AFRL-RI-RS-TR-2014-206

TOWARD AUTOMATED INTERNATIONAL LAW COMPLIANCE MONITORING (TAILCM)

LEIDOS, INC

JULY 2014

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was reviewed by IARPA and cleared for public release by ODNI Public Affairs, and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2014-206 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/ S /

ALEKSEY V. PANASYUK
Work Unit Manager

/ S /

MICHAEL J. WESSING
Deputy Chief, Information Intelligence
Systems and Analysis Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
JUL 2014	FINAL TECHNICAL REPORT	MAR 2013 – APR 2014

4. TITLE AND SUBTITLE

TOWARD AUTOMATED INTERNATIONAL LAW COMPLIANCE MONITORING (TAILCM)

5a. CONTRACT NUMBER
FA8750-13-C-0085

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
Other (SAF)

6. AUTHOR(S)

Leora Morgenstern

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Leidos, Inc
11951 Freedom Drive
Reston, VA 20190

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIED
525 Brooks Road
Rome NY 13441-4505

IARPA Gate 5
1000 Colonial Farm Road
McLean, VA 22101

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2014-206

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report was reviewed by IARPA and cleared for public release by ODNI Public Affairs, and is available to the general public, including foreign nationals. Date Cleared: 21 Jul 2014.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The IARPA seedling TAILCM (Toward Automated International Law Compliance Monitoring) was developed to explore the feasibility of translating regulatory text to formal executable rules that could input into a standard rule engine. This report presents the research performed during the year-long seedling. The four major research areas are expanding bulleted regulatory text, categorizing regulatory documents by discourse structure, ontology extraction and merging, and rule template slot filling.

15. SUBJECT TERMS

Ontology Extraction, Regulatory Compliance Assistant, Extraction of executable rules from regulatory text

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**ALEKSEY PANASYUK**
U	U	U	SAR	77	19b. TELEPHONE NUMBER (*Include area code*) **(315) 330-3976**

TABLE OF CONTENTS

List of Figures

List of Tables

1. Summary

The IARPA seedling TAILCM (Toward Automated International Law Compliance Monitoring) was developed to explore the feasibility of translating regulatory text to formal executable rules that could input into a standard rule engine. This report presents the research performed during the year-long seedling. The four major research areas are expanding bulleted regulatory text, categorizing regulatory documents by discourse structure, ontology extraction and merging, and rule template slot filling.

2. Introduction

The IARPA seedling TAILCM (Toward Automated International Law Compliance Monitoring) was developed to explore the feasibility of translating regulatory text --- for example, laws from the U.S. Code of Federal Regulations, or European Union regulations --- to formal executable rules in an established standard such as RIF (Boley and Kifer, 2013; Morgenstern et al., 2013) or Rulelog (Anderson et al., 2013). Such formal executable rules can be input into a standard rule engine (e.g., Drools) and could be a central part of an automated system to determine compliance with a set of regulations. At an IARPA-led workshop held in May 2012, the consensus among participants was that the translation of regulatory text to formal executable rules was the hardest and most central challenge to the development of an automated system to monitor law compliance.

Leidos (formerly SAIC) proposed simplifying the task by breaking it into two conceptually simpler and more manageable subtasks: first, extracting from the regulatory text an intermediate representation, specifically an ontology consisting of concepts and the relations between these concepts; second, using this intermediate representation to construct formal executable rules.

We chose a subset of U.S. financial regulations --- insider trading and anti-money laundering regulations --- as the domain for this seedling, although the research performed is expected to carry across many different domains. No domain-specific assumptions were made during the execution of this research. The choice of domain was made due to the ease of obtaining research materials.

The original architecture for this seedling and the final architecture are shown in Figures 1 and 2 respectively. The architecture was evolving due to unforeseen difficulties associated with parsing the regulations, gold standard produced by subject experts, and other difficulties described in section 3 of this report. Note the addition of components for expanding bulleted text, classification, semantic parsing, slot filler extraction, and added knowledge bases of rule templates and domain-specific dictionaries.

Most of the technical content in the report is contained in Section 3. Section 3.1 describes the process of corpus collection. Section 3.2 analyzes the differences between traditional domains, such as newswire, that have been used for Natural Language Processing applications, and the regulatory domain. Section 3.3 motivates the research that we performed. Section 3.4 presents our 4 areas of research: expanding bulleted text, categorizing regulatory text by discourse function, extracting and merging ontologies, and rule template slot filling. Section 4 presents detailed results. In Section 5 we discuss lessons learned and possible paths forward.

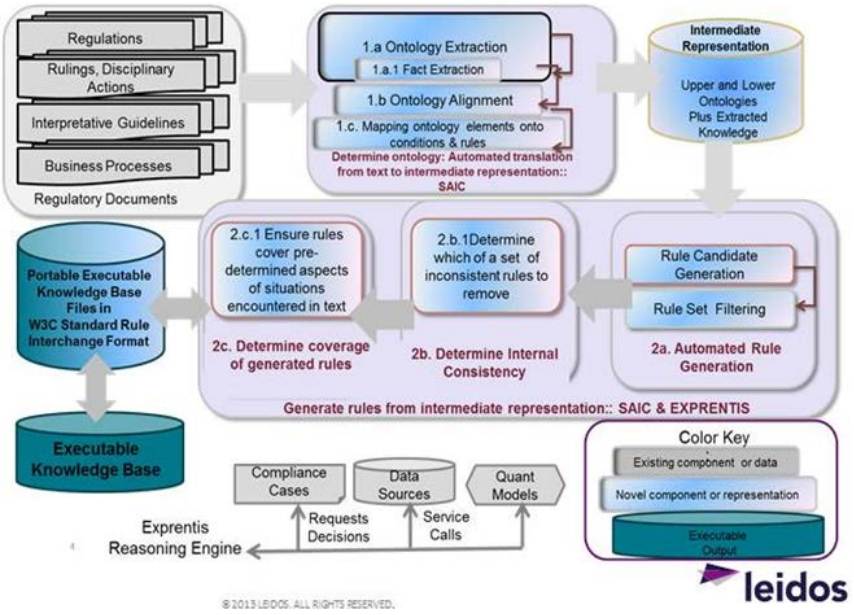


Figure 1: Original Architecture, April 2013

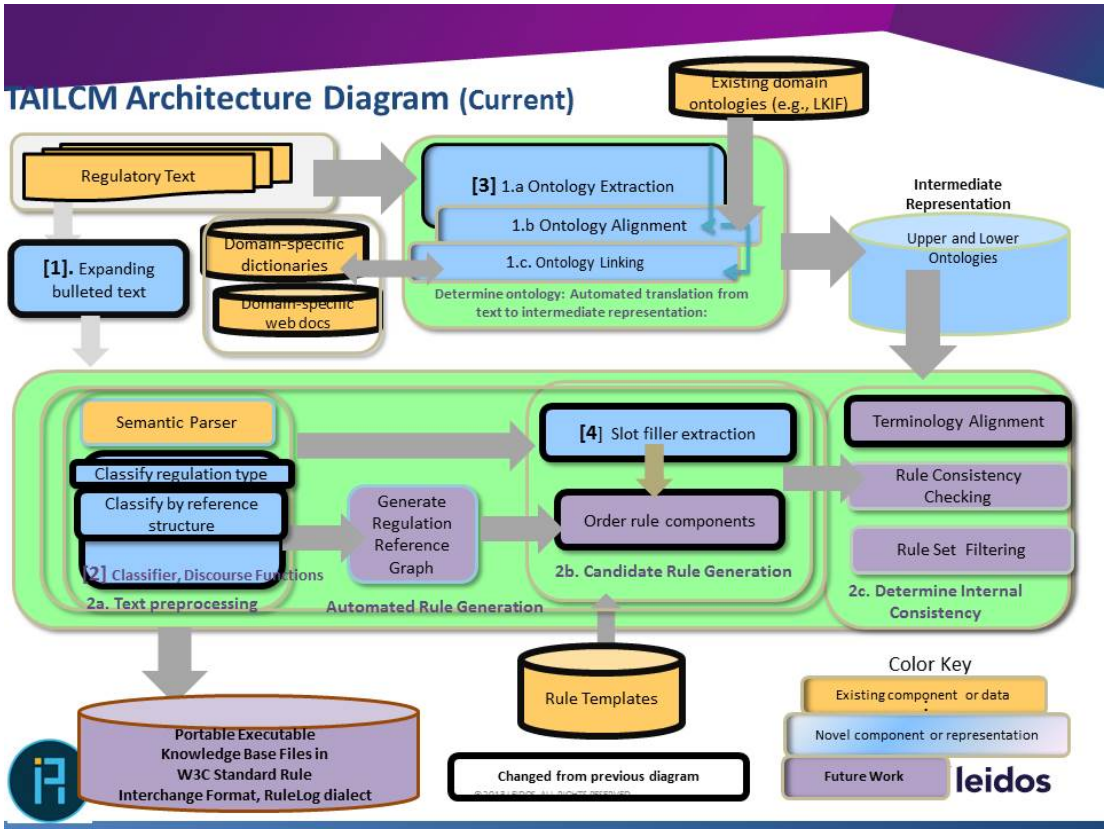


Figure 2: Final Architecture, April 2014

3. Methods, Assumptions, and Procedures

3.1 Corpus Collection

The initial step in this research program was the collection of a corpus of regulations, intended to serve as the training corpus. Although United States law includes a large set of financial regulations dealing with insider training and anti-money laundering, many regulations were not suitable for our research. We aimed to collect regulations which were

- of reasonable length (more than a few paragraphs, but not more than ten pages)
- primarily concerned with financial regulation
- of more than just theoretical interest

This eliminated many candidate regulations. For example, the Securities Exchange Act of 1934 runs to 371 densely typed pages. Likewise, other portions of regulations, but not entire regulations, are suitable: for example, only a minority of the Patriot Act (mostly Title 3) is concerned with anti-money laundering or other type of financial regulation. In addition, in any regulation, it is typically the case that some portions of the regulation are more widely applied than others, and we wished to ensure that we only included regulations which at least at some time had been referenced in a ruling.

These considerations led us to believe that an attempt to translate very long regulations from text to executable rules would be unfocused and unlikely to succeed. Rather, we determined to use commonly cited short regulations, or portions of longer regulations.[1]

In order to find such *regulation units*, we turned to specific rulings from federal agencies and associated organizations. For the domains that were the focus of this seedling, insider trading and anti-money laundering, we turned to the websites of FINRA, the Financial Industry Regulatory Authority (http://www.finra.org/), and FINCEN, the Financial Crimes Enforcement Network (http://www.fincen.gov/). These websites have enforcement sections that contain both summaries and full texts of rulings for thousands of cases. Each *ruling document*, the document that gives a ruling for a specific case decided in court, cites specific regulation portions used to determine judgment in a case. We collected 100 ruling documents, and then collected all regulation units referenced in these documents. The result was a corpus of 250 regulation units. There were some redundancies and inclusions among these documents, yielding approximately 230 distinct regulation units of appropriate length.

