classifications that the Smokey outcomes are probable rather than definitively known.

After the first set of probabilities and decision classifications was generated, the application guided editors to a new screen from which they could view the decision classifications along with full pre-processing item text for each pair of items. Results were displayed with the most extreme probable false positive values at the top of the screen and the most extreme probable false negatives at the bottom. These two categories were deemed the most crucial for editorial review, as the assumption was made that no review was needed for outcomes where the Smokey decision classification agreed with the enemy item codes.

For the review stage, editors were instructed to compare the item text to the Smokey decision and use their professional judgment to confirm or deny the decision classification for item pairs. If the decision was a probable false positive, but the editor decided the item pairs were in fact enemies, they could label them as either form enemies (not to appear on the same test) or subform enemies (can only appear on the same test if in different non-reviewable subforms). Editors reviewed the items in batches and were allowed to work at their own

Table 1. Decision classifications from Smokey

Smokey outcome (based on 0.05 threshold)	Actual bank flag value	Smokey decision classification
Probable Enemies	Enemies	Probable True Positive – bank status of this item pair coded as enemies is probably correct
Probable Enemies	Not coded as enemies	Probable False Positive – bank status of this item pair not coded as enemies is probably incorrect
Probable Non-enemies	Enemies	Probable False Negative – bank status of this item pair coded as enemies is probably incorrect
Probable Non-enemies	Not coded as enemies	Probable True Negative – bank status of this item pair not coded as enemies is probably correct

pace. Once all pairs in a review subset were reviewed, the medical editors' decisions were used to update enemy flag values and create a new item bank, after which the entire bank was used to generate a new set of logistic regression coefficients, generating a new set of probabilities. As more pairs underwent medical editor review, the expectation was that the model would continue to produce progressively more accurate results, as it would now have more accurate coding of enemies and non-enemies from which to learn.

Additional Interface Elements

As editors were provided the prioritized item enemy decision classifications for each pair of items, the full item text for both items appeared on the screen, enabling editors to review each pair in an efficient manner while referring to the full preprocessing text for each item. Smokey also included an option for editors to mark an item pair for review by a peer, flag the pair for discussion or add notes to a given pair for future reviewers. Marking an item for review was essentially a request for a second opinion, so that a second editor could render their judgement of the pair. The interface also provided a field for the initial editor to enter open-text notes to explain their hesitation regarding the pair's enemy status. Finally, Smokey captured the time that each editor spent on a given pair, across all editorial review, so that the total human review time needed for the pilot analyses could be totaled once all work was completed.

4. Results

The medical editors performed seven iterations of review. There were many more probable false positives (Smokey

Table 2. Detailed results of iterative review of false positives and false negatives across the seven Smokey runs

False Positives					
Run	**Exam Enemy**	**Subform Enemy**	**Not Enemy**	**Review**	**Total**
1	100	0	0	0	100
2	73	27	0	0	100
3	84	16	0	0	100
4	91	9	0	0	100
5	97	3	0	0	100
6	79	21	0	0	100
7	66	0	23	11	100
Total	590	76	23	11	700

False Negatives					
Run	**Exam Enemy**	**Subform Enemy**	**Not Enemy**	**Review**	**Total**
1	7	3	14	1	25
2	7	6	10	0	23
3	12	22	12	0	46
4	33	15	4	0	52
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
TOTAL	59	46	40	1	146

indicating that an item pair should be enemies, but the pair was not tagged as enemies in the bank) reviewed than probable false negatives (tagged as enemies in the bank, but not predicted by Smokey to be enemies). Of the 700 total probable false positive values reviewed, 666 were confirmed as enemies after medical editor review (95.1% accuracy within that subgroup of pairs) and 40 of 146 probable false negatives were confirmed as non-enemies by editorial review. This confirmed the initial suspicion that the bank contained many item pairs that were in fact enemies but had not been tagged as such during previous manual review steps. The results of the review process by iteration are shown in Table 2. It is notable that disagreement between Smokey and editorial review did not occur until the seventh iteration. Ideally, we would have continued the review process for several more iterations to better capture the point where disagreement became so prevalent that the results no longer justified the effort required. Unfortunately, this initial pilot study was limited by staff availability and other operational deadlines, so only seven iterations were possible given those constraints.

