INTELLIGENT AGENT DEVELOPMENT USING UNSTRUCTURED TEXT CORPORA AND MULTIPLE CHOICE QUESTIONS

By

Joseph Johnson

A Dissertation Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: COMPUTER SCIENCE

Examining Committee:

Selmer Bringsjord, Dissertation Adviser

Micah Clark, Member

Eugene Eberbach, Member

Sergei Niremberg, Member

Mei Sei, Member

Boleslaw Szymanski, Member

Rensselaer Polytechnic Institute
Troy, New York

December 2015
(For Graduation May 2015)

CONTENTS

LIST OF FIGURES

ABSTRACT

This thesis explores various approaches for developing an intelligent agent in a particular domain: fraud detection. The framework by which we measure our agent is *psychometric artificial intelligence*, or, more commonly, psychometric AI. As we'll explore further, psychometric AI is AI focused on the creation of agents that can successfully pass tests. This approach offers a number of benefits, including a well-defined domain, a built-in measure for quantifying the efficacy of the agent in the form of a test score, and a rich environment for deploying various forms of AI, including machine learning, natural language processing, computer vision, et cetera. Although, in this work, our attention we'll be centered around natural language processing.

As part of our commitment to this psychometric approach, we set our sights on one particular test in the fraud-detection domain, namely, the Certified Fraud Examiners (CFE) exam, administered by the Association of Fraud Examiners (ACFE). As will be discussed in more detail herein, the ACFE is a governing body overseeing the fraud examiners profession, administering the CFE exam as part of a credentialling process for its members. The CFE exam is a multiple-choice exam whose questions are based on the material of the Fraud Examiners Manual (FEM). Programmatic processing of both the FEM and of a training set of CFE exam questions generate the basis of the agent's knowledge and decisions for answering new questions on the test.

The approaches employed in the work presented herein range over a variety of techniques, from extremely shallow text-processing techniques to deep, cognitive, semantic-representation techniques. Version 1 of the agent focuses on shallow text processing techniques that leverage features of the exam and the high-level structure of the document. Analysis of these shallow algorithms on a training set is then used to apply these algorithms optimally on a test set. As we'll discuss, even these relatively simplistic techniques generate surprisingly decent scores, although not ones at the level of passing. Version 2 picks up where version 1 left off by employing more

sophisticated information retrieval-based approaches to the question-answer task, wherein the agent breaks up the document along much more granular, hierarchical lines reflecting the structure of the manual's sections, subsections, sub-subsections, and so on. Version 3 refines the techniques of Version 2 by incorporating machine learning in order to zero in on the paragraphs within the FEM relevant to each question. Finally, version 4 features an agent with a deep, semantic representation of the problem domain, whose knowledge-base consists of assertions expressed in the *deontic cognitive event calculus* $\mathcal{DCEC}^*$, and which serve as a foundation for the rigorous science of fraud detection.

It is hoped that by the end of this exploration, the reader will have a solid understanding of the various techniques discussed herein, an appreciation of the benefits of using psychometric AI as the backdrop for intelligent-agent development, and some insights into the relative potential of each of these approaches.

CHAPTER 1
INTRODUCTION AND OVERVIEW

1.1 Background

Question Answer (QA) [1] is a prominent and growing sub-field of natural language processing, and of the larger field of AI [2]. QA systems [1] consist of agents that provide responses (correct ones, hopefully) to questions posed in natural language that call upon the agent to retrieve and reason upon vast stores of information. QA systems are typically based on one of two approaches: an information-retrieval [3] approach or a knowledge-based [4] approach. Some of the most successful systems as of late, however, utilize a blend of these two approaches, such as IBM's Watson. A knowledge-base [4] approach offers the advantages of allowing automated reasoning, and thus, justifications for answers. But the knowledge bases involved have been manually generated by domain experts and knowledge engineers — a time-consuming, labor intensive endeavor. An information-retrieval approach [3], where candidate answers are screened from passages retrieved from documents retrieved from a core information-retrieval (IR) system, are designed without the need to manually curate a knowledge base, but typically offer no justification of answers. In fact, these systems have no understanding of the reasons for the answers they put forth, but instead typically provide answers based on statistical approaches that are utilized in black-box fashion, giving the user no insight into the rationale for responses.

1.2 Goals

The work proposed here seeks to advance the state of the art of QA by exploring this area within the context of developing an agent designed to correctly answer multiple-choice questions covering the domain of fraud detection. As iterations of this agent are developed and presented within this document, there will be an eye toward providing justification of answers, and as such, providing the user insight into the rationale for the response of the agent. A key element of this approach is

the use of multiple-choice questions as an information resource for generating assertions in the knowledge-base. Such questions are potentially highly useful. They are inherently semistructured and, as will be discussed, typically hold multiple pieces of information in their stems, correct choices, and also, incorrect choices.

In short, then, the goal of this thesis is threefold: First, introduce the use of psychometric AI [5, 6] for researching methods in QA, where psychometric AI, here, is the development and validation of intelligent systems by applying their algorithms to taking multiple choice tests. Second, explore techniques by which the agent can answer questions while providing reasoning for its answers, and specifically, using proof-based techniques. Third, lay a foundation for the rigorous science of fraud detection. Finally, after having explored the various techniques presented in this thesis, make recommendations about future research in the field of QA. In particular, we consider ways in which semantic approaches can be utilized in conjunction with the techniques presented here for developing knowledge-bases to advance the art of QA in the semantic direction.

1.3 The Intelligent Agent Defined, For This Project

In an effort to further detail the goals listed above, the definition of intelligent agent is made more explicit, here, as one capable not only of making intelligent decisions, but also of providing a justification for those decisions. This is to be contrasted with agents of other AI [2] systems, where decisions are provided without any justifications. This type of agent, which is the goal for this project, will be termed a "reasoning agent". It should also be mentioned the reasoning agent provides justifications in situations where there is uncertainty of outcome (which is typically the case for the domain in this project).

1.4 An Overview of the Approach

This section discusses the overall approach to developing the reasoning agent, as defined above. The overarching idea is to develop successively more sophisticated agents — starting with a naive agent that uses only shallow techniques that largely leverages question features with patterns among questions of certain

types discovered from exploring a training set. In version 2, we elevate the level of sophistication of the agent, but still limit it to shallow techniques, by basing answers on information-retrieval-based techniques that incorporate sophisticated query-generation and document-collection-development techniques. In version 3, we explore semantic approaches by attempting to leverage training-set questions to answer test-set questions. And finally, in version 4, we consider a sophisticated agent that uses deep, semantic techniques to answer questions.

1.4.1 Generation of KB assertions from Text Corpora and MCQs

In version 3, where as mentioned, the agent utilizes a knowledge-base to answer questions, we'll look at the generation of assertions from both the text corpus (the CFE Manual, discussed below), and from other multiple-choice questions that overlap in the problem domain. By overlap, we mean that the question stem and correct answer of one question serve as the basis for assertions from which inferences can be made about the answer to a second, separate question. Unfortunately, as we'll discuss later, the sparsity of the training set and test set were such that no such overlap could be found, so engineered test-battery questions were created to demonstrate this approach.

1.4.2 Reasoning with Uncertainty

In order to cover the nondeterminism inherent when extracting information using NLP techniques, the agent will incorporate uncertainty into its decisions. As we'll see in versions 1 and 2, the agent quantifies uncertainty by analysis of its various algorithms on the training set, and then utilizes this information to optimize its accuracy on the test set.

1.4.3 Verification of the Agent via Test-Taking

The method proposed in this work includes verification of these algorithms by assessing performance on multiple-choice questions. In this project, the domain of fraud detection will be used, for which there is a well-developed industry of testing subjects on knowledge in this domain. The principal organization responsible for this and its examination process are described below.

1.4.3.1 The ACFE and the CFE Exam

The ACFE (http://www.acfe.com) describes itself on its website as "the world's largest anti-fraud organization and premier provider of anti-fraud training and education." Generally speaking, in order to become a readily employable expert in the field of fraud detection, certification by this organization is required. (Among notable certified members of the ACFE is Harry Markopolos, the American forensic accountant who achieved fame by being the first to have uncovered the ponzi scheme perpetrated by Bernard Madoff and desperately tried to warn government securities officials years before the scam collapsed in the midst of the 2008 financial crisis [7].) The ACFE has well-defined requirements for becoming certified, based on a point system that considers a combination of professional experience and academic credentials. However, the CFE exam is the credentialing centerpiece for the ACFE, and the details of this exam are described briefly below.

The CFE Exam is a computer-based exam. The mechanics for preparing for and taking the exam begins with downloading a software package from the prep course page of the ACFE website. This package includes the exam software, the Fraud Examiners Manual (on which the test is based), and a self-study application consisting of a battery of sample test questions, a complete practice exam, and tools for monitoring progress.

The CFE exam consists of 4 sections, listed below:

- Financial Transactions and Fraud Schemes
- Law
- Investigation
- Fraud Prevention and Deterrence.

Each section consists of 125 multiple-choice and true-false questions. The candidate is limited to 75 seconds to complete each question and a maximum total allocated time of 2.6 hours to complete each section. Each CFE Exam section is taken separately. The timing for each section is subject to the candidates discretion. However, all four sections of the exam must be completed and submitted to the ACFE for grading within a 30-day period.

1.4.3.2 CFE manual

The Fraud Examiners Manual (known by the CFE Agent as the 'CFE Manual') is the text corpus on which all of the questions of the CFE Exam are based. Each question includes a section heading that loosely maps to an individual section within the manual. However, these sections are rough-grained; that is, relatively large (often 20–100 pages).

CHAPTER 2
Version 2 – A More Sophisticated Agent

In this chapter, we extend the agent developed in Chapter 3, making it more sophisticated by imbuing it with improved ability to locate the sections of the CFE manual where answers to questions it attempts to answer are located. In version 1 of the agent, the agent was limited to the very coarse-grained sections as defined by the question sections assigned to the questions. Unfortunately, these sections of the manual were effectively so large, it was very difficult for the agent to drill down to a point where text directly related to the question at hand could be pulled out. For intellectual-property-theft questions, for example, the "Theft of Intellectual Property" section of the manual is 89 pages long. Certainly, for any question on this topic, pulling the entire section of the CFE manual will assure that the answer sought lies somewhere within our target document. But with 89 pages of text in the section, we need to narrow the search to a much smaller passage. Version 2 of the agent attempts to do just that.

This chapter is laid out as follows: First, we review the basic elements of Information Retrieval, the branch of AI dealing with retrieving documents from within a large collection related to an information need. Next, we explore the open source software, Lucene, an Apache sofware program that provides information retrieval functionality. This package was used in the implementation of Version 2 of the CFE agent. Then, we look at the method used to break up the CFE manual into finer-grained sections, leveraging the table of contents and text metadata as means for dividing up the text along reasonable semantic lines into a collection of documents. And finally, we analyze a set of algorithms that were implemented using these information retrieval tools and this document collection to answer questions with improved accuracy over Version 1.

One final note before proceeding: The terms, "training set" and "test set" are used throughout this chapter in reference to the battery of questions against which the algorithms discussed below were applied. Indeed, we talk repeatedly

about testing our algorithms against the training set, specifically. It is important to note here training set is a term of convenience, here, and not necessarily one that should be taken to suggest a set for which we've optimized parameters of the algorithm a la machine learning. In fact, in the context of this chapter, the algorithms we discuss involve no training, per se, and so, it is perfectly valid to measure or test the performance of these algorithms against the "training set". (In chapter 5, however, where we *do* employ machine learning, the training set and test set are used according to their conventional definition; that is, the algorithms of chapter 5 are trained on the training set and tested against the test set.) Lastly, it should be mentioned that at the end of this chapter, we use performance of each algorithm on the questions of the training set to determine the optimal algorithm to use on each question in the test set in order to measure overall performance of Version 2 of the agent.

2.1 Information Retrieval

Information Retrieval (IR) is the branch of AI dealing with the creation of software that retrieves documents of an unstructured nature from among a large collection of documents in response to an information need expressed as a natural language query [3]. Of course, Google and Yahoo search engines are exemplars of implementations of the techniques subsumed by this subfield of AI. However, not only do search engines employe information retrieval - the field of question-answer does, as well.

Of importance in our discussion below are some terms that will be used, including document, document collection, and vocabulary. A document is a unit of data, typically unstructured or semi-structured, and typically expressed in natural language. (One of the fundamental questions to consider in IR is how to define a document - a sentence? a paragraph? a chapter? The answer typically depends on the nature of the problem domain. A document collection is simply a collection of such documents, as defined above. Finally, a vocabulary, $V = \{t_1, t_2, t_3, ..., t_n\}$ is the set of all terms over which the contents of the documents are defined.

2.1.1 Boolean Retrieval

Boolean information retrieval is based on the idea of dividing up the document collection into two sets for each query - one set of documents which meets the requirements of the boolean query and the other set whose documents does not. Boolean queries are structured as conjunctions of disjunctions; that is, of the form of query, q, where

$$q = (W_i \vee W_k \vee ...) \wedge ... \wedge (W_j \vee W_s \vee ...) \tag{2.1}$$

where $W_i = t_i, W_k = t_k, W_j = t_j, W_s = t_s$, or $W_i = NON\ t_i, W_k = NON\ t_k, W_j = NON\ t_j, W_s = NON\ t_s$ and where t_i means that t_i exists in the document and $NON\ t_i$ means it does not [8].