[1] The following observations may clarify the notion of a "short" regulation or regulation portion: Examples of short regulations are FINRA 5330 (1 page) and MSRB Rule G-37(8 pages). Examples of large regulations are the Securities Act of 1933 (93 pages), at http://www.sec.gov/about/laws/sa33.pdf; the Securities Exchange Act of 1934 (371 pages), at http://www.sec.gov/about/laws/sea34.pdf; and the Patriot Act (132 pages), at http://www.gpo.gov/fdsys/pkg/PLAW-107publ56/pdf/PLAW-107publ56.pdf. These large regulations are divided into many regulation portions, often identiried in the legal literature as sections, especially for regulations that are part of the US Code. Examples of regulation portions are CFR 103.18, at http://www.law.cornell.edu/cfr/text/31/103.18, and 26 USC x6621, at http://www.law.cornell.edu/uscode/text/26/6621. Regulation units are usually under 10 pages in length. For our purposes, a regulation unit will be characterized by a complete and integral bulleting structure. That is, a regulation unit always contains a single bulleted list, generally with multiple levels of nesting, in its entirety. Thus, for example, CFR 103 is not a regulation unit, since it contains many sections, each of which contains an entire list; while CFR 103.18(a) is not a regulation unit because it is itself a bullet of a larger list.

3.2 Differences between regulatory domain and standard NLP domains

Our initial analysis of our corpus of regulation units (CRU) showed that the domain of legal regulations differs in two important respects from domains frequently used in applications of Natural Language Processing (NLP) techniques, such as newswire stories or biomedical texts (LDC, 2009; McClosky et al., 2012). At least four types of differences between text such as news stories and regulatory text have been identified by our research team.

Facts vs. Irrealis:

News stories and biomedical texts generally relate facts; regulatory text generally concerns itself with rules to be followed in specific circumstances. Consider the first paragraph of a news story from the June 7, 2013 New York Times:

> "In the deadliest single episode for international forces in Afghanistan since August, a suicide bomber driving a truck packed with explosives attacked an isolated base staffed by Georgian troops in Helmand Province on Thursday evening, killing seven soldiers, according to Georgian and Afghan officials in Helmand."

In contrast, consider the following excerpt from FINRA 2265:

> "No member shall permit a customer to engage in extended hours trading unless the member has furnished to the customer, individually, in paper or electronic form, a disclosure statement highlighting the risks specific to extended hours trading. In addition, any member that permits customers either to open accounts on-line in which such customer may engage in extended hours trading or to engage in extended hours trading in securities on-line, must post an extended hours trading risk disclosure statement on the member's Web site in a clear and conspicuous manner."

The excerpt from New York Times relates facts about a suicide bombing in Afghanistan such as the fact that a suicide bomber drove a truck into an army base, the fact that this happened in Helmand Province in Afghanistan, and the fact that the bomb killed seven soldiers. The paragraph from FINRA regulation does not say anything about events that have happened but only specifies what a member (broker) can and cannot do in specific circumstances. Text that is not meant to indicate what a particular fact or event holds is said to be in the *irrealis* mood, or more simply *irrealis*. Nearly all regulatory text is in the deontic irrealis mood.

Lack of named entities:

Whether or not it is irrealis, news text typically has many named entities compared to regulatory text. For example, consider the first few paragraphs of a news article from June 3, 2013:

> "President Obama will nominate a slate of three candidates on Tuesday to fill the remaining vacancies on the United States Court of Appeals for the District of Columbia Circuit, a White House official said Monday. The president will name Cornelia T.L. Pillard, a law professor; Patricia Ann Millett, an appellate lawyer; and Robert L. Wilkins, a federal district judge, to fill out the appeals court, which is often described as the second most powerful court in the country because it decides major

cases and often serves as a launching pad for future Supreme Court justices. By making his choices in a group, the president and his strategists are hoping to put pressure on Senate Republicans to confirm them. Mr. Obama is expected to announce the nominations at a Rose Garden ceremony on Tuesday morning, said the White House official, who spoke on the condition of anonymity because the nominations had not been announced."

There are 15 named entities in this short excerpt. In contrast, there is just one named entity in the following rule, NFA Rule 2-2, and that is the NFA (National Futures Association) itself:

"No Member or Associate shall:

- Cheat, defraud or deceive, or attempt to cheat, defraud or deceive, any commodity futures customer;
- Bucket a customer's commodity futures order or engage in a business that is of the nature of a bucket shop;
- Willfully make or cause to be made to a customer a false report, or willfully to enter or cause to be entered for a customer a false record, in or in connection with any commodity futures contract;
- Disseminate, or cause to be disseminated, false or misleading information, or a knowingly inaccurate report, that affects or tends to affect the price of any commodity that is the subject of a commodity futures contract;
- Engage in manipulative acts or practices regarding the price of a commodity futures contract;
- Willfully submit materially false or misleading information to NFA or its agents;
- Embezzle, steal, purloin or knowingly convert any money, securities or other property received from or accruing to a customer, client or pool participant in or in connection with commodity futures contracts."

There are many entities (customer, false report, false record, commodity futures contract, price, commodity, agents, and manipulative acts) but only one named entity. These differences have important implications for the type of research needed to understand regulatory text, as described in Section 3.3.

Greater complexity of text:

Legal text is considerably more complex than newswire stories. Rudolf Flesch developed one of the best-known readability metrics, based on the average sentence length and average number of syllables per word in a piece of text:.

(206.835– [(1.05 * length of the average sentence) + (84.6 * average word length)])

The formula assigns a value of 0 to 100 to most of the text. The higher a text scores, the more readable it is. In Flesch's (1979) analysis, comic books scored 92 on the scale; the New York Daily News (a tabloid) scored 60; the New York Times scored 39; and Harvard Law Review scored 32. The hardest text to understand was that of the Internal Revenue Code, which scored a *negative* 6: literally off the scale because of the text's difficulty.

Sentence length is an issue for parsers which we tried. The Stanford parser, for example, will not parse sentences that are longer than 70 words; the initial part of the excerpted sentence of NFA Rule 2-2 whose initial portion was quoted above was more than double that length. Beyond the length of words and sentences investigated by Flesch, our analysis has shown that the grammatical structure is also considerably more complex than that found in newswire articles. In particular, there are many nested disjunctions and conjunctions that make sentences difficult to comprehend. The nested disjunctions in the NFA Rule 2-2 imply at least 72 forbidden actions for members or associates. Raw text without considering the nested structure will generate a difficult parse tree that will be a burden both for educated humans and for machines.

Complete-information-critical vs. complete-information-desirable tasks:

An important difference between newswire domains and the regulatory domain is the tasks that are associated with the domain. Tasks typically associated with the newswire domain are question answering and summarization. The aim is to answer a factual question (e.g., Question: What was the nationality of the soldiers killed in the June 2013 bombing in Afghanistan? Answer: Georgian) or give a summary of events (e.g. "Seven Georgian soldiers were killed by a suicide truck bomb in Helmand Province."). The task for regulatory text is to specify conditions that can be used to determine compliance and non-compliance for the parties involved.

It is nearly always desirable to have complete information. However, it is not critical for summarizing a news story: summary may still be useful even if it is missing many facts. For determining compliance, however, it is critical to understand the entire regulation. Not understanding a subtle exception that is buried in the regulation can lead to the formulation of an executable rule that does not capture the text. The TAILCM task is thus complete-information-critical (CIC), a much higher standard to achieve than complete-information-desirable (CID), the standard for many tasks associated with newswire domains.

3.3 Implications for TAILCM research: mapping out major challenges

The particular properties of the regulatory domain and of the TAILCM task suggest that it may not be straightforward to apply state-of-the-art Natural Language Processing (NLP) techniques to our task. The last two decades have seen great progress in the development of NLP techniques, and the combined used of NLP, statistical, and Machine Learning (ML) techniques. These combined techniques are currently capable of giving very good performance in specialized domains. (Olive et al., 2011) discuss performance in DARPA's GALE program: evaluations showed an ability to identify "nuggets" of information that sometimes surpassed that of humans (e.g., see Babko-Malaya et al., 2012).

These good results, however, hold for newswire texts, texts that tell a story and have many named entities; and for the task of identifying nuggets, a task that is similar to text summarization. One cannot assume such good results for the regulatory domain and the TAILCM task, and there are good reasons to believe that without development of new methods, one would get significantly worse results. The core software for the nugget identification, summarization, and questioning-answering tasks builds on identifying named entities; using these named entities, either in a standard train-and-test paradigm (Mitchell, 1997), or using semi-supervised learning (Mintz et al., 2009), in order to extract instances of relations; and using these relation instances in order to accomplish the desired task.

It is difficult to replicate this core technology in a domain where there are few named entities. Nor is it clear that this technology will suffice to do the desired task. Extraction of relation instances will not shed light on whether a regulation states a prohibition, permission, or obligation; what the exceptions are of a rule; or what the conditions are of a rule. But being able to extract this information from a regulation is a prerequisite to the ability to construct a formal executable rule that captures that regulation.

An additional challenge that we faced was the inadequacy of state-of-the-art parsers. Parsing is the keystone of any NLP technique. Parsers do best on simple sentences, such as those found in news stories. But even in such situations, performance is mediocre: studies (e.g., Manning, 2011) estimate sentence parsing accuracy at 56%, in contrast with a 97% accuracy rate for part-of-speech recognition. Our initial analysis of a sample of sentences from our corpus on three freely available, widely-used parsers, including Stanford CoreNLP, Enju, and C&C, (http://nlp.stanford.edu/software/corenlp.shtml , http://www.nactem.ac.uk/enju/, http://svn.ask.it.usyd.edu.au/trac/candc/wiki) yielded accuracy levels of less than 30% due to the complexity of the regulatory sentences. Further and more detailed analysis of a larger sample revealed an even lower accuracy level of about 15%.

Ideally, developing a parser that parsed sentences in regulatory documents with high accuracy would have been a strong desideratum. Given available time and resources, this was not a viable option.

The process of selecting a sample of sentences underscored two additional challenges of regulatory documents. First, much of regulatory text is written in bulleted form, which parsers cannot properly handle without preprocessing. Second, much of a regulatory document does not contain regulatory content: it may contain the history of a document, or definitions, or commentary. Automating the identification of actual regulatory sentences would facilitate further analysis.

3.4 Research Performed

We focused on four research areas:

1) Expanding bulleted text to extract sentences from regulatory text

2) Categorizing portions of regulatory documents by discourse function to target special-purpose NLP methods and analysis

3) Ontology extraction and merging with existing ontologies to standardize vocabulary for rules (Midterm task)

4) Rule template slot filling to extract critical components of regulations (Final task)

For each of these areas, we discuss motivation, our technical approach, and its novelty; and give a brief preview of the results. Detailed results are given in Section 4.