The sensitivity, specificity and precision values are shown in Table 3. Sensitivity was calculated as the percentage of total pairs correctly identified by Smokey as enemies, and specificity was the percentage of total pairs correctly identified by Smokey as non-enemies. By the seventh run, both values were higher than 0.70. Precision was calculated as the ratio of true enemies (probable true positives) to total predicted enemies (probable true positives + probable false positives). It should be noted, however, that while sensitivity, specificity and precision are typical metrics used to measure ML model performance (hence their inclusion), their use here is based on a fairly arbitrary baseline threshold (0.05) to consider items enemies. The operational priority is to ensure that editors are provided with a sufficient number of pairs for review, so in our case, the actual numeric value of the threshold (which affects sensitivity, specificity and precision) is largely irrelevant, as long as the model yields a sufficient number of pairs for editorial review.

During debriefing, the editors indicated that they found the Smokey application to be user-friendly and intuitive. For this particular item bank, most probable false positives reviewed across the seven runs were quickly determined to be actual enemies, with no second review needed. The editors more often chose to flag the false negatives for review, as this sometimes necessitated a discussion of why two items that had little or no text in common had been originally tagged in the bank as enemies by previous editors. The entire process of reviewing the Smokey outcomes and making decisions across all seven iterations took less than eight hours.

5. Conclusion

The goal of the development of this application and the pilot analyses was to inform discussion among test

Table 3. Sensitivity, precision and specificity results by Smokey run

Run	F+	F-	T+	T-	Sensitivity	Precision	Specificity
1	889	443	96	8524957	0.17811	0.09746	0.9999
2	1055	404	221	8524705	0.3536	0.1732	0.99988
3	1174	380	355	8524496	0.46853	0.222	0.99986
4	1278	360	443	8524304	0.55168	0.25741	0.99985
5	1373	346	553	8524113	0.61513	0.28712	0.99984
6	1459	335	664	8523927	0.66466	0.31276	0.99983
7	1608	327	772	8523678	0.70246	0.32437	0.99981

development professionals about the potential for a narrow AI application, using NLP and ML elements, to greatly improve accuracy and efficiency of human expert work, like enemy item identification, that is commonly needed to support high-quality test construction. Although the results presented here were based on a prototype and deployed within a short time period for a single exam, we consider them to be tremendously encouraging in this regard. The use of Smokey for these pilot data was successful in drastically reducing the time needed for human reviewers to identify uncoded enemy item pairs in a large item bank and make the necessary changes to item coding. The medical editors found the process easy to follow, and the utility of this kind of application was clearly shown, as fewer than eight hours of staff time were needed to review 846 prioritized item pairs and determine that coding changes needed to be made to enemy codes for 740 of them. This was reassuring, as Smokey was designed, from the start, with the end-user experience in mind.

One unexpected benefit was that, in addition to providing empirical data for the clear-cut examples of enemy items with calculations of semantic similarities and predictions based on the logistic regression, the experience also inspired editors to collaborate about what sort of enemies Smokey was best at identifying and what sort of enemies were not likely to be identified via NLP technology. This resulted in discussions on what elements of item text seemed most useful in determining whether two items are enemies and how the definition of an enemy item pair should be expanded if there are examples of enemy item pairs with relatively little text in common. While enemies are not hard to understand conceptually, not all distinctions between enemies and non-enemies are clear in practice, especially if synonyms or different phrasing can be used to describe the same assessment point. The use of Smokey's iterative review process revealed that two equally qualified human experts could have different opinions on whether the information that could be gleaned from one item would be enough to influence a test-taker's response to another item, especially as the semantic space between them grew larger. For this reason, despite the success of Smokey for this pilot study, the authors view the development of this kind of application, and the utility of NLP methods in general, to be an important step in improving efficiency and accuracy in enemy item identification. The additional challenge is developing a more articulate, specific and less mysterious definition of enemies that can be applied across all items.