2.1.1.1 Term-Document Incidence Matrix

Boolean retrieval may be implemented using a data structure called a term-document incidence matrix, A, where A is a two-dimensional array in which the ith row denotes the ith document in the document collection and where jth column denotes the jth term in the vocabulary, and where $A[i, j] = 1$ if the term, i exists in the document, j, and 0 otherwise. Based on A, the processing of a query involves finding those documents for which there is a 1 in each of the rows corresponding to the terms of the query [3].

A significant pitfall of this method is that it requires a vast amount of memory. Consider an example consisting of 1 billion documents and a vocabulary of 50,000 words. Then, the size of the matrix is 50 trillion (1 billion x 50,000). This matrix is also highly sparse since any given document has on average a small number of words relative to the size of the vocabulary. Suppose, for example, each document has 1,000 words. Then, for each document, among the 50,000 elements in its corresponding row of the matrix, only 1,000 (2%) are non-zero [3].

2.1.1.2 Inverted Index

An alternative implementation for boolean retrieval is based on a different data structure, referred to as an inverted Index. An inverted index is a hash table

in which the keys are the terms in the vocabulary and for each such key, the value is a linked list of all of the document identifiers in which that term appears. This technique exploits the sparsity of the term-document matrix requiring memory only for those elements in which $A(i, j) = 1$ [3].

2.1.2 Ranked Retrieval

Boolean retrieval has the unfortunate drawback of returning a set of documents in response to a query as a set of equally ranked units through which the user must sift in order to find the information sought [3]. Sometimes, this sifting can be sizable task depending on the number of documents returns from the boolean query. Ranked retrieval, on the other hand, is IR in which documents are ranked according to a score that measure the degree of similarity between the query and the terms of the document and in which documents are returned in decreasing sorted order of this score [3].

The Vector Space model (VSM) is one approach to ranked retrieval, and is based on representing the documents in the collection and the query as vectors and where documents are scored according to a measure of similarity between their vector representations and that of the query [3]. There are a number of measures, or weights, that are typically incorporated into a vector representation as discussed below.

The log of the term frequency, $\log_{10}(tf)$ is a measure of the number of occurences of a particular term in the document. (The log is used as opposed to the term frequency, itself, in order to dampen the effect for each additional occurrence of a term.) The log-frequency weight of term, t in document, d is given by [3]:

$$w(t, d) = 1 + \log_{10}(tf_d), \text{where } tf_d > 0, 0 \text{ otherwise} \tag{2.2}$$

Inverse document frequency is a measure of the relative scarcity of a term in a document relative to the other documents in the collection. A higher weight is assigned to a term that appears in relatively few documents compared with one that appears in many. The inverse document frequency, $idf(t)$, can be calculated for each

term, t, of the query as follows [3]:

$$idf(t) = \log_{10}(N/df_t), \tag{2.3}$$

where N is the number of documents in the collection and df_t is the number of documents containing term, t. Again, as was the case for the measure of term frequency, the log is used here to moderate the effect of the measure.

We can combine these two measures such that for each term, t, in each document, d, we have [3]:

$$w(t,d) = (1 + \log_{10}(tf_d)) \times \log_{10}(N/df_t). \tag{2.4}$$

The vector representation of a document, d, is $\vec{d}$, where

$$\vec{d} = [w(t_1,d), w(t_2,d), w(t_3,d), ..., w(t_n,d)], \tag{2.5}$$

where, as defined above our vocabulary, $V = \{t_1, t_2, ..., t_n\}$. Note, queries can also be vectorized in the same way.

In order to measure the relative relevance of document, d, to the query, q, the vector representations of both d and q, $\vec{d}$ and $\vec{q}$, are harnessed to compute a quantitative measure of the level of similarity between the two vectors using the concept of cosine similarity. Informally, the cosine similarity of two n-dimensional vectors is a measure of the cosine of the angle between them in n-dimensional space. Normalization of the lengths of the vectors is also incorporated into this calculation to assure that longer documents' weights do not outsize the weights of shorter (but perhaps, similar) documents, (such as the query itself, which is commonly much shorter than the document against which it is compared).

$$\begin{aligned} \text{sim}(q,d) &= \frac{\vec{q} \cdot \vec{d}}{|\vec{q}||\vec{d}|} \\ &= \frac{\vec{q}}{|\vec{q}|} \cdot \frac{\vec{d}}{|\vec{d}|} \end{aligned} \tag{2.6}$$

For a document, d whose length-normalized-vector representation is similar to that of q, the angle between its vector and that of q should be small, and thus have a cosine near 1. For those documents not similar to q, the cosine measure will tend toward 0, (note that all of the terms in these vector-representation vectors are greater than or equal to 0). Under the document vectorization approach, the highest K documents are returned in response to a query, q, in order of decreasining cosine similarity, where K is an arbitrary figure intended to limit the size of the return set.

2.2 Transforming the CFE Manual into a Document Collection

The CFE Manual, the definitive study guide for the CFE exam, is a text corpus structured as a text book. As such, it is structured hierarchically, as most textbooks are, complete with features embedded in the text that make this hierarchy apparent. The most obvious feature is the table of contents (TOC). In fact, the CFE Manual has a number of tables of contents, including a main table of contents, at the front of the manual, and a set of area-specific TOCs - one for each of the major test areas - Financial Transactions and Fraud Schemes, Law, Investigation, and Fraud Prevention and Deterrence. Figure 2.1 shows a section of the TOC relating to Financial Statement Analysis, a topic contained in the are of Financial Transactions and Fraud Schemes. The summary TOC combined with the area-specific TOCs combined with text features (capitalized sub-sub-sub section titles within the text itself) were all used to programmatically break up the manual into a hierarchical structure of documents. Figure 2.2 shows a portion of this document structure, where on the left we see the Bankruptcy Fraud subsection, a subtopic of Financial Transactions and Fraud Schemes, and the breakdown of documents, each of which named according to a numeric identifier and a title corresponding to a title for the subsection, along with indentation showing its level in the document tree. On the right side, we see the contents of one of the documents covering the subtopic of Bankruptcy Court. Notice that in each document we have not only the text of the section but also a a title field (Bankruptcy Court), a question section field (Financial Transactions and Fraud Schemes), a path giving the sequence of

Figure 2.1: A section of the table of contents from the CFE Manual

section titles starting from the root node of the CFE manual hierarchy to the current document, and finally, the stemmed contents of the document, using the Porter Stemmer algorithm. All of these elements were compiled for each document as possible inputs to algorithms developed downstream for answering questions.

2.3 Lucene – A Tool for Information Retrieval

Lucene, https://lucene.apache.org/, is a highly popular Apache software product that implements IR using a combination of Boolean Retrieval and the VSM [9]. First, it narrows the document set using boolean retrieval, and then it ranks the remaining documents using VSM. The algorithms described below were implemented using this tool.

However, before implementing any algorithms, the document collection into which the CFE manual was decomposed was indexed using the Lucene software. Indexing is the process of essentially creating files containing the underlying data stuctures required by Lucene's combined boolean retrieval/VSM retrieval algorithm, including an inverted index for the collection and document vectorization data. In fact, Lucene indexes were created for each question section of the manual, where each question section contains the documents from which an answer to a question pertaining to the material in that section. When implementing the QA algorithms discussed

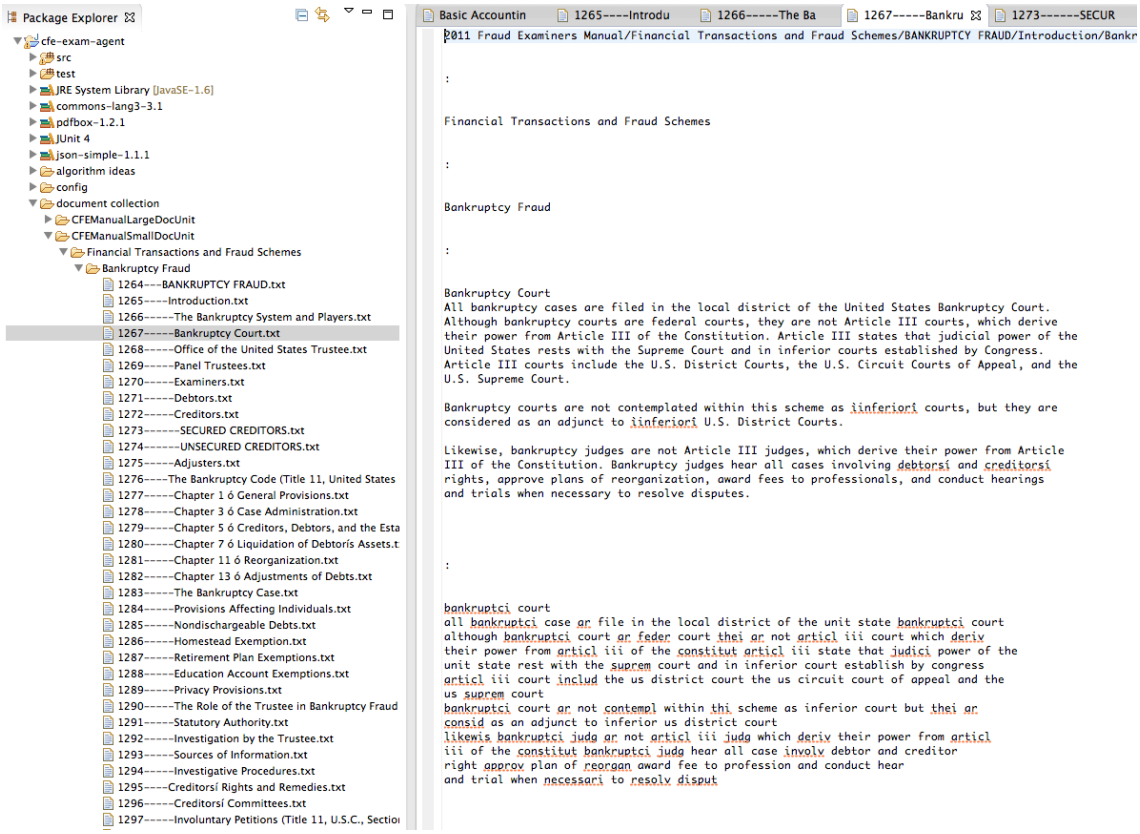


Figure 2.2: The CFE Manual as a Document Collection

below, the first step to any of these algorithms was locating the proper index within which to search for relevant documents, based on the exam section/question section that corresponds to the question at hand. Fig 2.3 shows a portion of the lucene indexes created by this process. Notice that for each question section within each exam section, there are three binary files created by the lucene indexer component that contain the inverted index and document vectorization information required for the query processing component to be used in the algorithms discussed below.

One other thing of note is that as part of this process, the contents field and the title field were stemmed according to the Porter Stemmer algorithm. Stemming is a form of semantic normalization, where words offering different senses of the

same semantic unit are transformed so that they are treated as equivalent during the document scoring computation in the IR process. For example, different words for run - run, ran, running - should all be considered semantically equivalent in most circumstances in IR, and stemming these words accomplishes this goal.

2.4 Analysis Tools for Algorithm Development

As outlined in prior chapters, the goal of the CFE agent is to answer questions correctly while providing justification for those answers. As algorithms were developed toward this end, and in particular, as we attempted to refine the accuracy of the agent by making its search functionality more fine-grained, it was determined early in the process that one of the most critical pieces of information was to understand how to target each type of question - What features for a given type of question could be exploited when searching for an answer? Specifically, how does the answer present itself in the manual to a question of a given type? Is it contained in a single document, or multiple documents? Are terms found in the options commonly found in the contents of the document, or are they found in the title? Depending on the answer, how often is that the case? Is it always true, or only sometimes? Tools that aided in this investigation were critical to the development of "smarter" algorithms.

At a macro level, the profiler component, developed for Version 1 of the agent provided an initial analysis tool. As discussed, it supplied a breakdown of question by macro-features, as well as the success rate of the initial algorithms created for that version. And here in Version 2, the profiler would be used again. However, analysis at a greater level of detail was needed.