3.4.1 Expanding Bulleted Text

3.4.1.1 Motivation

Most of the regulation units in our corpus contained, and in fact were comprised mostly of, bulleted text. Consider, for example, the initial fragment of FINRA Rule 3240, a regulation originating from the United States Financial Industry Regulatory Authority:
"Permissible Lending Arrangements; Conditions

(a) No person associated with a member in any registered capacity may borrow money from or lend money to any customer of such person unless:

> (1) the member has written procedures allowing the borrowing and lending of money between such registered persons and customers of the member;
> (2) the borrowing or lending arrangement meets one of the following conditions:
> > (A) the customer is a member of such person's immediate family;
> > (B) the customer (i) is a financial institution regularly engaged in the business of providing credit, financing, or loans, or other entity or person that regularly arranges or extends credit in the ordinary course of business and (ii) is acting in the course of such business;
> > (C) the customer and the registered person are both registered persons of the same member;"

As humans, bullets help us understand text. We understand that bullet (a) lists several ways in which lending is allowed. We realize that sub-bullet (2) specifies alternative necessary conditions that constrain the relationship between customer and lender. The structure breaks up the text and makes it easier to read, but also demands that we keep the appropriate context.

Parsers, however, cannot handle bulleted text. Bulleted text tends to be long, and parsers don't work large chunks of text. Even for short bulleted lists, parsers have difficulty understanding which piece of context belongs to which bullet. Hence for NLP purposes, it is best to understand bulleted structure as a set of expanded sentences. The rule above, for example, could be expanded into a set of rules, the first two of which would be:

> "No person associated with a member in any registered capacity may borrow money from or lend money to any customer of such person unless the member has written procedures allowing the borrowing and lending of money between such registered persons and customers of the member

No person associated with a member in any registered capacity may borrow money from or lend money to any customer of such person unless the borrowing or lending arrangement meets one of the following conditions: (A) the customer is a member of such person's immediate family …"
A properly performed expansion would thus preserve the original meaning of the text.

3.4.1.2 State of the Art

The difficulty of using bulleted text has been noted by several in the legal informatics field, Wyner (2012) and Wyner and Peters (2011). They, as well as Dell'Orletta et al. (2012) advocate using a process called "splitting": split a piece of text into sentences using punctuation cues. This approach could yield sentences such as:

> "No person associated with a member in any registered capacity may borrow money from or lend money to any customer of such person unless:"
> and
> "(1) the member has written procedures allowing the borrowing and lending of money between such registered persons and customers of the member"

Sentence splitting is problematic because:

- It separates most sentences from their bulleted context, which hampers comprehension and can change the semantics of the sentence.
- Fragmented phrases complicate other research goals such as being able to classify regulation type (obligation, permission, prohibition, penalty, and reparation).
- Many bulleted items are not full sentences, making comprehension even more difficult.

An example of a regulation in which bulleted items are not full sentences comes from FINRA Rule 7440, a fragment of which appears below:

"7440. Recording of Order Information
(a) Procedures …
(4) With respect to each order that is received or executed at its trading department, each Reporting Member shall record an identification of:
(A) each registered person who receives the order directly from a customer;
(B) each registered person who executes the order; and
(C) the department that originated the order if the order is originated by the member and transmitted manually to another department."

Bullets (A), (B), and (C) are all sentence fragments, and make sense only relative to the text belonging to their parent bullet (4). We can best understand the entirety of bullet (4) by expanding each bullet item by distributing parent preambles over children bullets:
"With respect to each order that is received ... each Reporting Member shall record an identification of each registered person who receives an order directly from a customer

AND

With respect to each order that is received ... each Reporting Member shall record an identification of each registered person who executes the order

AND

With respect to each order that is received ... each Reporting Member shall record an identification of the department that originated the order if the order is originated by the member and transmitted manually to another department."

We have found that this approach is valid not only for simple bulleted structures, but for complex nested structures, as in FINRA Rule 3240, above.

3.4.1.3 Working with HTML

This research is the first attempt (as far as we know) to specify and implement a method for expanding bulleted text in a way that preserves meaning in sentences. We use the HTLM version of the regulation in order to analyze the structure of the HTML tags for helping determine the bulleted structure. We then, build a tree structure representing the list, and traverse the tree to obtain the distributed text.

For properly extracting text we need to extract the formal bulleted structure from available text. Second, we need to build a tree structure that supports expansion and distribution of parent preambles over child bullets for arbitrary levels of nesting. Bulleting in regulations is often 4, 5, or deeper nested levels.

Humans are proficient at extracting bulleted structure from text, at least until the level of nested bullets becomes too large. As we skim a text, we notice the layout of the page, knowing that the relative indentation of paragraphs gives us important information about the hierarchy of bullet types. The more indented a paragraph is, the deeper its bulleting level. Bullets with same level of indentation are connected to the same higher level parent. As we skim a page, we build a mental model of this hierarchy.

The ability to analyze white space, related to indentation of paragraphs, is not a strength of computer systems, especially given the many formats in which regulations may appear. Modeling how humans approach the comprehension of bulleted text did not seem promising. Our first effort was to consider raw text files. We found that it was difficult to recognize bullets with current NLP approaches. For example, the introduction of a new bullet is usually heralded by a new line, but new lines are frequently used for other purposes, such as to start paragraphs. Indeed, it is often difficult in a text file to even see where a paragraph starts; detecting paragraphs can involve laborious counting of spaces. Moreover, sometimes new bullets and new levels of bulleting are introduced without new lines. (See FINRA Rule 3240(a)(2)(B) (i) and (ii)). In addition, there are some regulation portions in text files, such as those originating from the Investment Act of 1940, that have no labeled bullets at all; all bulleting is indicated merely by indentation.

Ideally, regulatory documents would be marked using some special-purpose markup language that would indicate, among other things, bulleted structure. Targeted XML schemas, such as Akoma Ntosa (Barabucci et al., 2009a} are currently being used in the European Union; while the legal rule markup language LegalRuleML (Athan et al., 2013) is a standard that is currently under development by the standards board OASIS. However, US regulatory documents marked up in these languages are not available as far as we know. Rather, we decided to try exploring recovering bulleted structure using simple HTML files. We knew that the presence of utilities like jsoup would at least allow us to detect paragraphs and indentation with greater ease. In addition, HTML tags allow us to get rid of junk text relatively quickly. Moreover, all HTML

files that we saw (that were larger than a few paragraphs) imposed some sort of genuine bulleting structure on the file, rather than just relying on indentation.

We began by collecting HTML versions of all documents in our corpus. Six websites ---- http://www.law.cornell.edu, http://finra.complinet.com, http://www.ecfr.gov, http://www.law.uc.edu, http://www.msrb.org, and http://www.nfa.futures.org --- sufficed to cover our collection. We had hoped that once we had the HTML documents in hand, extracting the bulleted structure would be quite easy, just entailing parsing the document for nested lists and list elements (using the <ol>,, <ul>, and tag pairs), but that was unfortunately not the case. Recovering the bulleted list structure from the files turned out to be a significant challenge. The primary difficulty was that each website had its own conventions about how it represented lists. A particular website might not even be internally consistent and might use different HTML markup conventions for different regulations. Thus, for each website, we needed to examine several regulations in detail, looking over the HTML to determine how the bulleted structure had been generated, and how this was generalized across other documents on that website.

Each website presented a set of unique problems that we needed to solve. For example:

1) http://www.ecfr.gov precedes many types of data by simple paragraph (<p>) tags. This is true for regulatory content as well as title information, links (such as the ``Back to Top" link),

   citations, update dates, and sources. This causes problems when one wants to isolate the content that needs to be extracted in a regulation. What helps to a large degree are the class attributes that are declared for particular paragraphs. However, while there are class attributes for links, for titles, and other such information, there are no class attributes for content. We handle this by first removing all paragraphs with class attribute to isolate the content paragraph and then extracting whatever was left as content.

2) CFR documents often have several nested labels appearing in a single line. For example, from CFR 1010.100:
   "(5) Money transmitter —(i) In general. (A) A person that provides money transmission services. The term ``money transmission services" …"
   It would have been much preferred if the labels had been presented as follows:
   "(5) Money transmitter
   (i) In general
   (A) A person that provides …"
   The primary problem with nested labels appearing in a single line is that it is difficult to distinguish labels that introduce a new bullet in a list from a reference to a label. Regulatory text frequently *refer* or *self-refer* to regulations (see Section 3.3.2); for example, MSRB G-32 refers in several places to "subsection (a)(i)" and "paragraph (B)". The potential for error in extraction exists without a firm convention to determine when a label serves as in introduction of a bullet and when it serves to refer to a regulation.

3) The Cornell website posed other difficulties. Much of the information on bulleting structure comes from the /div labels. These labels, however, are inconsistent within the set of Cornell

documents. The div tag is used for document sections with content with both the attribute "class" and the attribute "psection". Sometimes "class" indicates content, but not always. Important content sometimes doesn't carry the "div" tag at all.

3.4.1.4 Non-HTML Challenges

The Case of the Trailing Bullet:

It is sometimes difficult to determine what bullet some text belongs to. For example, consider FINRA 6730 (a)(3)(C):

> "Collateralized Mortgage Obligation and Real Estate Mortgage Investment Conduit Transactions Before Issuance Transactions in Asset-Backed Securities that are collateralized mortgage obligations ("CMOs") or real estate mortgage investment conduits ("REMICs") that are executed before the issuance of the security must be reported the earlier of:
>
> (i) the business day that the security is assigned a CUSIP, a similar numeric identifier or a FINRA symbol during TRACE System Hours (unless such identifier is assigned after 1:00:00 p.m. Eastern Time, and in such case, such transactions must be reported no later than the next business day during TRACE System Hours), or
>
> (ii) the date of issuance of the security during TRACE System Hours.
>
> In either case, if the transaction is reported other than on the date of execution, the transaction report must
> be designated "as/of" and include the date of execution."

(C)'s sub-bullets are (i) and (ii), however, there is text in the paragraph after (ii)'s paragraph starting with "In either case" that appears to apply to both (i) and (ii). In such a case, we have several possibilities for action:

1) Consider this text as part of the C bullet, in which case we would attach it to the text that ends with "earlier of".
2) Consider this text as a postscript to both 6730(a)(3)(C)(i) and 6730(a)(3)(C)(ii), and therefore mark it as a postscript, and in the distribution phase, distribute it to the ends of both.
3) Make this text another (third) bullet of (C).

While the above example suggests that the second approach would work best, due to the relative infrequency of such examples, we decided that we did not have enough data to determine that this solution always worked. We chose approach 3 as the most neutral approach. Our algorithm, however, is subject to change if enough examples convince us that approach 2 is better. (Note that this sort of bulleted structure is not entirely natural/easily supported in bullet-friendly word processing programs such as Microsoft Word, which seems to support the constraint that one decreases indent ("outdents") only when one wants to start another bullet at a higher level not when one wants to have higher-level unbulleted text.)

The Latin Letter-Roman Numeral Mashup:

There is a potentially difficult case in which, in the absence of clear indicators, it is impossible to tell if a new label is the next element in the current bulleting, or is a first element of a nested list. This is the case in which the bulleting structure has lowercase letters followed by Roman numerals, and the current lowercase letter is (h). In that case, if an (i) is detected, there is

no intrinsic way to detect whether this (i) is the letter following (h) or the first of a series of Roman numerals. As far as we know, this case has not occurred, but it is potential source of error, though, we believe, with very low probability.