One additional constraint that should be acknowledged is that the application of any software of this kind requires a training and implementation process, so that human experts who are used to being the sole decision-makers can become comfortable interacting with, confirming or rejecting judgments made by a system incorporating AI elements. In the case of this pilot, the medical editors were appreciative of the support that Smokey provided, and it was immediately apparent to them that the incorporation of this application would allow them to accomplish a routine task much more quickly and efficiently. It would not be wise for other researchers, however, to assume that all such human experts will be as appreciative and accepting at the introduction of a narrow AI application that replaces the need for their manual efforts, in part or in whole. The authors found during the pilot training that it was helpful to restate what narrow AI can and cannot do, and to consistently frame the development work as one in which the AI application is intended to support the human work by assuming only the small subset of tasks which computers can do faster and more accurately than humans, thus freeing up cognitive space and time for the humans to focus on what they can do best.

The pilot item bank, while sufficiently large for NLP estimations, did set limitations on the generalizability of the results. As noted above, given the expectation that many enemies were not classified, the decision was made not to split the dataset for test and training, as would normally be done, but instead to follow an iterative process. The content classification code used in the regressions was not necessarily independent of the actual text used in the items, which was unavoidable due to the method used for content coding, and this correlation of the cosine similarity indices with the dichotomous content code could have led to artificial overweighting in the final model. The items were often relatively short, factual items that tested recall, and the highly standardized language used in some content areas made Smokey's tasks relatively easy. It remains to be seen if the accuracy and time metrics shown here would be replicated for a dataset composed of lengthier items with less standardized descriptions of patient scenarios. As the level of complexity in the subject matter, vocabulary and sentence structure increases in

test items, so does the need for more human cognitive effort to define the gray areas and amorphous boundaries between enemy item pairs.

The feedback, while unanimously positive, was gathered informally as well. While the limited scope and deployment within a rigid operational timeline prevented a formal survey of the editorial staff, we plan to initiate this kind of effort for all studies going forward, to ensure that we capture the best feedback from the users on aspects such as the ease of navigation and the perceived value added by the application.

In addition, further enhancement and training, the use of alternative NLP techniques such as word embeddings and the inclusion of larger corpora would be useful to enhance an application like Smokey for the more subtle scenarios, allowing the models to correctly flag two items as enemies when the items do not have exact words in common, thus cuing test-takers who saw both items. Another capability that could be added to the application would be the calculation of agreement statistics between editors who reviewed the same item pairs, so that a study could be done to evaluate rate of agreement between editors for all item pairs falling into both probable false positive and probable false negative classifications. Nonetheless, the results of this pilot are incredibly encouraging, and future work is planned to evaluate larger datasets with lengthier, more complex items.

6. Acknowledgements

The authors would like to thank Kirk Becker, PhD for authoring the paper that inspired the initial exploration of what turned out to be Smokey, and Fang Peng, PhD for her invaluable collaboration and knowledge sharing during her time in the NBME Assessment Science and Psychometric Internship Program; Caroline Beaumont and Linda McManus for their contributions on the editorial side and Patricia Kinsman and Andy Shontz for their IT support; and Samson Bear for being the best 600-lb emotional support grizzly bear in history.

7. References

American Educational Research Association. (2018). *Standards for Educational and Psychological Testing.* American Educational Research Association.

Association of Test Publishers. (2021). *Artificial Intelligence and the Testing Industry: A Primer.* Association of Test Publishers.

Becker, K., Kao, S. (May, 2009). Finding stolen items and improving item banks. Paper presentation at the American Educational Research Council, San Diego, CA.

Becker, K. A., McLeod, J. (2013). Automated Item Bank Referencing: A Comparison of NLP Methods.