One component, called the Question Server, was created to at least partially meet this need. Simple in concept, it would select/pose to the agent only the questions of a particular profile, whether the desired profile be a definition question, a long answer question, and so on. This provided a means for zero-ing on each type of question in isolation, allowing for various theories and insights to be developed about the best approach for each question. Figure 2.4 shows output from the Question Server for definition/NOT questions, questions whose options are a small number of words (and are thus, typically relate to a definition of a conept), and contain the

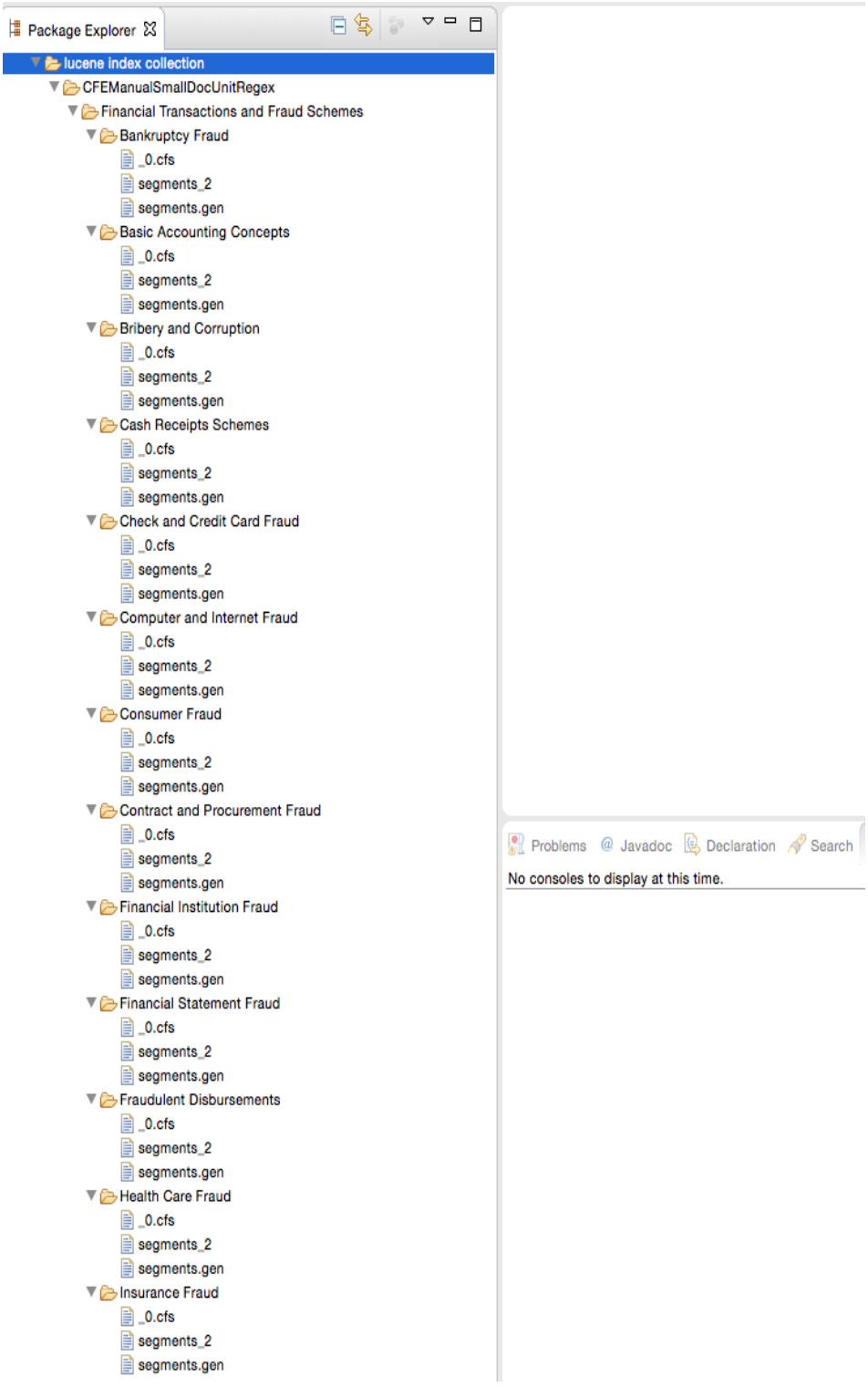


Figure 2.3: Lucene Indexes for a Portion of the Question Sections

```
Question 1 of 27:
Basic Accounting Concepts 8

Basic Accounting Concepts

Which of the following is NOT one of the major standards of generally accepted accounting principles?

a) Accuracy
b) Going concern
c) Matching
d) Full disclosure

a) Accuracy

Question 2 of 27:
Bribery and Corruption 2

Bribery and Corruption

Which of the following is NOT a type of a corrupt recipient?

a) The "rule breaker"
b) The "big spender"
c) The "complainer"
d) The "gift giver"

d) The "gift giver"

Question 3 of 27:
Cash Receipts Schemes 5

Cash Receipts Schemes

Which of the following is not a skimming scheme according to Corporate Fraud Handbook?

a) Understated sales
b) Forged checks
c) Short-term skimming
d) Unrecorded sales
```

Figure 2.4: Question Server Component Targeting Definition/NOT Questions

term, "not" in the stem, implying the task is to identify the "odd man out" among the options.

A second component, called the Algorithm Tester, was created on the shoulders of the Question Server, which would demonstate the behavior of each algorithm as it was applied to each question of a given type. This, too, was instrumental in algorithm development.

2.5 Algorithms for Version 2

This section describes the algorithms implemented for Version 2 of the CFE Agent. These algorithms rely heavily on IR, as implemented in Lucene. They also tend to each target a particular question type, in particular, the definition questions - that is, those questions in which each of the options is only a short phrase (consisting of four words or less), and which thus are thought to be most likely the kind of question in which the stem contains a phrase that defines a given concept and the examinee must choose the correct concept from among the four options to which that phrase corresponds. If we refer back to Figure **??**, we see that if we total up all the counts for question profiles whose description include the term, "definition" in them, (definition, definition/not, definition/except, ...), the total number of "definition" questions comes to 475, more than half of all 868 multiple choice questions in the training set. So, developing more refined algorithms that target this type of question is appears to be a good choice of where to focus our efforts to optimize our agent.

2.5.1 Concept Match Version 1

Concept Match Version 1 leverages IR to on both the question stem and on each of the question options in order to determine the option that fits best with the stem. The essence of the algorithm is to first, conduct an IR query against the document collection based on the terms of the question stem, then, conduct a query based on the terms of each of the question options, and finally, select the option whose return set has the "best overlap" with the return set for the question stem. How do we define "best overlap"? For the purpose of this algorithm, a return set with "best overlap" is the return set that includes the document that occurs in the return set of question stem query and posseses a score higher in that return set that is greater than any other document in the return sets of the other option queries.

Consider an example. Figure 2.5 shows a Bankruptcy Fraud definition question. (Although the criterion for being a definition question has only the naive requirement that all options be no more than 4 words, this example demonstrates how this criterion is often sufficient for properly categorizing this type of question,

since this question really *is* a definition-type question.) The justification by the agent shows its reasoning - First, it shows the return set for each option of the four options in the question, including for each document its title, id (automatically assigned by the Lucene indexing component), and score. (This example is particularly convenient as each option has the simplifying characteristic of at most one return document in its return set. This is not always the case.) Next, the agent shows return set for the query based on its stem, "a person who holds a perfected security interest against a person filing bankruptcy". The return set is sorted by decreasing score. Now, the option Secured creditor option has a result set that includes the "SECURED CREDITOR" document (it's capitalized because that's the way it appears in the text for the manual), which possesses a higher score in the question stem return set than any of the other documents in the return sets for the other options. (In this particular case, it's also *the* high scoring document in the question stem return set, although this fact is not a requirement of the algorithm.) This means that the "best overlap" is the "SECURED CREDITOR" document, and as a result, the agent picks option, "a) Secured creditor", the correct answer.

Some further details about the mechanics of this algorithm should also be mentioned here. The algorithm conducts a query against the document collection based on the stem to return a collection of 10 documents (this number chosen arbitrarily) relating to the question. This query is conducted against the *contents* field of each document. (Note that as alluded to previously, each document consists of four fields - a title field, a contents field, a question section field, and a path field.) This means that when retrieving documents in response to the query, the terms in the contents field, exclusively, are used to determine the relevance of the document to that query. The document's title and path are not considered (at least not in this algorithm). It should also be noted that before conducting the query, functional phrases including "is referred to as", "are referred to as", "which of the following", and so on, were removed from the stem as they do not offer semantic information about the nature of the question. Also, the words of the question stem were stemmed just as the terms in the contents field of each document were during the indexing process. This is necessary in order for the scoring to work properly.

As for the queries based on each of the question options, they are executed against the *title* field of each of the documents. So, in order for a document to be found to match well to an option, it must have one or more terms in its *title* in common with those of the option. This greatly narrows the range of documents that produce high ranked hits, and this was by design. It was noticed that on more than a few questions, the options mapped nicely to subsections whose titles aligned closely with the options. Unfortunately, however, in the event the question options did not have sibling sections in the manual, this algorithm is destined not to perform well. This is a calculated risk built into the agorithm.

2.5.1.1 Agent Justification for Selected Answer

Notice, in the above example, that the agent shows its reasoning for its argument. By giving the scored return set for the question stem, and for each option, the agent provides the detail for its argument for the option it selects. It is clear from the detailed output that among the documents in the returns set for the question stem query and among the set of documents in the return sets for the question options queries, there are two documents that appear in both - SECURED CREDITORS, (id = 9) and UNSECURED CREDITORS (id = 10). Since the SECURED CREDITORS document earns a higher vectorization match score of 0.3375, the reasoning for selecting the Secured creditors, option a, is clear.

2.5.1.2 Concept Match V1 Performance

Finally, we look at the performance of the Concept Match Version 1 algorithm. Figure 2.6 shows output from the algorithm tester tool described above, showing that among the 150 questions in the training set that were classified as strictly definition questions, (there were others that met the criterion for a definition question, but were classified in different but related categories, such as in the definition/not category, or definition/except category), this algorithm correctly answered 56 questions, or 37.3% – not an outstanding rate. In fact it is lower than the rate of the Version 1 agent on these questions (41.3%) using the maximum frequency algorithm based on the much shallower breakdown of the CFE manual into question sections. However, these results also show that for 46 questions this algorithm produced no answer

```
Bankruptcy Fraud

A person who holds a perfected security interest against a person filing bankruptcy is referred to as which of the following?

a) Secured creditor
b) Judgment debtor
c) Judgment creditor
d) Unsecured creditor

Doc results for: Secured creditor
1. SECURED CREDITORS(9)  0.9063726

Doc results for: Judgment debtor
1. Chapter 7  Liquidation of Debtors Assets(16)  0.54382354

Doc results for: Judgment creditor
** no docs returned **

Doc results for: Unsecured creditor
1. UNSECURED CREDITORS(10)  0.9063726

conceptDocs: {16=1, 9=0, 10=3}
Stem (lower case): a person who holds a perfected security interest against a person filing bankruptcy

Doc results for stem: a person who holds a perfected security interest against a person filing bankruptcy
1. SECURED CREDITORS(9)  0.33575153
2. Paragraph 7  Fraudulent Transfer or Concealment(44)  0.3160012
3. UNSECURED CREDITORS(10)  0.2941953
4. Knowing Disregard of Bankruptcy Law or Rule(49)  0.23285939
5. Adverse Interest and Conduct of Officers(48)  0.18710557
6. Paragraph 4  False Claims(41)  0.17282711
7. Embezzlement Against the Estate(47)  0.1723838
8. Paragraph 5  Fraudulent Receipt of Property(42)  0.15566334
9. Paragraph 8  Fraudulent Destruction or Alteration of Documents(45)  0.13475572
10. Paragraph 9  Fraudulent Withholding of Documents(46)  0.12726296

Option selected: a) Secured creditor
Correct!
```

Figure 2.5: Concept Match V1 Example

at all. This was because the laser-focused question option queries against the title field of the documents in some cases returned 0 documents for *all* options, thereby causing the algorithm to fail. Figure 2.7 shows an example of this scenario. With no documents in the set of return sets for the option queries, the algorithm has no other choice than to return -1, (for no option selected). In the next algorithm, Concept Match V2, however, this issue is addressed.

```
Total questions answered: 150
Number question correctly answered: 56(0.37333333333333335)
number of questions where option selected is -1 (no selection): 46
```

Figure 2.6: Performance of Concept Match V1 on Definition Questions

```
Financial Statement Fraud 9

Financial Statement Fraud

Revenue is recognized on long-term construction contracts under one of two methods. Which of the following is one of those methods?

a) Disbursement/recovery method
b) Partial-contract method
c) Percentage-of-completion method
d) Cost-to-market method

Doc results for: Disbursement/recovery method
** no docs returned **

Doc results for: Partial-contract method
** no docs returned **

Doc results for: Percentage-of-completion method
** no docs returned **

Doc results for: Cost-to-market method
** no docs returned **

conceptDocs: {}
Stem (lower case): revenue is recognized on long-term construction contracts under one of two methods.  is one of those methods

Doc results for stem: revenue is recognized on long-term construction contracts under one of two methods.  is one of those methods
1. LongTerm Contracts(16)  0.46679574
2. Improper Asset Valuation(21)  0.19583732
3. DEBT TO EQUITY RATIO(62)  0.1853602
4. Financial Statement Fraud Schemes(5)  0.12198833
5. Capitalized Expenses(36)  0.11294544
6. Trends in Financial Statement Fraud(4)  0.10464896
7. Premature Revenue Recognition(11)  0.10460665
8. Percentage Analysis  Horizontal and Vertical(51)  0.08904613
9. BOOKING FICTITIOUS ASSETS(28)  0.085955516
10. Concealed Liabilities and Expenses(34)  0.07611202

option selected: -1
Incorrect.  Correct answer: Percentage-of-completion method
```

Figure 2.7: Concept Match V1: An Example Where No Docs Are Returned for Options

2.5.2 Concept Match Version 2

Concept Match V2 extends Concept Match V1 by addressing two major concerns. The first is the situation outlined above in which for the query options no documents are returned because of the tight focus of the queries on the title field. In ConceptMatchV2, if this scenario occurs, the options docs set is rebuilt, but instead of searching on the title field, the the search is performed on the contents field, thus, loosening the focus of the query resulting in a greater likelihood of documents in the return set that also occur in the question stem return set.