3.4.1.5 Building the Regulation Tree and Distributing Preambles

Once the bulleted structure is extracted from the HTML documents, the work on expanding the bullets --- that is, of distributing the preamble(s) onto a bullet's text --- can be performed. The algorithm works as follows: The output of the Bullet Structure Extraction component is a set of labels attached to chunks of text of the document. All labels are assigned label types (e.g., uppercase letter, lowercase letter, Arabic numeral, etc.). The document is traversed as follows:

```
For each paragraph
  If the label type is different than the previous label type
    the label type is not on the stack
      Create a new node and add it as a child of the previous node
      Save previous node as the parent of this node
      Put this label type on the stack
    Else
      Remove everything above this label type from the stack
      Find the parent of the current label type
      Create a new node and add it as a child of that parent
  Else
      Create a new node and add it as a child of the same parent of
      previous node
```

Consider how the algorithm would work on a simplified version of our example, FINRA 3240.
“3240. Borrowing From or Lending to Customers
(a) Permissible Lending Arrangements; Conditions
No person associated with a member in any registered capacity may borrow money from or lend money to any customer of such person unless:
(1)the member has written procedures allowing the borrowing and lending of money between such registered persons and customers of the member;
(2) the borrowing or lending arrangement meets one of the following conditions:
(A) the customer is a member of such person's immediate family;
(B) the customer
(i) is a financial institution regularly engaged in the business of providing credit, financing, or loans, or other entity or person that regularly arranges or extends credit in the ordinary course of business and
(ii) is acting in the course of such business
(C) The customer and the registered person are both registered persons of the same member;”
The stack starts out as empty; the tree has just one root. The root's text is the title of the regulation unit. Upon encountering (1), the system places “Number” on the stack, and (1) plus associated text as a child of the root. Next, upon encountering (2), the stack remains the same (same label type), while (2) plus associated text is placed as another child of the tree, sibling to (1). Next, upon encountering (A), the system places “Uppercase” on the stack while putting (A) plus associated text as a child of (2). This continues for (B). Upon encountering (i), the label type “Roman numeral” is placed on the stack, and (i) plus associated text is assigned as a child of (B). Similarly for (ii). Upon encountering (C), since this is a different label type than the previous

label type, but since the label type is on the stack, the stack is popped until that label type appears.

Once the tree is built, it becomes trivial to effect the distribution of preambles over bullet content. Every path in the tree corresponds to one fully expanded bullet. One need only read out the text associated with the nodes in the path to obtain the fully expanded and distributed bullet. The text associated with all the ancestors of the bullet is concatenated with the text of the bullet itself. Note: the expanded bullets are highly redundant, but this redundancy seems to be necessary for the tasks for which we need the expanded text.

3.4.2 Categorizing portions of regulatory documents by discourse function

3.4.2.1 Motivation

In our initial analysis of the corpus of regulation units, we noted that each regulatory document contained much more than just a set of regulations. Consider the following fragment of FINRA Rule 5330:

“5330. Adjustment of Orders
(a) A member holding an open order from a customer or another broker-dealer shall, prior to executing or permitting the order to be executed, reduce, increase, or adjust the price and/or number of shares of such order by an amount equal to the dividend, payment, or distribution on the day that the security is quoted ex-dividend, ex-rights, ex-distribution, or ex interest, except where a cash dividend or distribution is less than one cent (\$0.01), as follows:
(1) Cash Dividends: Unless marked ``Do Not Reduce," open order prices shall be first reduced by the dollar amount of the dividend, and the resulting price will then be rounded down to the next lower minimum quotation variation …
…
(d) The term “open order” means an order to buy or an open stop order to sell, including but not limited to “good 'til cancelled,” “limit” or “stop limit” orders which remain in effect for a definite or indefinite period until executed, cancelled or expired.
(e) The provisions of paragraph (a) of this Rule shall not apply to:
(1) orders governed by the rules of a registered national securities exchange;
(2) open stop orders to buy;
(3) open sell orders; or
(4) orders for the purchase or sale of securities where the issuer of the securities has not reported a dividend, payment, or distribution pursuant to SEA Rule 10b-17.
Amended by SR-FINRA-2009-084 eff. April 19, 2010.
…
Amended by SR-NASD-94-46 eff. Sept. 15, 1994.
Selected Notices: 93-61, 94-9, 94-28, 94-63, 10-10.”

We can see, for example, that (a)(1) is a regulatory piece of text (in fact, an obligation), that (d) is a definition, that (e) is a particular type of regulation that gives certain exceptions to the rule, that the bolded portion gives some history of the regulation, and that the underlined portion gives references.

We were interested in developing automated methods to recognize different types of text:

1) Definitions vs. Regulations: We focused on distinguishing definitions from regulations, first, because definitions can be useful in constructing a domain ontology; second, because different techniques can be used to translate regulations than definitions.
2) Reference Structure: Regulations and definitions frequently refer to other regulations and definitions. In automating the translation of regulatory text into formal executable rules, we need to incorporate this referenced material. To aid in doing so, it would be useful to construct a dependency graph of regulations and definitions.

   We wanted to determine whether a regulation contained a reference and reference type:

   - *Cross-document*, a reference to a different regulatory document
   - *Intra-document*, a reference to a part of a regulation that is in the same regulatory document but refers to a different *branch* of the regulation in the tree induced by the bulleted structure (as discussed in Section 3.1.1, above), or
   - *Intra-branch*, a reference to one's own branch.

   In the example above, (e) gives an intra-document reference; (4) gives a cross-document reference. We were likewise interested in determining whether a definition is defining the term or is a reference to some other definition.
3) Exceptions: Determining whether a rule contains an exception is essential to formalizing regulatory text. A standard approach for representing exceptions in formal logic, McCarthy (1986), is the construction of rule sets such as $P(x) \& \sim ab(x) \rightarrow Q(x); C(x) \rightarrow ab(x)$, where *ab* predicates are used to formalize exceptional conditions. Recognizing exceptions is essential for doing such formalization.
4) Regulation Type: Identifying different types of regulation is of particular interest. Drawing from recent work on the developing standard LegalRuleML (Athan et al., 2013), we focused on identifying obligations, permissions, prohibitions, penalties, and reparations. The first three of these categories are the most frequently found in U.S. financial regulatory text.

3.4.2.2 Technical Approach

We used support vector machines (SVM) with supervised learning for our classification experiments, for reasons of both computational efficiency (Joachims, 2002) and performance (Yang and Liu, 1999). All experiments discussed in this paper considered bag-of-words features. We were interested in determining the baselines that we would get using even simple features. As it turned out, we had strong results for many categories, surpassing our expectations.

All text was preprocessed to expand all bullets. Then individual sentences were labeled by hand. We had 3106 total sentence annotations: 1029 regulations, 592 cross-document, 699 intra-document, 46 intra-branch (relatively infrequent in text considered), 572 definition-direct, and 168 definition-reference. For determining regulation type --- whether obligation, permission, prohibition, penalty, or reparation --- we had 733 annotations total: 449 obligation, 146 prohibition, 85 permission, 46 penalty, and 7 reparation. For exceptions, we had 952 annotations: 256 exceptions and 696 non-exceptions.

We ran several experiments examining identical categorizations using a one-, two-, and three-tiered approach. We ran regulation / definition / reference structure annotations as a one-tier experiment, and also ran it as three tiers: first classifying regulations and definitions; then classifying referencing regulations and non-referencing regulations, as well as referencing definitions and non-referencing definitions; and finally classifying by type of reference. While we had conjectured that experimenting by tier would yield higher Precision, Recall, and F1, this was not always the case. The results shown in Section 4 are the best results obtained. Our results were quite good with an F1 score of at least .8 in most categories. In general, there was a correlation between the quantity of training data and the F1 score, which indicates that more training data could be used to improve scores in those areas in which we did not do well.

3.4.2.3 Novel aspects of our approach

This appears to be the first attempt to categorize text using large amounts of regulatory documents for which bulleted text has been expanded. Indeed, being able to run these experiments was one of our motivations for developing automated methods to expand bulleted text. Our initial attempts to use standard splitting methods to divide bulleted text into sentences, and to then annotate these sentences, could not proceed because it was impossible to determine whether a bulleted fragment belonged to a prohibition, obligation, or permission. We believe that the high results we obtained are largely due to our preprocessing of bulleted text.

Other researchers (Sebasiani, 2002; Hachey et al., 2004) have attempted to categorize by regulation type, but some of the dimensions along which we characterized text, such as the existence and type of references, appear to be new. This may be due to the novel automated translation between regulatory text and formal executable text that drove this research. The simplicity of our approach in using simple bag-of-word features is often used in machine learning applications, but appears to be rarely used for this problem area.

3.4.3 Ontology Extraction and Merging with existing Ontologies

3.4.3.1 Motivation

Ontology extraction consisted of extracting important concepts and relations for each regulation, organizing them into an ontology, and merging them together into one consistent ontology. Ontology extraction and merging is crucial for translating regulatory text to formal executable rules. Merging is especially important, because unless one realizes when two terms --- e.g. “give notice” and “notify” --- refer to the same concept, or when one concept is a sub-concept of another --- e.g. “cash” and “money” --- it is not possible to ensure consistency and to form a set of rules.

Ontology merging is especially important in this domain because of the existence of the LKIF (Legal Knowledge Interchange Format) ontology (Hoekstra et al., 2009) and the emerging Financial Industry Business Ontology (see http://www.edmcouncil.org/financialbusiness).

3.4.3.2 Technical Approach

Our approach to ontology extraction in the legal domain is guided by the following observations:
First, entities correspond to highly complex noun phrases. Second, actions are crucial to the legal domain, while they are often of minor importance in other domains. Third, concepts of agency are often crucial.

The novelty of our technical approach as presented comes from:

- the set of regular expressions used to identify concepts
- the focus on actions as well as entities
- the development of a co-reference process for extracted entities
- the use of dictionaries, and in particular, the use of dynamic calls to online dictionaries during the merging process

3.4.3.2.1 Extracting Entities and Actions:

1) Extracting entities:

There is a close correspondence between desired entities in an ontology and noun phrases in text. For example, consider the following fragment from 15 USC § (b) (1), with several desired entities highlighted in yellow. (Note that we do not highlight repeat entities but our system does extract these):
"A broker or dealer may be registered by filing with the Commission an application for registration in such form and containing such information and documents concerning such broker or dealer and any persons associated with such broker or dealer as the Commission, by rule, may prescribe as necessary or appropriate in the public interest or for the protection of *investors*. Within forty-five days of the date of the filing of such application (or within such longer period as to which the applicant consents), the Commission shall…"

Entities that are candidates for an ontology in the regulatory domain would include *broker, dealer, Commission*, and *form.* It would also include more complex noun phrases such as *public interest*, *protection of investors*, *filing of (such) application*, and even *date of filing of (such) application*. We extracted noun phrases by putting regulatory text through a part-of-speech tagger and chunker (which does shallow parsing), then matching against the following regular expressions:
a. (NN|NNS|NNP|NNPS)+
b. (VBN | VBG) (NN|NNS|NNP|NNPS)+
c. ((RB* (JJ | VBN | VBG)+) | (JJ | VBN | VBG)*) (NN | NNS | NNP | NNPS)+
d. ((RB* (JJ | VBN | VBG)+) | (JJ | VBN | VBG)*) (NN | NNS | NNP | NNPS)+ ((IN | IN DT) ((RB* (JJ | VBN | VBG)+) | (JJ | VBN | VBG)*) (NN | NNS | NNP | NNPS)+)+

Part-of-speech tags in the regexes above correspond to Penn Treebank tags and can be found at http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html. For example, *public interest* matches regex 1, *forty-five days* matches regex 2, and *date of filing of such application* matches regex 4.