Cho, E., Xie, H., Lalor, J. P., Kumar, V., Campbell, W. M. (2019). Efficient semi-supervised learning for natural language understanding by optimizing diversity. IEEE Automatic Speech Recognition and Understanding Workshop. PMid: 30744338 PMCid: PMC6599963. https://arxiv.org/pdf/1910.04196.pdf https://doi.org/10.1109/ASRU46091.2019.9003747

Gierl, M., Lai, H., Tanygin, V. (2021). *Advanced methods in automatic item generation* (pp. 138-141). Routledge: New York. https://doi.org/10.4324/9781003025634

Jurafsky, D., Martin, J. H. (2009). *Speech and language processing.* Pearson: Upper Saddle River, NJ.

Kaplan, A., Haenlein, M. (2019). Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations and implications of artificial intelligence. *Business Horizons, 62*(1), 15–25. https://doi.org/10.1016/j.bushor.2018.08.004

Kusner, M., Sun, Y., Kolkin, N., Weinberger, K. (2015). From word embeddings to document distances. In: *Proceedings of the 32nd International Conference on Machine Learning*, pp. 957–66.

Lai, H., Becker, K. A. (May, 2010). *Detecting enemy item pairs using Artificial Neural Networks.* Poster presented at the National Council for Measurement in Education annual meeting, Denver, CO.

Lane, S., Raymond, M. R., Haladyna, T. M. (Eds.). (2016). *Handbook of test development* (pp. 3-18). New York, NY: Routledge.

Mao, X., Zhang, Q., Clem, A. (2021). An exploration of an integrated approach for enemy item identification. *International Journal of Intelligent Technologies and Applied Statistics*, *14*(2).

McCarthy, J., Minsky, M. L., Rochester, N., Shannon, C. E. 1956. A proposal for the Dartmouth Summer Research Project on Artificial Intelligence.

NBME. (2021). *NBME item writing guide: Constructing written test questions for the Health Sciences (6th Ed).* Philadelphia, PA: NBME. URL: https://www.nbme.org/item-writing-guide

Peng, F. (2020). *Automatic enemy item detection using natural language processing* (Doctoral dissertation, University of Illinois at Chicago).

Peng, F., Swygert, K. A., Micir, I. (April, 2019). *Automatic item enemy detection using natural language processing: Latent semantic analysis.* Paper presented at the National Council for Measurement in Education annual meeting, Toronto, CN.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–7. https://doi.org/10.1108/eb046814

Punjabi, S., Arsikere, H., Garimella, S. (2019). Language model bootstrapping using neural machine translation for conversational speech recognition. *IEEE Automatic Speech Recognition and Understanding Workshop.* URL: https://arxiv.org/pdf/1912.00958.pdf https://doi.org/10.1109/ASRU46091.2019.9003982

Semmler, S., Rose, Z. (2017). Artificial Intelligence: Application today and implications tomorrow. *Duke L. & Tech. Rev.,* 16,85.

Sitikhu, P., Pahi, K., Thapa, P., Shakya, S. (2020). A comparison of semantic similarity methods for maximum human interpretability. In 2019 *Artificial Intelligence for Transforming Business and Society, 1, pp.* 1–4. URL: https://arxiv.org/pdf/1910.09129.pdf. https://doi.org/10.1109/AITB48515.2019.8947433

Taulli, T. (2019). *Artificial Intelligence basics: A non-technical introduction*. Apress: Monrovia, CA. https://doi.org/10.1007/978-1-4842-5028-0

Wang, Z., & von Davier, A. A. (2014). Monitoring of Scoring Using the e-rater® Automated Scoring System and Human Raters on a Writing Test. *ETS Research Report Series, 2014*(1), 1-21. https://doi.org/10.1002/ets2.12005

Woo, A., Gorham, J. L. (2010). Understanding the impact of enemy items on test validity and measurement precision. *CLEAR Exam Review, 21*(1), 15–7.