The two figures, Figure 2.8 and Figure 2.9 show the output for Concept Match V2 for the same example as Figure 2.7 shows for Concept Match V1. Again, this output shows the failure to return any documents or the option queries against the title field. But whereas the V1 algorithm gives up and returns -1, V2 redoubles its efforts by re-issuing the same option queries against the contents field. We see that in this example this form of recourse results in the algorithm producing the correct answer.

The second major concern this this algorithm addresses is demonstrated by the following example shown in Figure 2.10. In this case, we have a question in which the highest scoring document, (and by the way, the document which does, in fact, contain the answer to this question), "The Business Profile Analysis", (id = 38), in the question stem return set is returned for *two* options - option b, Preparing the business profile, and option c, Preparing the vertical analysis. Because Concept Match V1 simply loads these documents into the concept documents hash map in order by option, document 38 is *initially* mapped to option b (the correct answer). But then, this mapping is overwritten with a mapping to option c. We can see this association in the display of the contents of the concept docs hash map, (the line in the output that starts with "conceptDocs:"), in which we see the key/value association, 38=2, signifying that document 38 is associated with option 2 (i.e., option c; note that the option ids are 0-based, so option a is 0, option b is 1, option c is 2, and option d is 3). As a result, the agent gets this question wrong.

Concept Match Version 2 corrects this problem by recognizing a situation in which a document is included in multiple question-option-return-sets. In this case,

```
Financial Statement Fraud 9

Financial Statement Fraud

Revenue is recognized on long-term construction contracts under one of two methods. Which of the following is one of those methods?

a) Disbursement/recovery method
b) Partial-contract method
c) Percentage-of-completion method
d) Cost-to-market method

Doc results for: Disbursement/recovery method
** no docs returned **

Doc results for: Partial-contract method
** no docs returned **

Doc results for: Percentage-of-completion method
** no docs returned **

Doc results for: Cost-to-market method
** no docs returned **

optionsDocs: {}

Stem (lower case): revenue is recognized on long-term construction contracts under one of two methods.  is one of those methods

Doc results for stem: revenue is recognized on long-term construction contracts under one of two methods.  is one of those methods
1. LongTerm Contracts(16)  0.46679574
2. Improper Asset Valuation(21)  0.19583732
3. DEBT TO EQUITY RATIO(62)  0.1853602
4. Financial Statement Fraud Schemes(5)  0.12198833
5. Capitalized Expenses(36)  0.11294544
6. Trends in Financial Statement Fraud(4)  0.10464896
7. Premature Revenue Recognition(11)  0.10460665
8. Percentage Analysis  Horizontal and Vertical(51)  0.08904613
9. BOOKING FICTITIOUS ASSETS(28)  0.085955516
10. Concealed Liabilities and Expenses(34)  0.07611202

Search for options docs based on title field unsuccessful.
Repeating search for options docs using contents field...
```

Figure 2.8: Concept Match V2: Fixing the No Docs in Option Queries Return Sets Problem, Part 1

the algorithm maps the document to the option for which that document earned the highest rank score. (Note that the lucene scoring algorithm is normalized so that scores for documents from different queries may be compared.) Explaining this a bit more rigorously, for each concept document, D, with score, S, with respect to option O, if there is already an entry in the hash table for which the there is key/value pair, $D = (O', S')$, where O' is a different question option and S' is the score for D with respect to O', then scores, S and S' are compared, and the option

```
Doc results for: Disbursement/recovery method
1. LongTerm Contracts(16)  0.10989416
2. Concealed Liabilities and Expenses(34)  0.049146157
3. Tax Return Review(65)  0.049146157
4. Fictitious Revenues(6)  0.035104398
5. Multiple Deliverables(17)  0.035104398
6. Percentage Analysis  Horizontal and Vertical(51)  0.035104398
7. Ratios Analysis(54)  0.028083518
8. LiabilityExpense Omissions(35)  0.017552199

Doc results for: Partial-contract method
1. LongTerm Contracts(16)  0.12732214
2. Concealed Liabilities and Expenses(34)  0.05694019
3. Tax Return Review(65)  0.05694019
4. Fictitious Revenues(6)  0.040671565
5. Multiple Deliverables(17)  0.040671565
6. Percentage Analysis  Horizontal and Vertical(51)  0.040671565
7. Ratios Analysis(54)  0.03253725
8. LiabilityExpense Omissions(35)  0.020335782

Doc results for: Percentage-of-completion method
1. LongTerm Contracts(16)  1.5838112
2. Concealed Liabilities and Expenses(34)  0.06884125
3. Tax Return Review(65)  0.06884125
4. Fictitious Revenues(6)  0.049172323
5. Multiple Deliverables(17)  0.049172323
6. Percentage Analysis  Horizontal and Vertical(51)  0.049172323
7. Ratios Analysis(54)  0.03933786
8. LiabilityExpense Omissions(35)  0.024586162

Doc results for: Cost-to-market method
1. LongTerm Contracts(16)  0.18363643
2. Improper Asset Valuation(21)  0.1624789
3. Inventory Valuation(22)  0.1624789
4. Concealed Liabilities and Expenses(34)  0.0821247
5. Tax Return Review(65)  0.0821247
6. Fictitious Revenues(6)  0.058660503
7. Multiple Deliverables(17)  0.058660503
8. Percentage Analysis  Horizontal and Vertical(51)  0.058660503
9. Ratios Analysis(54)  0.046928402
10. LiabilityExpense Omissions(35)  0.029330252

optionsDocs: {16=2, 65=3, 17=3, 34=3, 51=3, 35=3, 21=3, 6=3, 54=3, 22=3}

Options doc with best match: LongTerm Contracts(16)
Option selected: c) Percentage-of-completion method
Correct!
```

Figure 2.9: Concept Match V2: Fixing the No Docs in Option Queries Return Sets Problem, Part 2

whose score is maximum is chosen. If $S' > S$, then the entry with $D = (O', S')$ is left as is in the hash map. On the other hand, if $S' < S$, then the entry is overwritten with $D = (O, S)$.

For the "Bribery and Corruption" example discussed below, we see in Figure 2.11 that Concept Match V2 properly associates the document, "The Business Profile Analysis" (id = 38), with the correct option, "b) Preparing the business profile". The printout of the conceptDocs hash map shows the correct key/value

```
Bribery and Corruption 17

Bribery and Corruption

in proving corrupt payments, the fraud examination often begins with which of the following?

a) Interviewing the target
b) Preparing the business profile
c) Preparing the vertical analysis
d) Interviewing of the co-conspirator

Doc results for: Interviewing the target
** no docs returned **

Doc results for: Preparing the business profile
1. The Business Profile  Analysis(38)  1.2844884
2. Sources of Information for the Business Profile(45)  1.2844884
3. Business Diversions(74)  0.28824288
4. Diverting Business to Vendors(3)  0.2305943
5. WHAT IS THE FINANCIAL CONDITION OF THE BUSINESS?(43)  0.2305943
6. BUSINESS REPORTING COMPANIES(49)  0.2305943
7. HOW IS THE BUSINESS ORGANIZED, LEGALLY AND STRUCTURALLY?(39)  0.20177001
8. PRINCIPALS, EMPLOYEES, AND RECORDS OF SUSPECT BUSINESS(46)  0.20177001

Doc results for: Preparing the vertical analysis
1. The Business Profile  Analysis(38)  0.41790554

Doc results for: Interviewing of the co-conspirator
** no docs returned **

conceptDocs: {49=1, 3=1, 38=2, 39=1, 74=1, 43=1, 45=1, 46=1}
Stem (lower case): in proving corrupt payments, the fraud examination often begins with

Doc results for stem: in proving corrupt payments, the fraud examination often begins with
1. Proving OnBook Payments(50)  0.349766
2. The Corrupt Recipient(31)  0.17838113
3. Methods of Proving Corrupt Payments(37)  0.17700312
4. EXAMINATION FROM THE POINT OF RECEIPT(63)  0.17128104
5. ACCOUNTING BOOKS AND RECORDS(54)  0.14522481
6. The Business Profile  Analysis(38)  0.13609743
7. Loans(25)  0.11591018
8. The Corrupt Payer(32)  0.101687975
9. Bribery(1)  0.1012375
10. Proving Payments in Cash(62)  0.10036731

option selected: 2
Incorrect.  Correct answer: Preparing the business profile
```

Figure 2.10: Concept Match V1: An Example Where A Document is Returned/Ranked for More than One Option

association, 38=1, signifying that document 38 is now associated with option b, as opposed to option c.

2.5.2.1 Concept Match V2 Performance

Next, we look at the performance of the Concept Match Version 2 algorithm on the same collection of questions which fall into the strictly defined definition-question category that were used to test Concept Match V1. Figure 2.12 shows output from

```
Bribery and Corruption 17

Bribery and Corruption

in proving corrupt payments, the fraud examination often begins with which of the following?

a) Interviewing the target
b) Preparing the business profile
c) Preparing the vertical analysis
d) Interviewing of the co-conspirator

Doc results for: Interviewing the target
** no docs returned **

Doc results for: Preparing the business profile
1. The Business Profile  Analysis(38)  1.2844884
2. Sources of Information for the Business Profile(45)  1.2844884
3. Business Diversions(74)  0.28824288
4. Diverting Business to Vendors(3)  0.2305943
5. WHAT IS THE FINANCIAL CONDITION OF THE BUSINESS?(43)  0.2305943
6. BUSINESS REPORTING COMPANIES(49)  0.2305943
7. HOW IS THE BUSINESS ORGANIZED, LEGALLY AND STRUCTURALLY?(39)  0.20177001
8. PRINCIPALS, EMPLOYEES, AND RECORDS OF SUSPECT BUSINESS(46)  0.20177001

Doc results for: Preparing the vertical analysis
1. The Business Profile  Analysis(38)  0.41790554

Doc results for: Interviewing of the co-conspirator
** no docs returned **

optionsDocs: {49=1, 3=1, 38=1, 39=1, 74=1, 43=1, 45=1, 46=1}

Stem (lower case): in proving corrupt payments, the fraud examination often begins with

Doc results for stem: in proving corrupt payments, the fraud examination often begins with
1. Proving OnBook Payments(50)  0.349766
2. The Corrupt Recipient(31)  0.17838113
3. Methods of Proving Corrupt Payments(37)  0.17700312
4. EXAMINATION FROM THE POINT OF RECEIPT(63)  0.17128104
5. ACCOUNTING BOOKS AND RECORDS(54)  0.14522481
6. The Business Profile  Analysis(38)  0.13609743
7. Loans(25)  0.11591018
8. The Corrupt Payer(32)  0.101687975
9. Bribery(1)  0.1012375
10. Proving Payments in Cash(62)  0.10036731

Options doc with best match: The Business Profile  Analysis(38)
Option selected: b) Preparing the business profile
Correct!
```

Figure 2.11: Concept Match V2: Addressing the Problem of Multiple Options for a Document

the algorithm tester shows this algorithm correctly answered 101 questions out of the collection of 150 definition questions, or 67.3%, indicating a dramatic improvement in performance over that for Concept Match V1. We also see that this algorithm has a much lower population of unanswered questions, (3 instead of 46), as a result of the fallback query measure incorporated into V2.

Figure 2.13 shows the results of a hypothesis test determining whether we can conclude a statistically significant improvement in accuracy as a result of the

```
Total questions answered: 150
Number question correctly answered: 101(0.6733333333333333)
number of questions where option selected is -1 (no selection): 3
```

Figure 2.12: Performance of Concept Match V2 on Definition Questions

Concept Match V2 vs. Max Frequency

H0: Agent accuracy = 48.0%
H1: Agent accuracy > 48.0%

E(Question Score):	0.48
Var(Question Score):	0.2496
Number of Exam Questions:	150
E(Exam Score):	72
Var(Exam Score):	37.44
Std(Exam Score):	6.11882342
Observed Exam Score:	101
Z-score for Observed Exam Score	4.73947327
Z-score for p-value of 1%	2.325

Conclusion: Reject H0 in favor of H1 at the 99% confidence level (p-value < 0.01)

Figure 2.13: Concept Match V2 vs. Max Frequency Hypothesis Test on Definition Questions

Concept Match V2 algorithm over that employed by Version 1 of the CFE agent, the Max Frequency algorithm, whose accuracy rate was 48.0%. This analysis shows that we can, in fact, draw the conclusion that Concept Match V2 offers a statistically significant improvement in performance at the 99% confidence level.

2.5.3 Concept Match Version 3

Concept Match V3 attempts to build on Concept Match V2 by leveraging a behavior that was noticed in the results of the Lucene search results of the stem query. In a plurality of cases, the return sets for the question-stem query were

headlined by a document whose score was head-and-shoulders above the rest of the docs in the return set, sometimes by a factor of three or more. In these cases, it was commonly the case that this first-place document was, in fact, the correct document which contained the answer to the question. So, when we have a document that hereafter will be referred to as a "premier document", the algorithm should focus on finding the option most closely associated with that premier document. Concept Match V3 implements this approach by conducting queries for the options and choosing the option for which the premier document scores highest. If no query-option queries based on the title field yield the premier document, then the algorithm repeats the query-option queries against the contents field of the document collection. If the premier document *still* does not appear in any result set, the algorithm returns -1, representing no selection.