2) Extracting Actions:

While noun phrases typically refer to entities of interest, using verb phrases to identify actions results in very low recall. For example, consider the following excerpt (Section 15 c.1.A) of the Securities Exchange Act of 1934:

> "No broker or dealer shall make use of the mails or any means or instrumentality of interstate commerce to effect any transaction in, or to induce or attempt to induce the purchase or sale of, any security (other than commercial paper, bankers' acceptances, or commercial bills) otherwise than on a national securities exchange of which it is a member, or any security-based swap agreement (as defined in section 206B of the Gramm-Leach-Bliley Act), by means of any manipulative, deceptive, or other fraudulent device or contrivance."

Note that most of the verbs in this passage ---- *make, effect, induce, attempt, is* --- don't correspond to actions of interest for this domain.

We explored two other strategies for identifying actions. The first was finding action verbs that were in close proximity to deontic operators related to obligation, permission, and prohibition. We identified word phrases corresponding to such deontic operators, wrote regular expressions that corresponded to these word phrases, and extracted these. For example, the regular expression *[MD] not [VP]* corresponds to phrases such as *shall not distribute* or *may not transfer*, which in turn would yield *distribute* and *transfer* as action concepts.

A second proposed strategy was using a semantic parser to help identify agents and actions in text. While we ultimately used a semantic parser for the Final Task, Rule Template Slot Filling, semantic parsers were too slow and unreliable to be used for the ontology extraction task. In general, since parsers fail on many of the sentences in our documents, it is best to use them only for tasks for which there is no other alternative. Since part-of-speech tagging, chunking, and matching regular expressions gave us good results for ontology extraction, we decided to avoid parsers given the current state of parsing technology.

3.4.3.2.2 Ontology Merging

Successful merging of ontologies depends both on recognizing when nodes in different ontologies match (a version of the co-reference problem) and on recognizing when different nodes in different ontologies are related (ontology linking).

Having an automated co-reference procedure is important for ontology extraction and alignment. If there are two documents in a corpus, one referring to "the Treasury Department" and one referring to "the Department of the Treasury", one would not want to have two distinct concepts in one's ontology. Rather, it is desirable that the system realize that these are two labels for the same concept. Automated co-reference is also important for ontology evaluation, especially when evaluating against a Human Gold Standard. A human could look at a text that has the string "associated person of the member" and extract the concept "person associated with member" (this is one of hundreds of actual examples that we have come up with in our set of training documents). For the evaluation, an additional (neutral) human could be asked to

determine when phrases co-refer, but it was determined to be too time-consuming to be practical. There are many types of co-reference, including anaphora reference, determining what noun a pronoun such as “they” or “it” refers to. We distinguish between semantic co-reference, in which entirely different syntactic phrases may refer to the same entity (e.g., “President Obama” vs “the President of the US”) and syntactic co-reference, in which similar syntactic phrases refer to the same concept (e.g. “cashing of checks” vs “check cashing”). For the ontology alignment and evaluation tasks, we focus on developing automated syntactic co-reference checkers. Our procedure works by stemming words in a phrase, removing common “stop” words such as *of, the, or,* and *to*, and checking to see if the list of remaining words are identical. This allowed us to successfully co-ref hundreds of pairs of phrases that refer to the same entity.

For ontology linking we used shallow parsing methods (such as chunking) on external sources to help determine where in an ontology a new concept belongs. When extracting concepts from text, it is often the case, if one is given a sufficiently rich upper / background ontology that concepts extracted from new documents will correspond to concepts in the upper/ background ontology. However, there is nearly always a set of concepts that do not match any existing node in the ontology. In such cases, we turned to external sources. The following sources have proved particularly helpful: investopedia.com, glossary.reuters.com, businessdictionary.com, and www.thefreedictionary.com (a legal dictionary and a financial dictionary). In most cases, the first few words of the first sentence in a definition gave us valuable information. From a sentence fragment the core noun is often the concept under which the new sentence should be organized under.

Full parsing of the definitions in these dictionaries was not feasible. We realized that because these dictionaries are so limited in scope (containing only a few thousand terms compared to several hundred thousand terms in standard dictionaries), the presence of a term in one of these dictionaries itself conveys useful information. Thus if a concept appears in a financial dictionary it is then treated as a financial concept; if a concept appears in a legal dictionary it is then treated as a legal concept. We hypothesized that creating such links would be useful for the intended purpose of our ontology and the construction of automated rules as it allows for general class of financial concepts to be inherited by subclasses.

Checking that a newly extracted concept appears in one of these dictionaries presents was challenging. Because several of these dictionaries do not have good internal search engines, we have found it to be most effective to use Google search that is restricted to the sites of these dictionaries. While this approach worked for manual testing, it did not work when we automated the search to thousands of extracted concepts because Google banned so many requests from single user. The only way to do more searches was to enter CAPTCHAs provided by Google: this involved having a human sitting by the system as it was processing. Not only was this very time consuming, several thousand terms took several workdays, but it also violated the spirit of the seedling task, which was to develop an automated end-to-end process to extract the ontology. We solved this problem by web scraping the dictionaries and thus removing our dependency on Google searches.

As discussed in detail in Section 4, results of our ontology extraction software, measured against two human gold standards, were strong: Post-adjudication, precision for concepts was .83 and for relations.74; recall was .85 and .97 respectively, yielding F1 of .83 for concepts and .86 for relations.

3.4.4 Rule Template Slot Filling

During the course of TAILCM research and as a result of our greater understanding of the research problem and of the available technologies, it became apparent that full scale translation of regulatory text to formal executable rules would not be as we had first envisioned. Instead of having executable rules that determined compliance vs. non-compliance we created rule templates that would identify all major pieces of a regulation and would serve the same purpose. The rule template identified:

- The type of regulation: obligation, prohibition, or permission. (Since penalties and reparations are relatively uncommon in U.S. financial regulatory text, we did not attempt to deal with them.)
- The regulated action
- The agent of the regulated action: that is, the organization(s) or individual(s) performing the action
- The patient of the regulated action: that is, the entity on which the regulated action is performed
- The conditions under which the action is thus regulated
- The exceptions to the regulation

These components can be seen as the main building blocks that constitute a formal executable rule. There is still work to be done in formal rule construction, most especially in determining and ordering quantifiers, but this is a large part of the task. In addition, this output is independently useful. For example, this can speed up the process by which humans manually construct formal, executable rules, and it can be used to detect many (though certainly not all) inconsistencies in rules. Finally we had used rule templates as an overall evaluation of how well core pieces of regulations are being extracted.

Our approach combined extensive preprocessing of regulatory text, the use of a semantic parser and dependency parser, post-processing of the output of the parsers, and special-purpose rules to extract the desired information from the post-processed parser output. Preprocessing began with expanding bulleted text, as explained in section 3.3.1. Additional preprocessing was necessary to temporarily remove section titles, number headings, and other informative meta-text that could prove useful in post-processing stages, but would interfere with parsing.

We used the LTH semantic parser (at http://nlp.cs.lth.se/), which is built on top of a dependency parser. Semantic parsers, based on case grammars (Charniak and Wilks, 1976), are intended to not only recognize the syntactic function of words and clauses in a sentence, but the semantic roles, in particular of actions, as well. Consider the sentence “The form shall be filed by the broker.” While a correct syntactic parse should identify “by the broker” as a prepositional phrase modifying “filed” and “the broker” as the object of that prepositional phrase, a correct semantic parse should identify “the broker” as the agent of the action “filed.” We were especially interested in the ability of the semantic parser to recognize agents, actions, and patients. Unfortunately semantic parsers, like syntactic parsers, have low accuracy rates. In post-testing and end-of-program analysis, we calculated that the LTH parser and its associated

dependency parser had been able to parse 72% of the sentences it had been given. Of the sentences it was able to parse 14% accurately yielding a 10% overall accuracy rate.

A common error was the unwarranted identification of prepositions to a single role. For example (in the sentence above) “by the broker” indicates that “the broker” is the agent of the action preceding the prepositional phrase, but this is not true for the sentences like: “Taxes shall be filed by April 15” and “Trash cans must be placed by the curb.” “by April 15” denotes the time by which the action must be done; “by the curb” denotes the location where the action must be done. Similarly, LTH generally identified the object of a prepositional phrase beginning with “to” as a location, although that is only one of the word’s functions. (Consider the phrases “to the extent” or “to no end”). In addition, the dependency parser that LTH used made frequent errors in parsing prepositional phrases.

We solved the problem through special-purpose rules that we developed to handle regulatory text. Because of the limited ways in which regulations are expressed, we found that a few dozen rules sufficed to correctly handle most of the regulatory text. These special-purpose rules used the output of the semantic and dependency parsers, but had checks that overrode assignments from the parsers for specific cases. This allowed us to sometimes auto correct part-of-speech tagging, for example: TAILCM software recognized that a word ending in “ing” was more likely to be a noun than a verb when it came right before a deontic operator, and thus overrode the tag. Also TAILCM software recognized that in a construct like “NN <deontic operator> be VBN”, the NN was likely to be a patient, rather than an agent. Much of the recognition of conditions and exceptions was driven by recognizing particular keywords and key phrases (for example, “in the event that” indicates a condition).

We used human adjudicators for judging the output of the developed software. Below are selected screen shots of what the human adjudicators saw. In figure 4a note the missing condition in this sample output. In Figure 4b note that catalog has been tagged by the parser as a noun, so nothing is extracted.