Consider the example in Figure 2.14 and Figure 2.15. The question stem query for this "Criminal Prosecutions for Fraud" question returns a result set in which the "Arraignment" document, (id=12), with a score of 0.4654 outscores the next place document, "Sentencing" (id=45) by a facor of more then 2.5x. The algorithm, therefore, categorizes this document as a premier document, and thus approaches the option selection process by attempting to find the option whose query result ranks this document higher than that for any other option. In this example, we see that for the option queries based on the title field, the result sets are thin and there's no match to the premier document. In this case, the algorithm re-issues the option queries, but this time casts a wider net by going aginst the contents field of each document in the collection. In Figure 2.15, we see that this approach prevails – the result set for the correct answer, option b, the Alford plea, includes the Arraignment document. Further, we see that although this document appears in the result sets for other options' queries, it scores highest for the Alford plea option.

Figure 2.16 shows the "Arraignment" document. It provides a couple of key insights as to the behavior of the algorithm on this particular question. First, we notice that the answer to the question is provided on lines 39 through 42. The brief segment shown here concerning a description of the Alford plea suggests (and, in fact, it turns out to be the case upon further checking) that there is no document

```
Question 22 of 26:
Criminal Prosecutions for Fraud 39

Criminal Prosecutions for Fraud

In certain circumstances, the defendant may be allowed to plead guilty, although continuing to assert his or her innocence. This procedure is called:

a) The Brady plea
b) The Alford plea
c) The Johnson plea
d) The Katz plea

Stem (lower case): in certain circumstances, the defendant may be allowed to plead guilty, although continuing to assert his or her innocence. this procedure

Doc results for stem: in certain circumstances, the defendant may be allowed to plead guilty, although continuing to assert his or her innocence. this procedure
1. Arraignment(12)  0.46549812
2. Sentencing(45)  0.15588728
3. DUPLICITY(25)  0.108737275
4. The Trial Process(33)  0.06501623
5. The Charging Process(8)  0.064099945
6. A MOTION CHALLENGING THE SUFFICIENCY OF THE INDICTMENT(24)  0.06306724
7. The Burden of Proof in Criminal Trials(15)  0.053804673
8. Arrest and Interrogation(6)  0.047518227
9. Prosecutorial Discretion and Plea Bargains(13)  0.04420231
10. Appeal(52)  0.040197868

Doc results for: The Brady plea
1. Prosecutorial Discretion and Plea Bargains(13)  0.7560996
2. Exculpatory Information (Brady Material)(31)  0.7560996

Doc results for: The Alford plea
1. Prosecutorial Discretion and Plea Bargains(13)  0.69746906

Doc results for: The Johnson plea
1. Prosecutorial Discretion and Plea Bargains(13)  0.69746906

Doc results for: The Katz plea
1. Prosecutorial Discretion and Plea Bargains(13)  0.69746906

Stem doc is premier: Arraignment(doc=12 score=0.46549812)
Searching for option whose matching doc is the first stem doc.
Search for options docs based on title field unsuccessful.
Repeating search for options docs using contents field...
```

Figure 2.14: Concept Match V3 Example - Part 1

in the collection specifically dedicated to (and whose title field would be) the Alford plea. Second, a cursory examination of this document reveals that there are number of occurrences of the term, "plea" throughout the document which would explain why this document appears in the result set for not only the Alford plea option, but also for the Brady plea, Johnson plea, and Katz plea options. However, the name, "Alford" is the only one of these names mentioned in this document, and this explains why for the Alford plea option, this document scores significantly higher than for any other option.

```
Doc results for: The Brady plea
1. Exculpatory Information (Brady Material)(31)  0.3185585
2. Arraignment(12)  0.24279846
3. Prosecutorial Discretion and Plea Bargains(13)  0.13765378
4. Disclosures by the Defendant(32)  0.12978123
5. Motion to Suppress Evidence(18)  0.09733592

Doc results for: The Alford plea
1. Arraignment(12)  0.7828563
2. Prosecutorial Discretion and Plea Bargains(13)  0.13765378
3. Disclosures by the Defendant(32)  0.12978123
4. Motion to Suppress Evidence(18)  0.09733592

Doc results for: The Johnson plea
1. Arraignment(12)  0.22012393
2. Prosecutorial Discretion and Plea Bargains(13)  0.12479854
3. Disclosures by the Defendant(32)  0.117661186
4. Motion to Suppress Evidence(18)  0.08824589

Doc results for: The Katz plea
1. Arraignment(12)  0.22012393
2. Prosecutorial Discretion and Plea Bargains(13)  0.12479854
3. Disclosures by the Defendant(32)  0.117661186
4. Motion to Suppress Evidence(18)  0.08824589

Search successful for option whose matching doc is first stem doc: b) The Alford plea
Option selected: b) The Alford plea
Correct!
```

Figure 2.15: Concept Match V3 Example - Part 2

2.5.4 Concept Match V3 Performance

Figure 2.17 shows the performance of the Concept Match V3 algorithm on the 150 definition-type questions of the training set. It shows that V3 improves on V2 slightly, getting 105 question correct compared with V2's 101. However, this improvement is not significant enough to be statistically significant at the 99% level, as the figure 2.18 shows. Nonetheless, the fact that V3 outpaces V2 means that the CFE agent will use V3 over V2 when confronted with a definition-type question, (and, as we'll see later, the agent will use this algorithm on other types of questions as well). And finally, lest we forget, compared with Version 1 of the

```
18
19 Arraignment
20 Once the defendant is formally charged, he is brought before the court to enter a plea. This
21 process is called the arraignment. A defendant named in an indictment, if not already in
22 custody, may be arrested on a warrant. Alternativelymore often in white-collar crime
23 cases  the defendant is summoned to appear before a magistrate at a stated time and place
24 to be arraigned.
25
26 The arraignment must take place in open court, and it consists of reading the indictment or
27 information to the defendant and calling on him to enter a plea. The defendant may plead
28 guilty, not guilty, or nolo contendere. If the defendant pleads guilty, the sentencing phase of the
29 criminal justice process begins. A plea of not guilty sets in motion the adjudicative process as
30 described below. A plea of nolo contendere means the defendant does not contest the charges,
31 without formally admitting or denying them. A defendant may plead nolo only with the
32 consent of the court. If accepted, a nolo plea is the same as a plea of guilty for purposes of
33 punishment, but it cannot be used as a formal admission of guilt. This makes it a favored
34 plea for corporate defendants facing subsequent civil litigation.
35
36 Before the court will accept a guilty plea, it must follow procedures to ensure that the plea is
37 voluntary and accurate; that is, that there is a factual basis for the plea. This usually means
38 that the defendant must admit to committing acts that satisfy each element of the offense. In
39 some circumstances, however, a defendant may be allowed to enter an Alford plea (named
40 after the Supreme Court case that upheld the practice) under which he pleads guilty,
41 Law  Criminal Prosecutions for Fraud
42 2011 Fraud Examiners Manual    2.509
43 although continuing to assert innocence. Such a plea may be made to obtain the benefits of a
44 plea agreement and to avoid potentially more dire consequences, such as the death penalty, if
45 the defendant is convicted after trial. Before the court accepts an Alford  plea, it must
46 determine that there is strong evidence of guilt and that the defendant understands the
47 consequences.
48
```

Figure 2.16: The Arraignment Document

agent, this algorithm extends the gains we achieved with Concept Match V2 that were, themselves, found to be statistically significant relative to Version 1 of the CFE agent.

2.5.5 Concept Match NOT

ConceptMatchV3NOT extends the logic of ConceptMatchV3, but turns it on its head to handle questions of the type, "Which of the following is NOT ...", where for the phrase that follows, all options are true except for one, and of course, the agent must choose that option to correctly respond to the question. An example of

```
Total questions answered: 150
Number question correctly answered: 105(0.7)
number of questions where option selected is -1 (no selection): 4
```

Figure 2.17: Performance of Concept Match V3 on Training Set

Concept Match V3 vs. Concept Match V2

H0: Agent accuracy = 67.3%
H1: Agent accuracy > 67.3%

E(Question Score):	0.673
Var(Question Score):	0.220071
Number of Exam Questions:	150
E(Exam Score):	100.95
Var(Exam Score):	33.01065
Std(Exam Score):	5.74548954
Observed Exam Score:	105
Z-score for Observed Exam Score	0.70490077
Z-score for p-value of 1%	2.325

Conclusion: Accept H0.

Figure 2.18: Concept Match V3 vs. V2 Hypothesis Test on Definition Questions

a question of this type is "Which of the following is NOT a plea a defendant may enter at an arraignment?" Concept Match NOT takes an approach that inverts the over-arching approach of the algorithms we've discussed above. Instead of looking for the option for which there's the greatest affinity between option query result sets and the question stem result set, Concept Match NOT find the option whose result set has the least affinity.

Figure 2.19 and Figure 2.20 show an example of this algorithm at work. First, as we saw in the agent's justification in earlier algorithms, we see the result set for the question stem query, and then those for the option queries. The agent then sets about attempting to map the overlap between the question stem result set and the option result sets by building two hash maps. The first one, the "docOptionScores"

map, associates each document among the option query result sets with the the option for which that document earned the highest score. The algorithm then utilizes this data to construct the second hash map, "optionScoreDocs", which consists of option/document key/value pairs for which the documents are present in both the "docOptionScores" map and in the question stem result set. Using this data structure, the agent makes a selection; specifically, it selects the option whose document has the weakest affinity with the question stem result set. In this case, that's option d, skimming, since this option has no representation in the optionScoreDocs data structure, implying it has no documents that overlap with the result set of the question stem. (Looking closely, we see that whereas all of the other option queries have result sets that are non-empty, the the result set for the skimming option has no documents, and so, it's reasonable that it has the weakest affinity with the question stem.)

2.5.6 Performance of the Concept Match Not on the Training Set - Definition/NOT Questions

Figure 2.21 shows the performance of the Concept Match Not algorithm on Definition/NOT questions in the training set - 8 out of 27 correct, or 29.6%. It is not unreasonable to expect that this algorithm would show weaker performance than its inverted cousin, Concept Match V3, as discovering the negative, as we are attempting to do in the case for this question type, is difficult to do. In fact, we cannot reject the null hypothesis that this algorithm performs any better on Definition/NOT questions than random guessing, as shown in Figure 2.22. Further investigation is required to refine this algorithm or to take a different approach with these types of questions altogether.

2.5.7 Concept Match Version 3 NOTA

ConceptMatchV3NOTA leverages the logic in ConceptMatchV3 for concept matching, and extends it for addressing definition questions in which the last option is "none of the above". Hereafter, we'll refer to such questions as NOTA questions.

The first concern to investigate in developing this algorithm was the frequency with which the "none of the above" option was actually the correct response in

```
Fraudulent Disbursements 2

Fraudulent Disbursements

Which of the following is NOT a form of fraudulent disbursement?

a) Payroll schemes
b) Check tampering
c) Billing schemes
d) Skimming

Stem (lower case):  is not a form of fraudulent disbursement

Doc results for stem:  is not a form of fraudulent disbursement
1. Segregation of Duties(126)  0.40012556
2. Detection of Register Disbursement Schemes(10)  0.3340926
3. Register Disbursement Schemes(1)  0.32943964
4. Overbilling with a Nonaccomplice Vendors Invoices(74)  0.2597543
5. ASSET MISAPPROPRIATION  FRAUDULENT DISBURSEMENTS(0)  0.23733978
6. Check Tampering(15)  0.19778316
7. Periodic Review and Analysis of Payroll(127)  0.1812591
8. Forming a Shell Company(64)  0.17629585
9. Billing Schemes(62)  0.17272948
10. Check Disbursement Controls(58)  0.1727163

Doc results for: Payroll schemes
1. Detection of Payroll Schemes(116)  2.364177
2. Prevention of Payroll Schemes(125)  2.364177
3. Payroll Fraud(98)  0.94408447
4. Adding the Ghost to the Payroll(100)  0.75526756
5. Independent Payroll Distribution(117)  0.75526756
6. Periodic Review and Analysis of Payroll(127)  0.75526756
7. Indicators of Payroll Fraud(128)  0.75526756
8. Analysis of Deductions from Payroll Checks(123)  0.6608591
9. Billing Schemes(62)  0.5335261
10. PassThrough Schemes(71)  0.5335261

Doc results for: Check tampering
1. Check Tampering(15)  3.441594
2. Concealing Check Tampering Schemes(46)  2.7532752
3. Detection of Check Tampering Schemes(52)  2.7532752
4. Prevention of Check Tampering Schemes(57)  2.7532752
5. Physical Tampering Prevention(60)  0.7951444
6. Obtaining the Check(17)  0.7268665
7. Converting the Check(29)  0.7268665
8. To Whom Is the Check Made Payable?(20)  0.5814932
9. Converting the Stolen Check(36)  0.5814932
10. Check Disbursement Controls(58)  0.5814932
```

Figure 2.19: Concept Match Not Example - Part 1

NOTA questions. In order to determine this, we developed a trivial algorithm, called the None Of the Above Algorithm, which simply selects the last option, that is, the "none of the above" option, always. Then, we ran this algorithm on all 162 NOTA questions in the training set. The ratio of the correctly answered questions for this algorithm gave us our answer. Figure 2.23 shows the performance results. We see that the "none of the above" option is under-represented as a correct answer, serving as the correct answer only 7.4% of the time. Whereas we'd expect that it should be the correct answer 25% of the time, it is, in fact, drastically under-represented as the correct answer. This served as the inspiration for the simplistic but strategic

```
Doc results for: Billing schemes
1. Billing Schemes(62)  3.3741188
2. Detection of Billing Schemes(81)  2.699295
3. Prevention of Billing Schemes(89)  2.699295
4. PassThrough Schemes(71)  0.46728876
5. PayandReturn Schemes(73)  0.46728876
6. Commission Schemes(112)  0.46728876
7. Register Disbursement Schemes(1)  0.373831
8. Concealing Register Disbursement Schemes(7)  0.373831
9. Detection of Register Disbursement Schemes(10)  0.373831
10. Prevention of Register Disbursement Schemes(14)  0.373831

Doc results for: Skimming
** no docs returned **

docOptionScores: {128=option: 0 score: 0.7552675604820251, 1=option: 2 score: 0.37383100390434265,

optionScoreDocs: {0=doc=127 score=0.1812591, 1=doc=15 score=0.19778316, 2=doc=10 score=0.3340926}

Not all options present in optionScoreDocs.  Picking a missing option.

missing option selected: 3

Option selected: d) Skimming
Correct!
```

Figure 2.20: Concept Match Not Example - Part 2

```
Total questions answered: 27
Number question correctly answered: 8(0.2962962962962963)
number of questions where option selected is -1 (no selection): 0
```

Figure 2.21: Performance of Concept Match Not on Training Set - Definition/NOT Questions

approach of this algorithm - simply remove the "none of the above option" as one of the possible options and select from the remaining three options (using the logic of Concept Match V3).