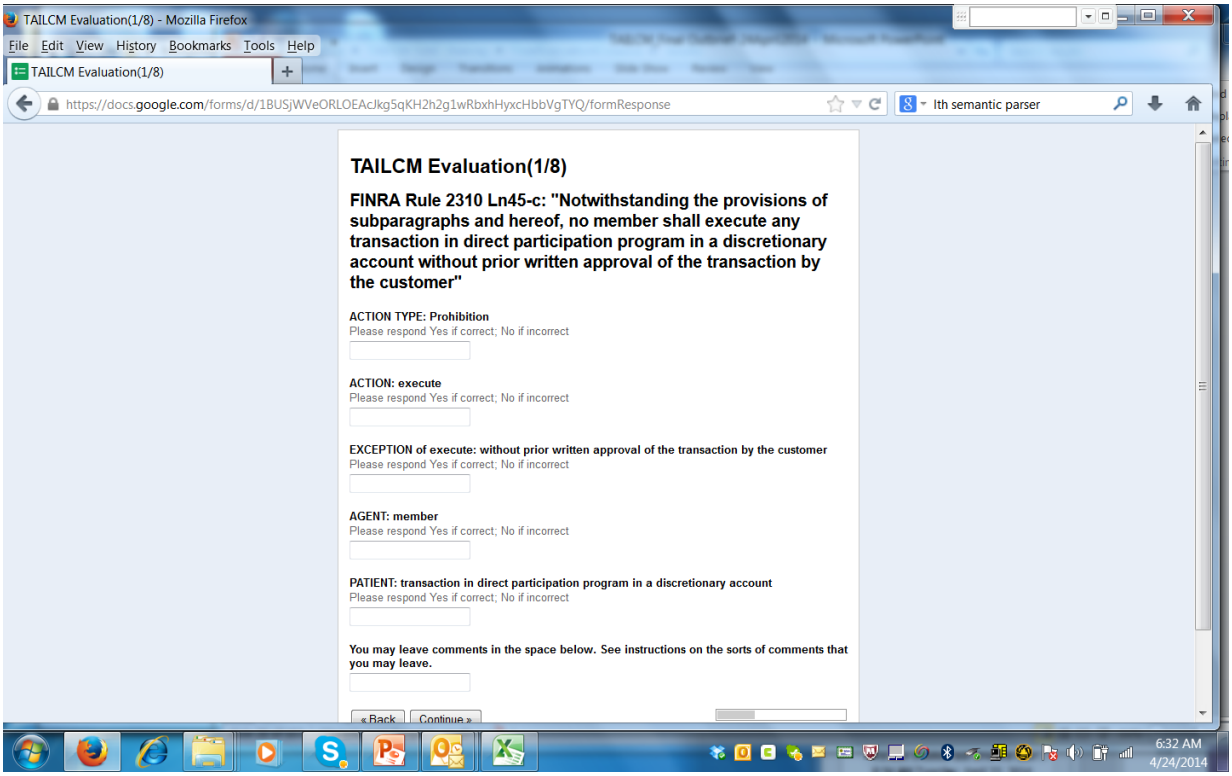


Figure 3: Sample screen shot of evaluation in progress.

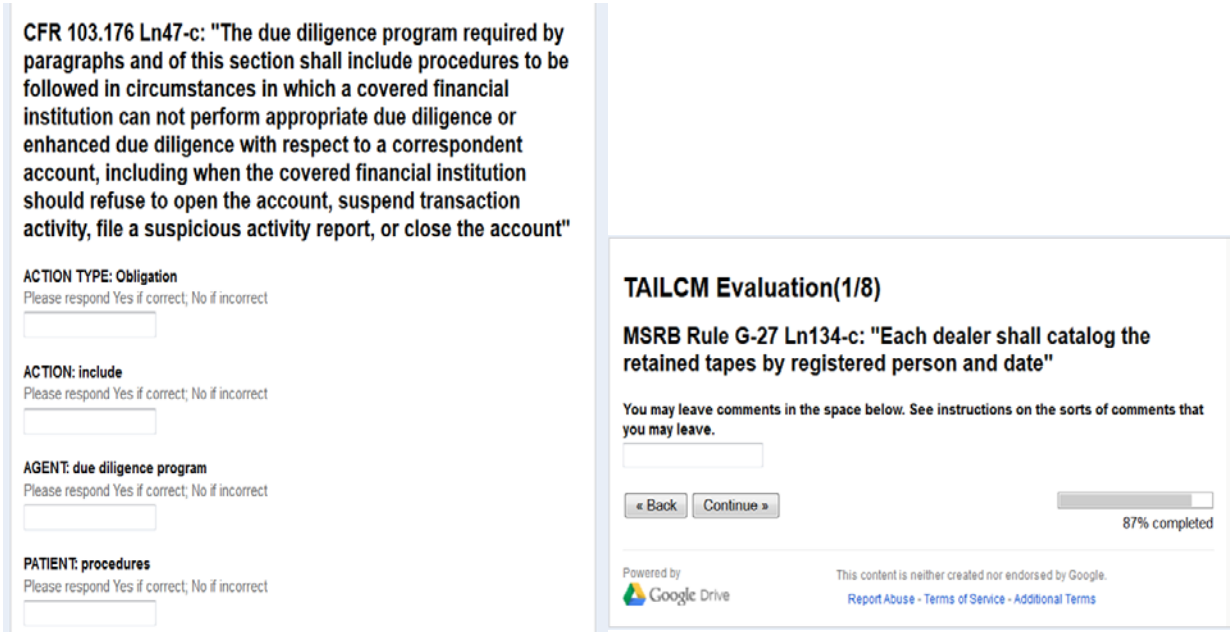


Figure 4: Extracted concepts for (a) CFR 103.176 Ln47c (b) Rule G-27 Ln134-c.

Overall results were very strong: Precision ranged from .88 - .90 on two different data sets; recall on a sample was .76, for an extrapolated F1 of .82. Our best performance was on regulation type; our poorest performance was on extracting conditions. As far as we know, there has been no similar work in automating the extraction of rule components from regulatory text. Semantic and dependency parsers are widely used, but not for this task and not for sentences of this complexity.

4 Results and Discussion

We give results below for three of our four research areas. Our work on expanding bulleted text was evaluated by analysis of the algorithm that implemented the specification, but since this was intended to serve as a preprocessing step for our other research, we did only internal tests.

4.1 Results for Categorizing portions of regulatory documents by discourse function

Our results are given in Table 1. Overall, the results were very good. The few cases of poor or mediocre performance --- e.g., recognizing reparations, and intra-branch references --- appear to be due to a lack of training data.

Classification	Utility of Classification	Results: F1 >, R >, P >	Comment
Definition vs. Regulation	Definitions used to build ontology; early targets for automated translation	0.95, 0.99, 0.92	Specific phrases ("definition", "is taken to mean") probably help clustering algorithm.
Referencing vs. non-referencing regulation	Enables the automated construction of dependency graph	0.86, 0.83, 0.90	Improve by training to recognize regex
Type of referencing regulation (intra-branch, intra-doc, cross-doc)	Constrains way dependency graph is used to generate rules	Cross doc: 0.82, 0.81, 0.83 Intra doc: 0.78, 0.78, 0.78 Intra branch: 0.53, 0.44, 0.67	Relatively weak results for intra-branch, probably due to lack of training data.
Type of definition (direct: def is spelled out in text vs. ref to other document for definition)	Important for constructing parts of dependency graph	Def'n direct: 0.93, 0.96. 0.9 Defn ref: 0.70, 0.62, 0.81	Results could be improved by recognizing and parsing referenced rule labels.
Exception vs. non-exception	Supports automating structure of rule sets	0.93, 0.92, 0.93	Specific phrases ("with the exception of","unless") probably help clustering algorithm
Type of reg (obligation, prohibition, permission, penalty, reparation)	Templates for each rule type to help automate translation	Obligation: 0.90, 0.95, 0.86 Prohibition: 0.83, 0.77, 0.91 Permission: 0.78, 0.76, 0.82 Penalty: 0.75, 0.67, 0.86 Reparation 0.40, 0.33, 0.50	Possible need to annotate at finer grain; also use more features than bag-of-words

Table 1: Categorizing Regulatory Documents by Discourse Function

4.2 Results for Ontology extraction and merging with existing ontologies

The ontology extraction and merging software was evaluated against two human gold standards (HGS), an HGS developed based on experts' outside knowledge of the financial regulatory domain (constructed by Exprentis, Inc.), and an HGS developed based on a close reading of the text (constructed by an ontology expert at Leidos).

We computed an initial count of True Positives, False Positives, and False Negatives, and then asked an adjudicator to determine if any of the entries in the False Positive List could count as True Positives. This was done because machines identify concepts better than humans, and

penalizing machines for their thoroughness would not give an accurate picture. Post-adjudication precision rose significantly against the HGS constructed by the domain experts, as can be seen in Figures 5 and 6.