Figure 2.24 shows an example of the execution of this algorithm. The agent collects the result set for the question stem query, notices that this question is a NOTA question and thus, removes the "none of the above" option from its set of options, infers from the score of the "Quiet Rooms" document (id=171) relative

Concept Match Not vs. Random

H0: Agent accuracy = 25%
H1: Agent accuracy > 25%

E(Question Score):	0.25
Var(Question Score):	0.1875
Number of Exam Questions:	27
E(Exam Score):	6.75
Var(Exam Score):	5.0625
Std(Exam Score):	2.25
Observed Exam Score:	8
Z-score for Observed Exam Score	0.55555556
Z-score for p-value of 1%	2.325

Conclusion: Accept H0.

Figure 2.22: Concept Match Not vs. Random Hypothesis Test on Definition/Not Questions

```
Total questions answered: 162
Number question correctly answered: 12(0.07407407407407407)
number of questions where option selected is -1 (no selection): 0
```

Figure 2.23: Performance of NOTA Algorithm on NOTA Questions

to that of the second place document that its a premier document, and picks the option for whose query scores the "Quiet Rooms" document higher than any other option. This leads to the correct selection of a) Quiet Rooms.

2.5.8 Performance of Concept Match NOTA

Figure 2.25 shows the performance of this algorithm on the Definition/NOTA questions at 65.4% accuracy. Figure 2.26 shows a hypothesis test of this algorithm against the Max Frequency algorithm which had the best results in the CFE Agent Version 1. Since Max Frequency showed results of 56.5% on 162 questions, the results of Concept Match NOTA were *just short* of showing statistically significant improvement at the 99% confidence level. However, notice that the improvement

```
Theft of Intellectual Property 5

Theft of Intellectual Property

A special room that is acoustically shielded and radio-frequency shielded so that conversations within the room cannot be monitored outside the room is called:

a) "Quiet" room
b) "Dead" room
c) "Shield" room
d) None of the above

Stem (lower case): a special room that is acoustically shielded and radio-frequency shielded so that conversations within the room cannot be monitored outside the room

Doc results for stem: a special room that is acoustically shielded and radio-frequency shielded so that conversations within the room cannot be monitored outside the room
1. Quiet Rooms(171)  1.2052177
2. SPIKE AND CAVITY MICROPHONES(140)  0.28825757
3. Hotels(80)  0.14467405
4. Warning Signs of Bugging(144)  0.0760033
5. Travel Plans(123)  0.07587285
6. Video Surveillance(152)  0.07587285
7. Electronic Mail and Voicemail(173)  0.061087526
8. Technical Surveillance Countermeasures (TSCM) Survey(180)  0.05942014
9. ABOVE THE CEILING(149)  0.051834494
10. OTHER PLACES TO SEARCH(150)  0.03455633

This is an NONE-OF-THE-ABOVE question.  Eliminating the NONE-OF-THE-ABOVE option for this algorithm...

Doc results for: "Quiet" room
1. Quiet Rooms(171)  1.1508396

Doc results for: "Dead" room
** no docs returned **

Doc results for: "Shield" room
** no docs returned **

Stem doc is premier: Quiet Rooms(doc=171 score=1.2052177)
Searching for option whose matching doc is the first stem doc.
Search successful for option whose matching doc is first stem doc: a
Option selected: a) "Quiet" room
Correct!
```

Figure 2.24: Concept Match NOTA Example

posted by this algorithm *is* significant at the 98% confidence level. Given this, we incorporate this algorithm into the agent as part of its enhanced suite of tools of Version 2.

2.6 CFE Agent Version 2 Results

Figure 2.27 summarizes the performance of the CFE Agent Version 2. It shows that with a 70.0% accuracy rate on definition questions, Concept Match V3 is the preferred algorithm for that question type. And Concept Match NOT is the preferred algorithm for definition/NOT questions, even with the disappointing accuracy rate of only 29.6%. This algorithm also emerges as the favorite for definition/EXCEPT questions ("All of the following are .... EXCEPT ....") as well, with an accuracy rate of 46.7%. This is not surprising as this type of question is

```
Total questions answered: 162
Number question correctly answered: 106(0.654320987654321)
number of questions where option selected is -1 (no selection): 12
```

Figure 2.25: Performance of Concept Match NOTA on Training Set - Definition/NOTA Questions

Concept Match NOTA vs. Max Freq

H0: Agent accuracy = 56.5%
H1: Agent accuracy > 56.5%

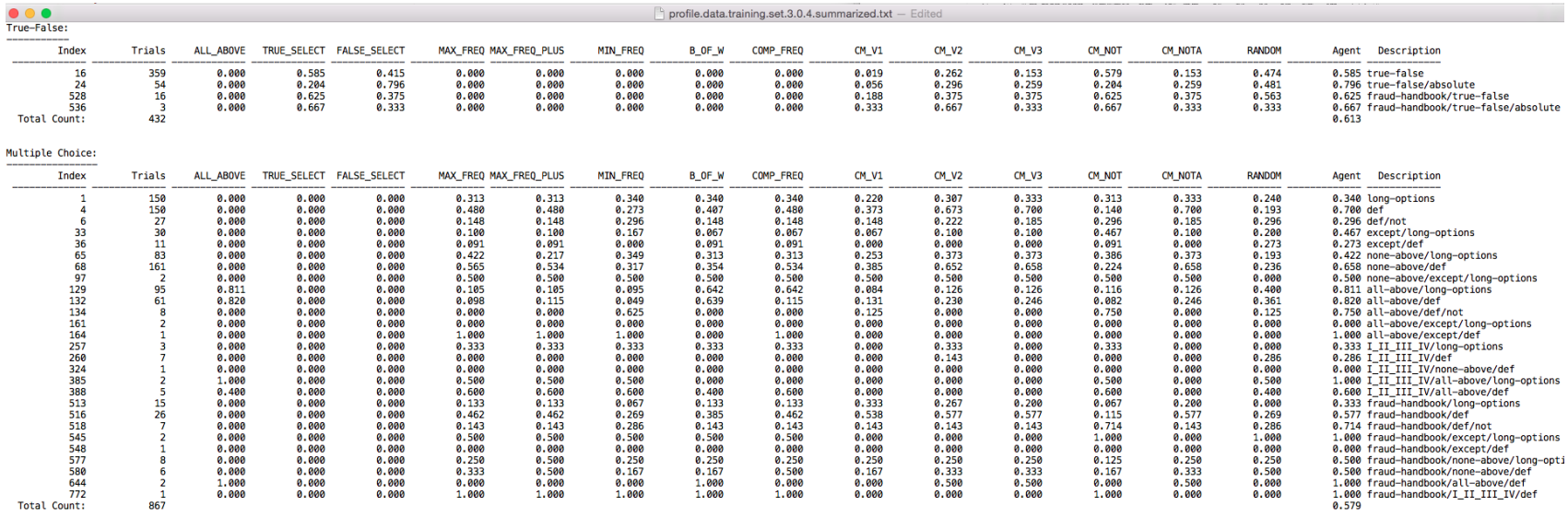
E(Question Score):	0.565
Var(Question Score):	0.245775
Number of Exam Questions:	162
E(Exam Score):	91.53
Var(Exam Score):	39.81555
Std(Exam Score):	6.30995642
Observed Exam Score:	106
Z-score for Observed Exam Score	2.29320126
Z-score for p-value of 1%	2.325
Z-score for p-value of 2%	2.055

Conclusion: Reject H0 in favor of H1 at the 98% confidence level
(0.01 < p-value < 0.02)

Figure 2.26: Concept Match NOTA vs. Max Frequency Hypothesis Test on Definition/NOTA Questions

semantically equivalent to the Definition/NOT question, so it is reasonable for this algorithm to be the top performer for this type as well. Finally, Concept Match NOTA is the preferred algorithm for NOTA questions, with an accuracy of 65.8%.

Figure 2.28 shows the results of a hypothesis test of the performance of the CFE Agent Version 2 on the entire battery of training set questions relative to that of Version 1. The analysis shows the improvement for Version 2 over Version 1 to be statistically significant at the 99% level. This should not be surprising since the question types for which we've targeted our new algorithms, namely Definition,



profile.data.training.set.3.0.4.summarized.txt — Edited

True-False:

Index	Trials	ALL_ABOVE	TRUE_SELECT	FALSE_SELECT	MAX_FREQ	MAX_FREQ_PLUS	MIN_FREQ	B_OF_W	COMP_FREQ	CM_V1	CM_V2	CM_V3	CM_NOT	CM_NOTA	RANDOM	Agent	Description
16	359	0.000	0.585	0.415	0.000	0.000	0.000	0.000	0.000	0.019	0.262	0.153	0.579	0.153	0.474	0.585	true-false
24	54	0.000	0.204	0.796	0.000	0.000	0.000	0.000	0.000	0.056	0.296	0.259	0.204	0.259	0.481	0.796	true-false/absolute
528	16	0.000	0.625	0.375	0.000	0.000	0.000	0.000	0.000	0.188	0.375	0.375	0.625	0.375	0.563	0.625	fraud-handbook/true-false
536	3	0.000	0.667	0.333	0.000	0.000	0.000	0.000	0.000	0.333	0.667	0.333	0.667	0.333	0.333	0.667	fraud-handbook/true-false/absolute
Total Count:	432															0.613	

Multiple Choice:

Index	Trials	ALL_ABOVE	TRUE_SELECT	FALSE_SELECT	MAX_FREQ	MAX_FREQ_PLUS	MIN_FREQ	B_OF_W	COMP_FREQ	CM_V1	CM_V2	CM_V3	CM_NOT	CM_NOTA	RANDOM	Agent	Description
1	150	0.000	0.000	0.000	0.313	0.313	0.340	0.340	0.340	0.220	0.307	0.333	0.313	0.333	0.240	0.340	long-options
4	150	0.000	0.000	0.000	0.480	0.480	0.273	0.407	0.480	0.373	0.673	0.700	0.140	0.700	0.193	0.700	def
6	27	0.000	0.000	0.000	0.148	0.148	0.296	0.148	0.148	0.148	0.222	0.185	0.296	0.185	0.296	0.296	def/not
33	30	0.000	0.000	0.000	0.100	0.100	0.167	0.067	0.067	0.067	0.100	0.100	0.467	0.100	0.200	0.467	except/long-options
36	11	0.000	0.000	0.000	0.091	0.091	0.000	0.091	0.091	0.000	0.000	0.000	0.091	0.000	0.273	0.273	except/def
65	83	0.000	0.000	0.000	0.422	0.217	0.349	0.313	0.313	0.253	0.373	0.373	0.386	0.373	0.193	0.422	none-above/long-options
68	161	0.000	0.000	0.000	0.565	0.534	0.317	0.354	0.534	0.385	0.652	0.658	0.224	0.658	0.236	0.658	none-above/def
97	2	0.000	0.000	0.000	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.000	0.500	none-above/except/long-options
129	95	0.811	0.000	0.000	0.105	0.105	0.095	0.642	0.642	0.084	0.126	0.126	0.116	0.126	0.400	0.811	all-above/long-options
132	61	0.820	0.000	0.000	0.098	0.115	0.049	0.639	0.115	0.131	0.230	0.246	0.082	0.246	0.361	0.820	all-above/def
134	8	0.000	0.000	0.000	0.000	0.000	0.625	0.000	0.000	0.125	0.000	0.000	0.750	0.000	0.125	0.750	all-above/def/not
161	2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	all-above/except/long-options
164	1	0.000	0.000	0.000	1.000	1.000	1.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	all-above/except/def
257	3	0.000	0.000	0.000	0.333	0.333	0.333	0.333	0.333	0.000	0.333	0.000	0.333	0.000	0.000	0.333	I_II_III_IV/long-options
260	7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.143	0.000	0.000	0.000	0.286	0.286	I_II_III_IV/def
324	1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	I_II_III_IV/none-above/def
385	2	1.000	0.000	0.000	0.500	0.500	0.500	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.500	1.000	I_II_III_IV/all-above/long-options
388	5	0.400	0.000	0.000	0.600	0.600	0.600	0.400	0.600	0.000	0.000	0.000	0.600	0.000	0.400	0.600	I_II_III_IV/all-above/def
513	15	0.000	0.000	0.000	0.133	0.133	0.067	0.133	0.133	0.333	0.267	0.200	0.067	0.200	0.000	0.333	fraud-handbook/long-options
516	26	0.000	0.000	0.000	0.462	0.462	0.269	0.385	0.462	0.538	0.577	0.577	0.115	0.577	0.269	0.577	fraud-handbook/def
518	7	0.000	0.000	0.000	0.143	0.143	0.286	0.143	0.143	0.143	0.143	0.143	0.714	0.143	0.286	0.714	fraud-handbook/def/not
545	2	0.000	0.000	0.000	0.500	0.500	0.500	0.500	0.500	0.000	0.000	0.000	1.000	0.000	1.000	1.000	fraud-handbook/except/long-options
548	1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	fraud-handbook/except/def
577	8	0.000	0.000	0.000	0.250	0.500	0.250	0.250	0.250	0.250	0.250	0.250	0.125	0.250	0.250	0.500	fraud-handbook/none-above/long-opti
580	6	0.000	0.000	0.000	0.333	0.500	0.167	0.167	0.500	0.167	0.333	0.333	0.167	0.333	0.500	0.500	fraud-handbook/none-above/def
644	2	1.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.500	0.500	0.000	0.500	0.000	1.000	fraud-handbook/all-above/def
772	1	0.000	0.000	0.000	1.000	1.000	1.000	1.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000	1.000	fraud-handbook/I_II_III_IV/def
Total Count:	867															0.579	