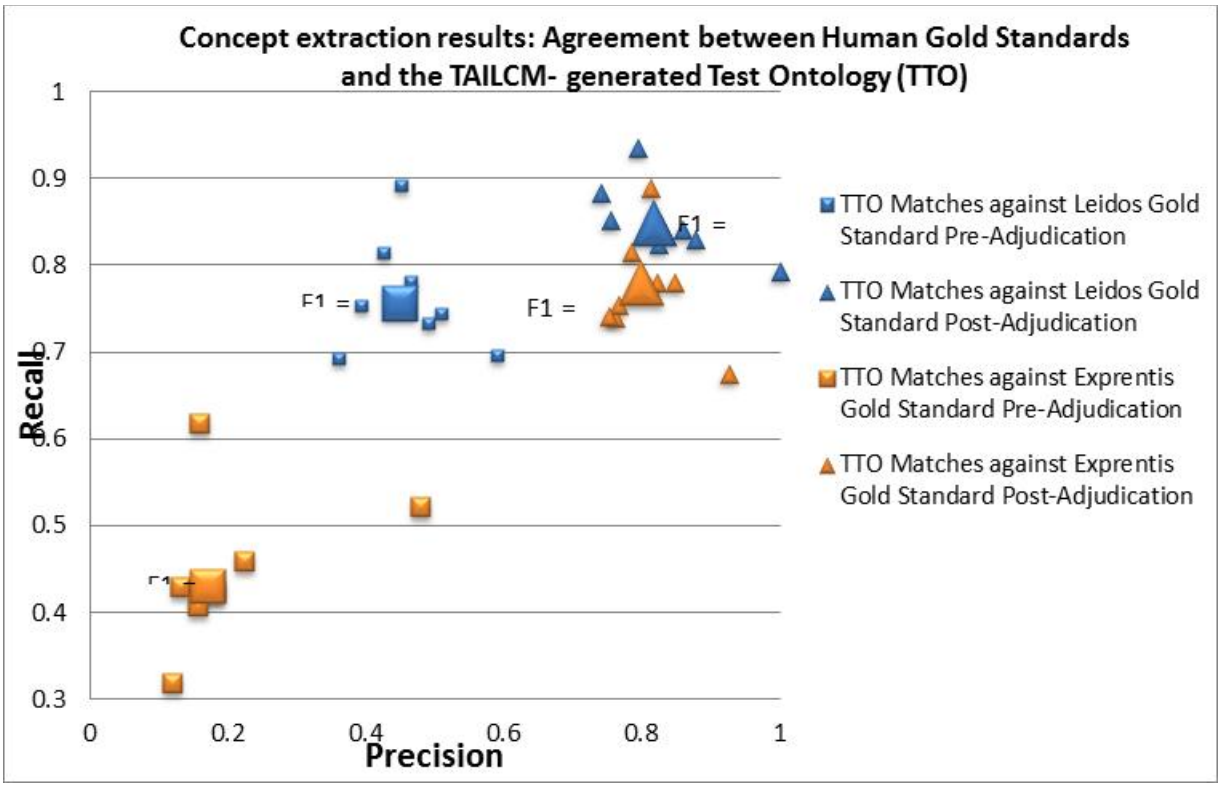


Figure 5: Pre-adjudication recall and precision

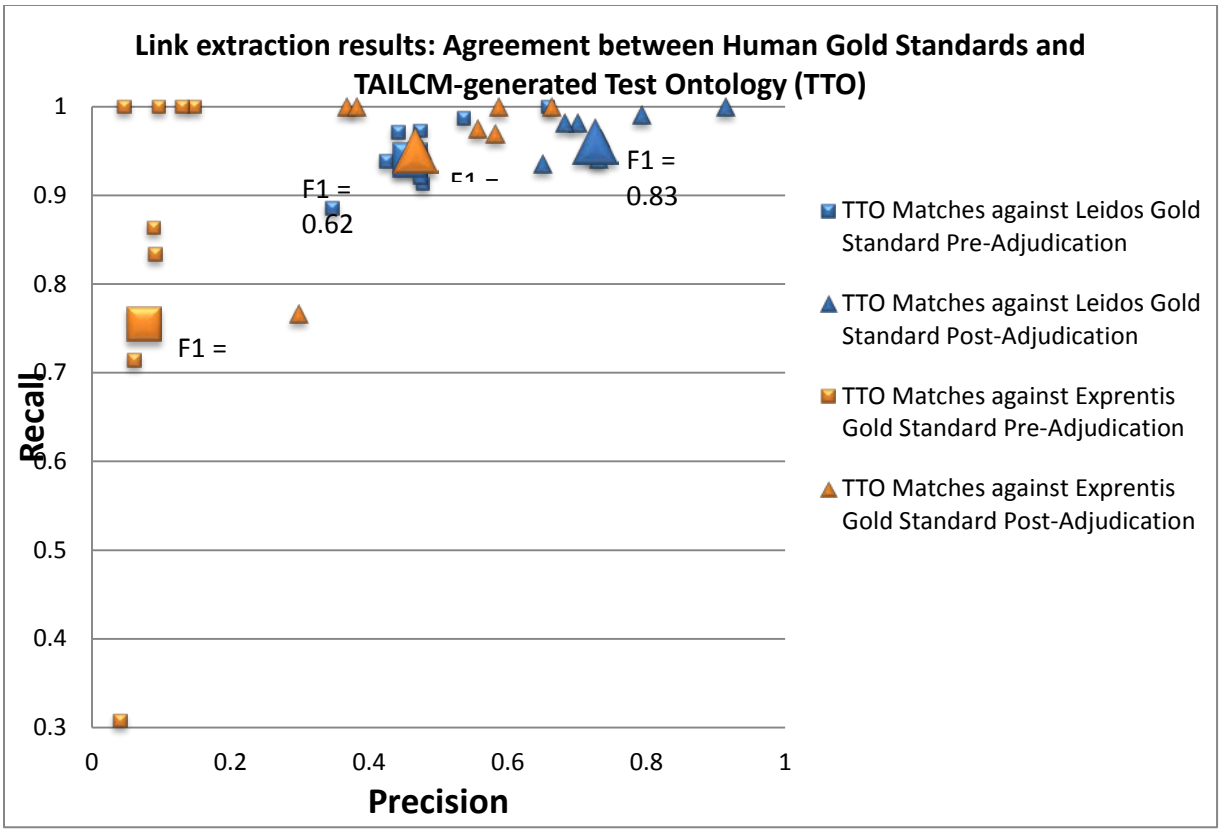


Figure 6: Post-adjudication recall and precision

A per-document comparison of TAILCM extraction vs. human extraction can be seen in Table 2. Note that TAILCM always extracts more concepts than humans.

Table 2: Concept count for 8 regulations.

	7 USC 9A	7 USC 13a	17 CFR 166	FINRA 6460	FINRA 6120	FINRA 5350	FINRA 5220	NASD 2340
Word count	885	1045	80	608	455	344	417	1085
Exprentis concepts	28	77	21	41	50	26	22	90
Leidos concepts	79	125	23	76	75	39	85	179
TAILCM concepts	158	239	28	146	124	75	122	265

Figure 7 demonstrates the relationship between TAILCM's True Positives and False Positives and the two human gold standards.

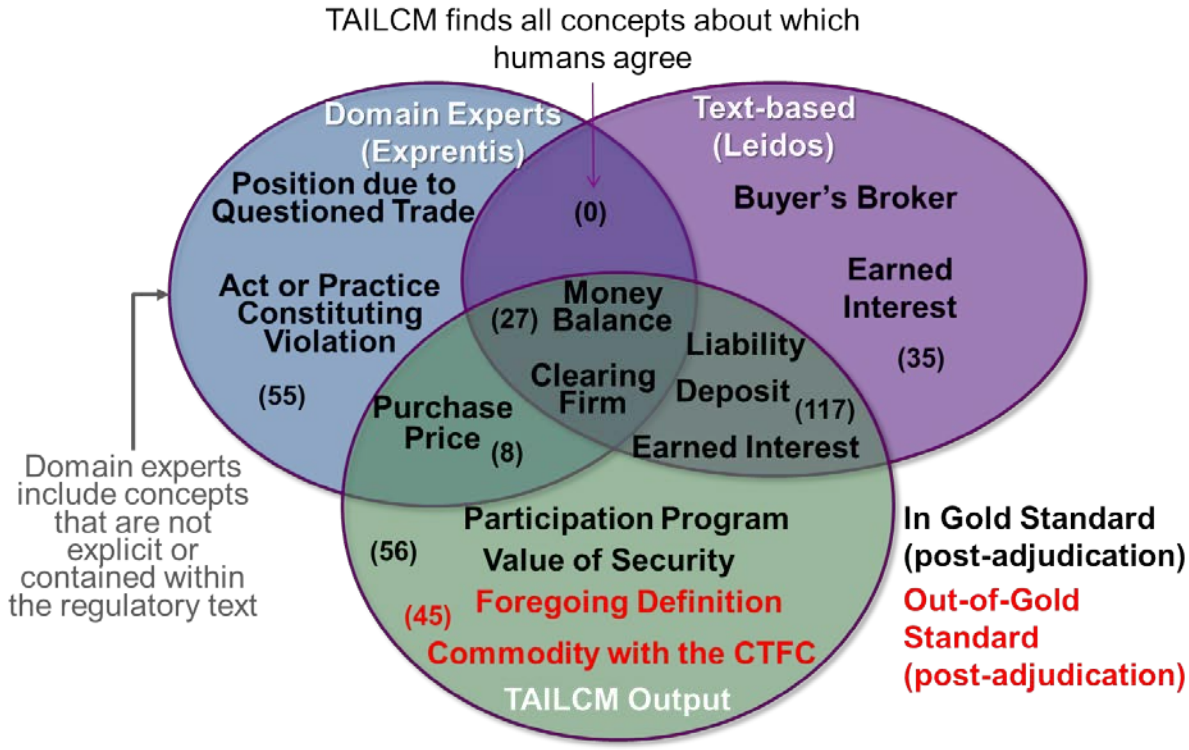


Figure 7: TAILCM's TP and FP for two human gold standards

4.2 Results for Rule Template Slot Filling

Human adjudicators judged two sets of output produced by TAILCM's software. The first set was created with humans correcting minor errors in the output of the dependency parser such as incorrect determinations of what word a prepositional phrase modified. No major errors, such as mislabeling of a noun or verb, were corrected, and no errors of the semantic parser (as opposed to the dependency parser) were corrected. In the second set, no parse was corrected. Results were similar on both sets, apparently because our special purpose rules had already taken into account the possibility of these errors. Figure 8 shows the precision, by majority vote, on each of the categories of rules components that we extracted.

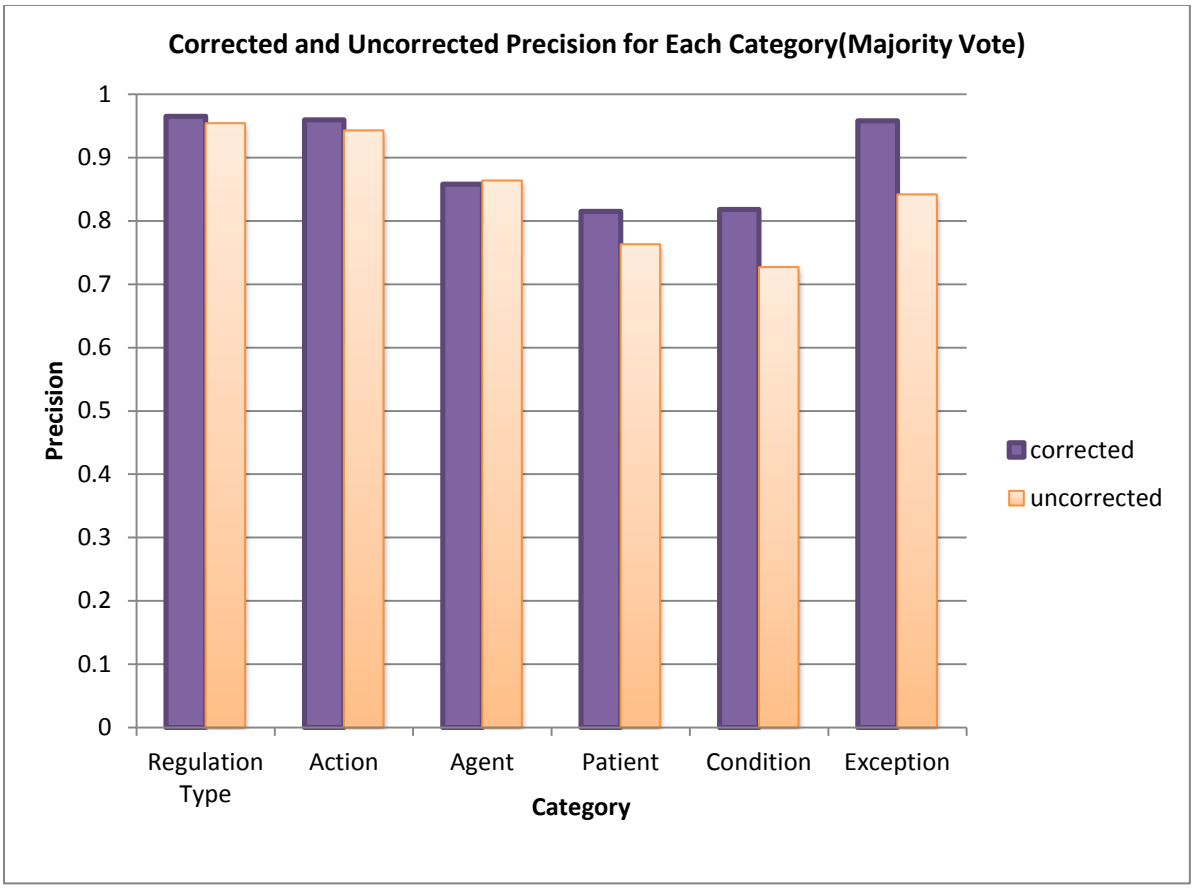


Figure 8: Precision for two output sets from TAILCM prototype

Figure 9 shows the precision calculated according to each adjudicator. As can be seen, inter-rater agreement among the annotators was very high.