Figure 2.27: Performance of CFE Agent Version 2 on Training Set

CFE Agent Version 2 vs. CFE Agent Version 1
Multiple Choice Questions

H0: Agent accuracy = 47%
H1: Agent accuracy > 47%

E(Question Score):	0.477
Var(Question Score):	0.249471
Number of Exam Questions:	867
E(Exam Score):	413.559
Var(Exam Score):	216.291357
Std(Exam Score):	14.7068473
Observed Exam Score:	502
Z-score for Observed Exam Score	6.01359342
Z-score for p-value of 1%	2.325

Conclusion: Reject H0 in favor of H1 at the 99% confidence level

Figure 2.28: CFE Agent Version 2 vs. CFE Agent Version 1 Hypothesis Test on Multiple Choice Questions

Definition/NOT, and Definition/NOTA , constitute a large portion of the question battery (348 of the 867 questions), as indicated by the Trials column of the table in Figure 2.27.

Running the CFE Agent on the test set of 200 questions yields a score of 119 out of 200, or 59.5%, a dramatic improvement over the 49% of Version 1.

CHAPTER 3
Toward a Passage-Sensitive Agent – Version 3

In this chapter, we attempt to take the idea we introduced in Chapter 4 a step further, that idea being to imbue the agent with the ability to zero in on the relevant passage for each question more narrowly. Whereas in Version 1, the agent attempted to answer questions with a focus on the text corpus that was no more specific than at the level of question section, in Version 2, the agent answered questions after having subdivided the sections into documents, allowing it to zero more accurately on areas of the text that were more specifically related to the question. With Version 3, the agent focuses on passages within each document, narrowing its attention, (hopefully) to even more precisely to the material it needs to generate a correct answer. In order to accomplish this refinement, however, Version 3 does not attempt to answer every type of question, but instead concentrates only on definition-type questions. No attempt is made here, for example to answer NOT or NOTA questions.

The general approach for Version 3 is to apply supervised machine learning (ML) [10], (in conjunction with IR) to the problem of selecting the "correct" passage for each question. By correct passage, we mean the paragraph which most directly relates to the correct response of the question. For questions of the type requiring higher levels of information synthesis drawing from multiple sources, this approach would not be appropriate. This is the reason we focused on definition questions, only, for this version of the agent. The ML model constructed uses logisitic regression for discriminating a correct passage from an incorrect one, based on a prescribed set of features. More specifically, however, the model actually assigns a probability to each passage. Sorting these passages in descending order of probability, the agent selects a prescribed number from the top of this list as most likely candidates, from which it attempts to answer the question by applying algorithms similar to those outlined in Version 1 and 2, albeit to a more refined body of text. By employing ML, however, we must concede that justifications for answers are more opaque, as it becomes more difficult to explain the coefficients that the agent uses in the logit

regression model. Doing so would require a step-by-step, passage-by-question walk through of each of the passages and their categorizations in the training set, which is not practical. We acknowledge the weakness of this approach up front, but attempt to uphold our ideal of answer justification for other reasoning aspects outside of this particular element of the answering algorithm.

In developing this logit regression-based agent, a number of considerations needed to be addressed, including the proper unit/size a passage, the selection of features upon which to base the passage selection model, the selection of questions for the training set and test sets, the manual curation of the training set and test set, as well as, of course, the actual development of the model from the training set and its application to the test set. These issues are discussed in the sections below.

This chapter is laid out as follows: First, we discuss the preparation of the training and test sets, and in so doing explicitly describe how each of the issues mentioned above were handled, in detail. Next, we describe the development of the logit regression model for classifying passages as either relevant or irrelevant. Third, we discuss the application of the model to the test set and discuss its performance. And finally, we discuss the set of algorithms applied in conjunction with logit model and their performance.

3.1 Development of the Training Set

This section discusses the development of the training set for the passage classification model, describing in detail the questions selected for the training set and test set, and the up-front activities that were necessary to develop the model.

3.1.1 Targeted Questions

As mentioned above, Version 3 of the agent is intended to further refine the text body upon which it answers each question. As was the case in Version 2, the primary target for its algorithms are the definition questions. Thus, the clear place to start for the training set was the pool of questions classified as definition questions. Performance reports in Chapter 3 show that for the training set, there are 196 definition questions (profile 4).

However, before moving any further, an even greater in-depth review of the definition questions was in order, as part of the build-up of the training set from this pool of qeustions. At this point in the project, a number of issues arose about some of these questions, causing a need for them to be re-classified out of the definition category into other categories. So, the first task in this effort to focus exclusively only on questions that meet the tight definition for definition questions was to remove these incorrectly classified questions. Causes for the misclassification, including the following: some questions originally classified as defintiion questions were actually of the I, II, III, and IV variety (9 questions) (they're answers were short, and thus slipped through the definition filter), while others were were actually All-of-the-Above variety, but labeled "Any of the Above", (6 questions). There were other questions whose answers weren't derived from the Fraud Examiners Manual, as indicated by the ACFE, but instead from a different text, The Corporate Fraud Handbook (26 questions). Finally, a few more questions in the definition category were actually misclassified because the "NOT" was not correctly picked up in the question stem, and thus, they should have been classified in the definition category (5 questions). After reclassifying these questions, we are left with 150 definition questions (instead of 196) in our training set.

Among these 150 questions, it was found that 133 questions had natural language explanations from which it was easy enough to programmatically extract the page number on which the relevant passage appeared for the question using regular expressions. So, ultimately, the training set was whittled down to 133 definition questions. Using the same approach, 16 questions were selected from among the 200 questions in the test set. Certainly, it would have been desirable to have more questions in the test set, but it does reflect a selection percentage, 16 / 200 = 8% that is roughly similar to the selection percentage from the training set, 133 / 1300 = 10%.

3.1.2 Passage Training Set

Now that the definition of the training set and the test set have been identified, we set about the task of developing the training set of passages upon which the

passage classification model is based. The approach for accomplishing this can be described as follows:

1. Determine the proper unit for breaking up the documents into passages. The choice here was to simply break up the passages along the lines of paragraphs. The rationale for this was twofold. First, paragraphs generally can be thought to contain a collection of units of meaning that make up a single larger unit, so there's a semantic rationale. Second, the practical reason – it's relatively simple to break up text along the lines of paragraphs.

2. Determine the relevant passage for each question, manually. This process, of course, required simply looking through the CFE Manual to find the right paragraph. As mentioned above, using the page numbers programmatically extracted from the explanations for the questions was a huge help, here. Still, this step constituted the most tedious part of the process.

3. Use Lucene to identify the relevant documents for each question. This step utilized Lucene in the same way it was employed for Version 2. That is, elements of the stem were isolated and used as the basis for the formation of a query which was then executed against the document collection for the applicable question section. The 20 top-scoring documents were selected as the ones assumed to provide adequate coverage of the relevant passage. That is, it was presumed likely that one of the top 20 documents contained the relevant passage determined in the last step. The decision to use the number, 20, was arbitrary. However, this number proved to be sufficient in all but 8 out of the 133 training set questions.

4. Create the training set of passages. This was accomplished by extracting each passage for each of the documents from the IR step, and correctly labeling it as relevant/irrelevant. Note that for each question, the assumption was made that exactly one passage was relevant. Again, as mentioned above, for all but 8 questions in the training set, there was one passage marked relevant. All others were marked as irrelevant. For the 133 definition questions of the training set, we generated 9419 passage records from the related documents.

5. Identify and determine the features for each passage. There were a number of features used here as inputs to the logit regression model. Not surprisingly, we used the document search rank. The higher the rank of the containing document, the more likely it would seem that the passage would be the relevant passage for the question. Two other features were also used: the number of distinct words in common between the question stem and the passage, and the length of the longest common sequence of words between the question stem and the passage.

Note that the process of identifying the correct passage and of generating the passages was also applied to the test set. For the 16 questions in the test set, we generated 971 passage records from the related documents.

3.2 Development of the Passage Classification Model

Fig 3.1 shows the output of the program written in R [11] for constructing the passage selection classification model. As mentioned above, this model uses logistic regression to arrive at the constant coefficients shown in the figure. Some clarification of the names of the input and output variables is needed here: dr is the lucene document rank variable, nwic is the variable containing the number of words in common between the passage and the question stem, and llcs is the variable giving the length of the longest sequence of words in common. The output variable, icp, (short for is-correct-passage), is the variable indicating whether the passage is the relevant passage, 1 if it is the correct passage, 0 otherwise. All of the input variables show a significant relationship with the output variable as evidened by their respective z-scores, all of which indicate significance at the 99% level of confidence.

As for the coefficients, themselves, the coefficient for the document rank variable, has a negative sign. This indicates, not surprisingly, a negative relationship with the output variable. That is, the lower the value of document rank variable, (i.e., the closer it is to the value of 1), the higher the likelihood the passage is assigned the value 1, as opposed to 0. The other two variables, nwic and llcs, both have

```
> # build logistic regression model for passages based on training set.
> glm.fit = glm(icp~dr+nwic+llcs,data=passages.train,family=binomial)

> summary(glm.fit)

Call:
glm(formula = icp ~ dr + nwic + llcs, family = binomial, data = passages.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.6811  -0.1076  -0.0421  -0.0136   4.7312

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.10782    0.29280 -14.029  < 2e-16 ***
dr          -0.40287    0.05425  -7.426 1.12e-13 ***
nwic         0.22068    0.04908   4.497 6.91e-06 ***
llcs         0.48648    0.07650   6.359 2.03e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1328.88  on 9418  degrees of freedom
Residual deviance:  713.87  on 9415  degrees of freedom
AIC: 721.87

Number of Fisher Scoring iterations: 10
```

Figure 3.1: Passage Classification Model Summary

positive coefficients, indicating a positive relationship with the output variables, as one would expect.

3.3 Application to the Test Set

The application of the classification model to the passage test set yields favorable results. Figure 3.2 shows an extract of the data, where each record corresponds to a passage (whose text is not shown). The fields show the question id to which the passage applies, the input variables, including dr, nwic, and llcs, and the output

variable, ics. Finally, the field on the far right shows the probability, according to the model, that the passage is considered relevant. In this case, we're not so much interested in whether this value is greater than 0.5, as we typically are in a logistic model, but instead, the records for which this value is highest among all passages for the question. This figure shows the top passages according to probability ranked in order of decreasing probability for each question. The reader will notice that with the exception of the first question id, the record assigned highest probability for the other questions are, the correct passages for those questions. In fact, the classification model correctly assigns the highest probability to the correct passage for 12 of the 16 questions (that is, it shows a precision of 75%). And it omits the correct passage from its top 7 passages once for the 16 test questions, as evidenced in Figure 3.3. In other words, its recall is 93% using a capture size of 7.

3.4 Answer Processing Algorithms for Version 3

The answer processing algorithms discussed in this section leverage the top seven passages extracted using the passage classification logit regression model. Both of these algorithms are relatively straight-forward in nature and utilize the refined text body in a significant way.

3.4.1 MLPassage1

MLPassage1 essentially uses the same bag-of-words-type algorithm that we've used before in Version 1 of the agent. That is, this algorithm calculates the geometric mean frequency of the words in each answer option within the text body for the question. Except this time, of course, the text body is much more refined to the question. We see an example of MLPassage1 in Figure 3.4, where this algorithm successfully selects the correct answer. We also see its justification for its selection by virtue of the computations of the geometric mean and its selection of the option with the highest value. However, we are also reminded in this justification that the text body on which it is performing these calculation results from a logistic regression model whose justification is, at best, opaque.

```
> print(passages.best[c("rid","qid","dr","nwic","llcs","icp","prob")])
    rid                                qid dr nwic llcs icp        prob
1     1          Bribery and Corruption 17  1    3    2   0 0.053367653
2     2          Bribery and Corruption 17  2    4    2   0 0.044877673
3     3          Bribery and Corruption 17  3    3    3   0 0.039355735
4     7          Bribery and Corruption 17  5    5    2   0 0.017194378
5     5          Bribery and Corruption 17  4    1    2   0 0.010711565
6     4          Bribery and Corruption 17  4    3    1   0 0.010243744
7     6          Bribery and Corruption 17  4    0    2   0 0.008608689
8    46                  Consumer Fraud 29  1    4    1   1 0.041427076
9    49                  Consumer Fraud 29  3    3    2   0 0.024567829
10   48                  Consumer Fraud 29  3    2    2   0 0.019799156
11   47                  Consumer Fraud 29  2    2    1   0 0.018239996
12   52                  Consumer Fraud 29  4    3    2   0 0.016556015
13   50                  Consumer Fraud 29  3    0    2   0 0.012824772
14   54                  Consumer Fraud 29  5    2    2   0 0.008943407
15   83  Contract and Procurement Fraud 14  1   16    6   1 0.874241722
16   84  Contract and Procurement Fraud 14  1    3    2   0 0.053367653
17   85  Contract and Procurement Fraud 14  2    3    2   0 0.036313572
18   86  Contract and Procurement Fraud 14  3    3    2   0 0.024567829
19   87  Contract and Procurement Fraud 14  4    2    2   0 0.013321216
20   89  Contract and Procurement Fraud 14  5    2    1   0 0.005517293
21   88  Contract and Procurement Fraud 14  4    0    1   0 0.005310114
22  115        Financial Statement Fraud 9  1    4    3   1 0.102610451
23  119        Financial Statement Fraud 9  2    0    3   0 0.030646224
24  120        Financial Statement Fraud 9  3    2    2   0 0.019799156
25  118        Financial Statement Fraud 9  2    0    2   0 0.019065963
26  117        Financial Statement Fraud 9  2    1    1   0 0.014681084
27  128        Financial Statement Fraud 9  4    2    2   0 0.013321216
28  130        Financial Statement Fraud 9  4    2    2   0 0.013321216
29  187                Money Laundering 26  1   11    9   1 0.908467037
30  189                Money Laundering 26  1    3    4   0 0.129798077
31  191                Money Laundering 26  2    4    2   0 0.044877673
32  188                Money Laundering 26  1    2    2   0 0.043256767
33  190                Money Laundering 26  1    3    1   0 0.033498366
34  195                Money Laundering 26  3    2    3   0 0.031810302
```

Figure 3.2: Application of the Passage Classification Model to the Test Set

3.4.1.1 MLPassage1 Performance

For the test set of 16 questions, this algorithm answered 8 questions correctly, 50%. Despite the refined approach to selecting passages, this algorithm did not perform up to expectations. Reasons for this are discussed below.

For a number of other questions in the test set, the algorithm failed to select the correct answer, as shown in the Figure 3.5. As shown in the answer justification, the problem here stems from the fact that because there is no smoothing (such as,

```
88  714 Criminal Prosecutions for Fraud 66  1    1    2   0 0.034990656
89  713 Criminal Prosecutions for Fraud 66  1    1    1   0 0.021805697
90  715 Criminal Prosecutions for Fraud 66  1    1    1   0 0.021805697
91  720 Criminal Prosecutions for Fraud 66  2    0    2   0 0.019065963
92  783 Criminal Prosecutions for Fraud 73  2    5    4   0 0.134206944
93  784 Criminal Prosecutions for Fraud 73  2    4    3   0 0.071000672
94  780 Criminal Prosecutions for Fraud 73  1    4    2   0 0.065679627
95  781 Criminal Prosecutions for Fraud 73  1    3    2   0 0.053367653
96  782 Criminal Prosecutions for Fraud 73  1    2    2   0 0.043256767
97  786 Criminal Prosecutions for Fraud 73  4    3    3   0 0.026653209
98  785 Criminal Prosecutions for Fraud 73  3    3    2   1 0.024567829
99  847       Legal Rights of Employees 23  1    9    5   1 0.476963148
100 857       Legal Rights of Employees 23  3    4    4   0 0.076717407
101 854       Legal Rights of Employees 23  2    4    2   0 0.044877673
102 849       Legal Rights of Employees 23  2    3    2   0 0.036313572
103 853       Legal Rights of Employees 23  2    2    2   0 0.029333585
104 851       Legal Rights of Employees 23  2    2    1   0 0.018239996
105 852       Legal Rights of Employees 23  2    2    1   0 0.018239996
106 914       Legal Rights of Employees 43  1    5    5   0 0.273902084
107 916       Legal Rights of Employees 43  2    4    2   0 0.044877673
108 922       Legal Rights of Employees 43  4    4    3   0 0.033017091
109 918       Legal Rights of Employees 43  2    2    2   0 0.029333585
110 915       Legal Rights of Employees 43  1    2    1   0 0.027044315
111 917       Legal Rights of Employees 43  2    2    1   0 0.018239996
112 919       Legal Rights of Employees 43  3    3    1   1 0.015248275

> # print out the number of records for which we have icp (is correct passage) = 1.
> # in this case, if we have all of them, the number should be 16. .... [TRUNCATED]
[1] 15
```

Figure 3.3: Test Set - Count of Correct Passages Retrieved By Classification Model

Laplace smoothing) of word frequencies, those options with at least one word that does not occur in the narrowly focused text body are assigned a score of zero. In the case of the example shown here, the word, "scam" does not appear anywhere in the text body, and thus all options are assigned a score of zero since all of them include the word "scam". Other problems are made apparent in the test set, as well. MLPassage1 also includes no stemming of the words in order to account for word transformations between the options and the text body.

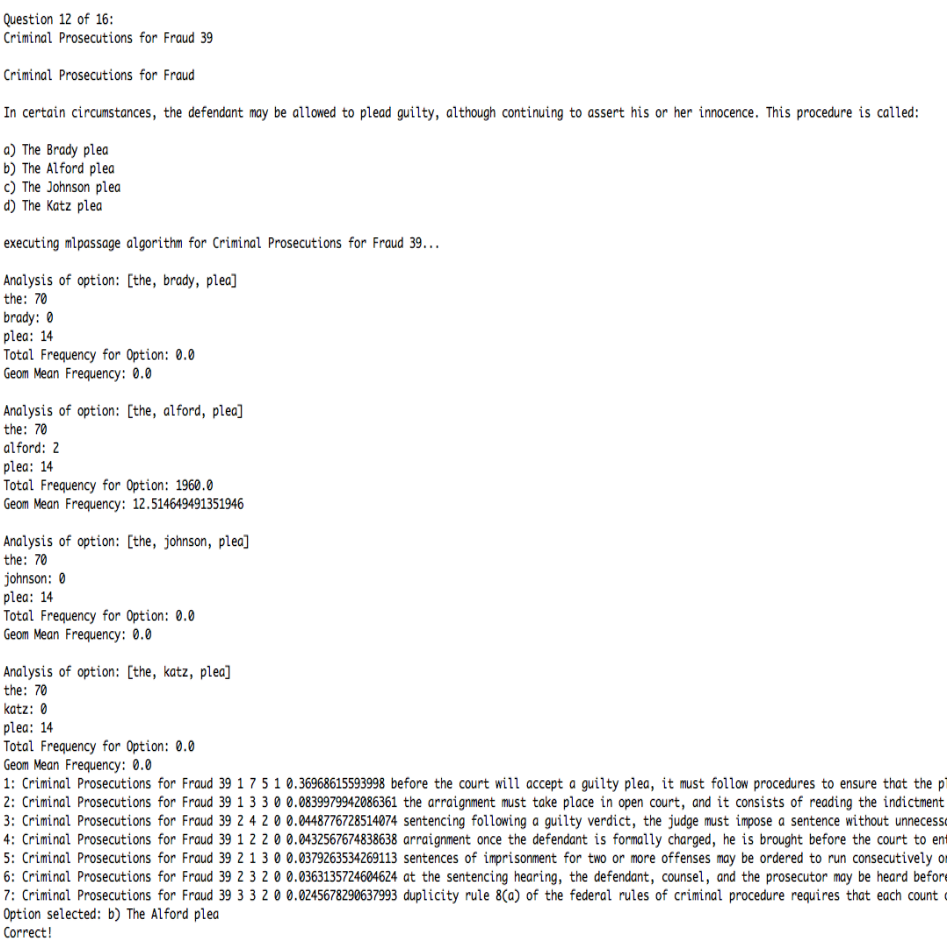


Figure 3.4: MLPassage1 Algorithm - Test Case 1

Finally, we must remember that only one of the passages is actually considered the correct passage among the seven passages extracted by the passage classification model. The other six are, in fact, not relevant. To the extent MLPassage1 includes all of the passages in its text body for each question, these other six may have a deleterious effect on the performance of the algorithm. The next algorithm takes this phenomenon into account by using only the first passage among the seven (that is, the passage with highest probability of relevance).

```
Question 2 of 16:
Consumer Fraud 29

Consumer Fraud

"Boiler rooms," "fronters," "closers," and "verifiers" are all terms associated with which of the following?

a) Real estate scams
b) Telemarketing scams
c) Advance fee frauds
d) Internet fraud

executing mlpassage algorithm for Consumer Fraud 29...

Analysis of option: [real, estate, scams]
real: 1
estate: 1
scams: 0
Total Frequency for Option: 0.0
Geom Mean Frequency: 0.0

Analysis of option: [telemarketing, scams]
telemarketing: 3
scams: 0
Total Frequency for Option: 0.0
Geom Mean Frequency: 0.0

Analysis of option: [advance, fee, frauds]
advance: 0
fee: 0
frauds: 0
Total Frequency for Option: 0.0
Geom Mean Frequency: 0.0

Analysis of option: [internet, fraud]
internet: 0
fraud: 4
Total Frequency for Option: 0.0
Geom Mean Frequency: 0.0
1: Consumer Fraud 29 1 4 1 1 0.0414270763706494 boiler room staff work in a boiler room is shared by fronters, closer
2: Consumer Fraud 29 3 3 2 0 0.0245678290637993 the salespeople in boiler rooms are sometimes as desperate as their v
3: Consumer Fraud 29 3 2 2 0 0.0197991561758464 staff exploitation the customers of fraudulent telemarketing operatio
4: Consumer Fraud 29 2 2 1 0 0.0182399960989958 closers the closer is a veteran. fronters pass an interested caller t
5: Consumer Fraud 29 4 3 2 0 0.0165560148868311 900 numbers/800 numbers/international calls 900 numbers are usually a
6: Consumer Fraud 29 3 0 2 0 0.0128247723027334 naturally, there are no social security or payroll taxes deducted fro
7: Consumer Fraud 29 5 2 2 0 0.0089434073113793 3 other common hustles occur in the privacy of ones own home or throug
Option selected: a) Real estate scams
Incorrect.  Correct answer: Telemarketing scams

Explanation: Terms in the telemarketer??s vocabulary include banging, or nailing, the customer (i.e., closing the dea
Manual page: 1.1709
```

Figure 3.5: MLPassage1 Algorithm - Test Case 2

3.4.2 MLPassage2

MLPassage2 addresses two of the problems cited with MLPassage1, namely the stemming problem and the passage selection problem. An implementation of the Porter Stemmer algorithm was incorporated into this algorithm to address the first issue. As for the second, this algorithm includes only the first text passage out of the seven. As noted above, the passge classification algorithm has good precision - among the 16 questions, it accurately ranks the correct passage at the top of the list of passages for 12 of them. So, reducing the text body to only the first passage

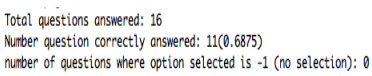


Figure 3.6: MLPassage2 Algorithm Performance

seemed like a prudent choice. As a result of these modifications, the MLPassage2 algorithm shows improved performance of 11 of 16 correct (68.75%), as shown in Figure 3.6.

The performance of MLPassage2 is comparable to that seen in Version 2 of the agent on definition questions. There is more room for enhancements and modifications to MLPassage2, however, which should result in better performance, still, upon further research.

REFERENCES

[1] Dragomir R Radev, John Prager, and Valerie Samn. Ranking suspected answers to natural language questions using predictive annotation. In *Proceedings of the sixth conference on Applied natural language processing*, pages 150–157. Association for Computational Linguistics, 2000.

[2] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[3] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[4] Ronald Brachman and Hector Levesque. *Knowledge representation and reasoning*. Elsevier, 2004.

[5] S. Bringsjord, S. & Schimanski. What is Artificial Intelligence? Psychometric AI as an Answer. *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 03)*, pages 887–893, 2004.

[6] S. Bringsjord and B. Schimanski. What is Artificial Intelligence? Psychometric AI as an Answer. In *Proceedings of the 18th International Joint Conferenc e on Artificial Intelligence (IJCAI–03)*, pages 887–893, San Francisco, CA, 2003. Morgan Kaufmann.

[7] Harry Markopolos. *No One Would Listen*. John Wiley Sons, 2010.

[8] Wikipedia. Standard boolean model — wikipedia, the free encyclopedia, 2015. [Online; accessed 1-February-2016].

[9] Michael McCandless, Erik Hatcher, and Otis Gospodnetic. *Lucene in Action, Second Edition: Covers Apache Lucene 3.0*. Manning Publications Co., Greenwich, CT, USA, 2010.

[10] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[11] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.