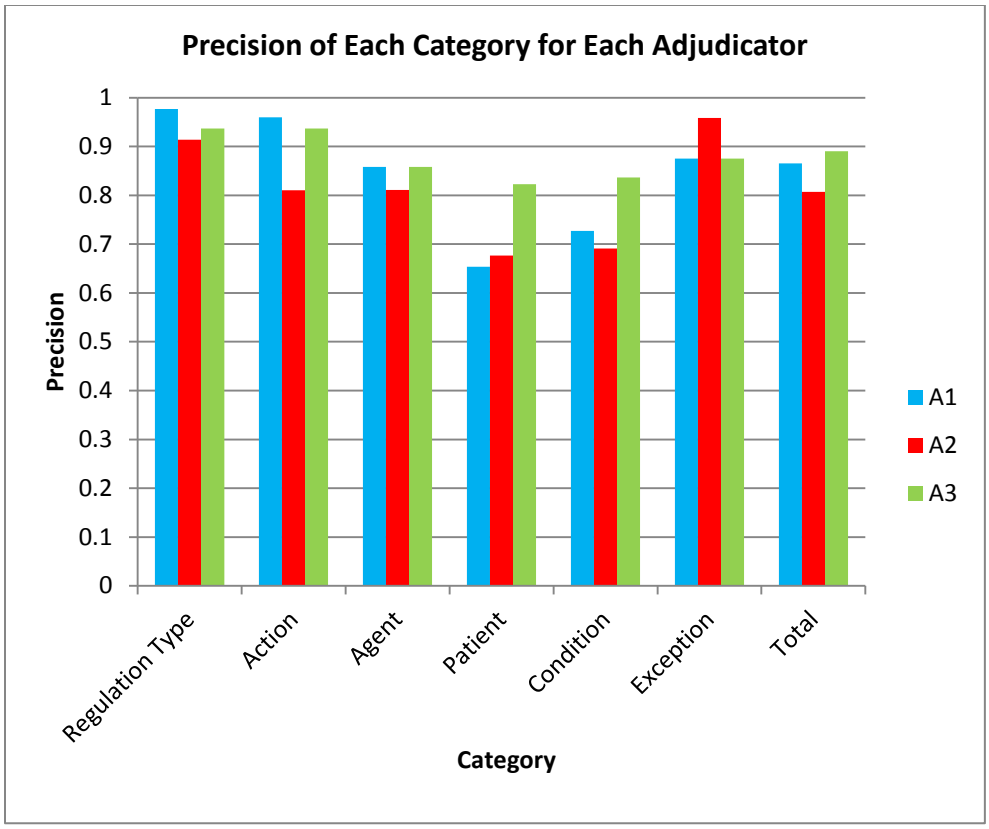


Figure 9: Inter-rater agreement among the human testers

In contrast, inter-rater agreement for recall figures was close to 0; probably due to unclear instructions given to adjudicator. We then constructed a Human Gold Standard for a random sample of the output, which we used to calculate recall and extrapolated F1 based on the precision scores already obtained. Note that recall and F1 rates are not given for regulation type, since that was a multiple choice question, for which recall is meaningless.

Table 3: recall on sample, with extrapolated F1

	Number in gold standard (Q1)	Found by software (Q1)	Recall (Q1)	Precision over questionnaires (Q1-Q8) (corrected and uncorrected averaged)	Estimated F1 (Q1 – Q8) (extrapolating recall from Q1 to all)
Actions	51	46	.90	.95	.925
Agents	47	40	.85	.86	.86
Patients	47	31	.66	.79	.72
Conditions	18	6	.33	.77	.47
Exceptions	8	7	.87	.90	.89
Total items	171	130	.76	.89	.82

The results are shown in Table 3. Figures 10, 11, and 12 display same results graphically.

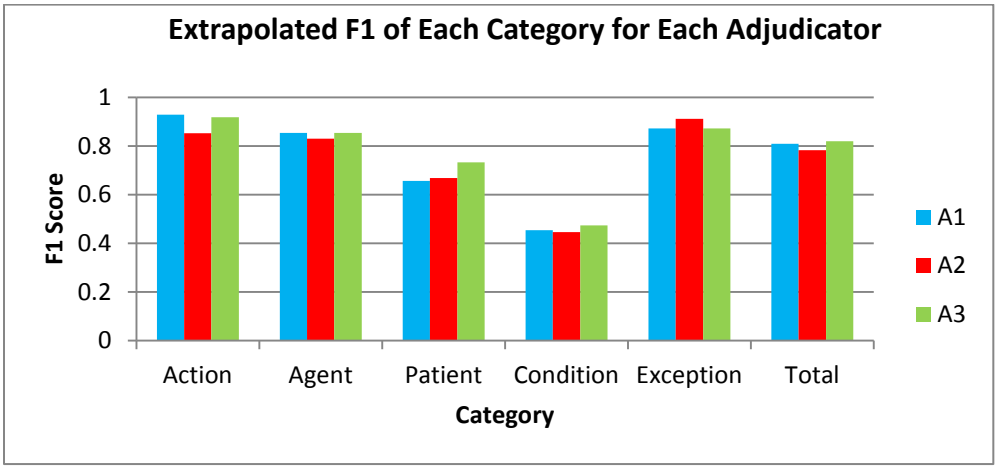


Figure 10: Extrapolated F1 for each category for each adjudicator

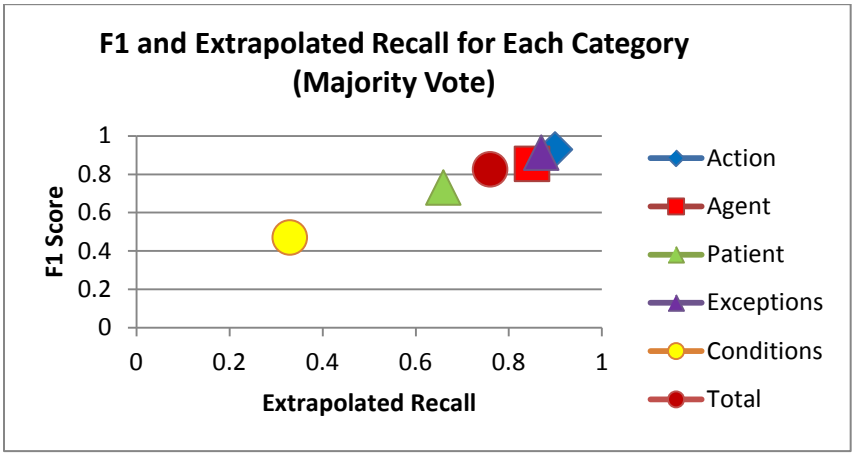


Figure 11: F1 and Extrapolated Recall for each category (Majority Vote)

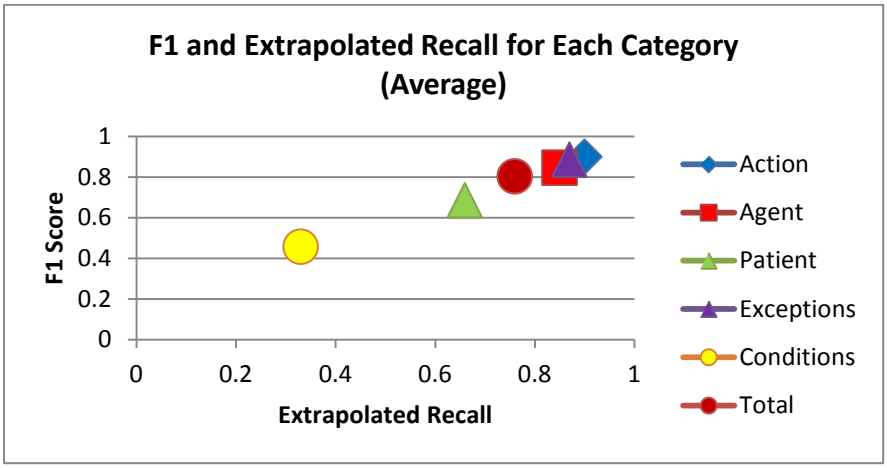


Figure 12: F1 and Extrapolated Recall for each category (Average)

5 Conclusions

Limitations of parsing technologies:

The most important lessons learned related to the state-of-the-art of parsing technology. When we began, we assumed Manning's (2011) figure of 57% parsing accuracy. Further analysis suggested that this figure was too high. Based on a sample of sentences taken from our training corpus at the beginning of our seedling research, we estimated parsing accuracy at 30%. Probably because we unknowingly chose relatively simple sentences, this figure also turned out to be much too high. We currently estimate that parsing accuracy is at about 10% for highly complex sentences such as the ones that are typical in regulatory text.

Despite low accuracy, we performed very well on the final evaluation, to which parsing was central. This was due in large part to the careful construction of rules that overrode the parsers. One lesson to be learned from this may be that we can, in certain circumstances, get around poor parsing technology. We believe that the more fundamental lesson is that we need to develop better and more accurate parsing technology.

Developing parsers is traditionally a slow enterprise, involving the painstaking annotation of parse trees for large training sets (treebanking). Humans treebank at a rate of less than 100 words per hour, and require more than six months of training to reach competence. LDC has shared with us some ideas that could be used to greatly speed up the treebanking process. They have suggested relying on untrained annotators for judgments they can easily make (at a rate of >1000 words per hour, with no training necessary), such as the scope of conjunction, and then constraining the parser to obey these judgments. Even before better parsing technology is available our results on the Final task indicate that we can make good progress now.

Importance of constraints:

During the course of the last few months of our research, we realized that regulations cannot be full spanned by the categories of actions, agents, patients, conditions, or exceptions. Also very important are constraints. While constraints in mathematics are often conflated with conditions, in regulations, they act quite differently. For example, in the regulation "Forms must be submitted to the commissioner", "to the commissioner" is not the condition, but is a constraint specifying how the action is to be done.

While an accurate semantic parser would in the long term help distinguish between conditions and constraints, we believe that a near-term situation would involve the development of a set of rules to extract constraints.

Importance of idioms:

It is often difficult for semantic parsers to identify actions correctly because of the mismatch between verbs and actions. "Notify John" and "give notice to John" indicate the same action; however, while a semantic parser would accurately identify "notify" as the action and "John" as the patient of the first phrase, it would inaccurately identify "give" as the action and "notice" as the patient of the second phrase. Extending our ontology building through the use of dictionaries could be a first step in solving this problem. In the short run, compiling a list of the top several dozen action idioms could greatly enhance performance.

Next steps for complete translation:

To generate complete rules, three subtasks are still necessary:

- Mapping snippets of text onto formal logical terms. This is a feasible task, given our work on extracting and merging ontologies.
- Extract constraints. As discussed above, this is probably feasible given construction of special-purpose rules, or ideally, a better semantic parser.
- Determining quantification. This is feasible for simple cases, where all quantification is universal, or there are limited existential quantifiers. Regulations with a large number of existential quantifiers would probably have to be handled separately, at least in the short term.

Overall lessons learned from the TAILCM seedling:

1) Limitation of knowledge-based approaches: Our experience during the midterm evaluation with two different human gold standards, one developed by domain experts, and one closely modeled on text, showed that knowledge-based approaches are limiting. Humans only capture a subset of concepts and relations in the best case, but often read out less from the text when they approach it with too many preconceived notions.
2) Intermediate representations are necessary but not sufficient: The best intermediate representation is likely to omit important elements needed for accurate formal rule generation. Therefore, source text should be consulted throughout the translation process.
3) Problems of vagueness: Some legal constructs are difficult and vague ("reasonable", "sufficient"), and will probably take some variation or extension of first-order logic to formalize.

Our overall conclusion is that rule extraction is hard but feasible for many cases.

6 Recommendations

The results obtained in this seedling suggest that a program that is focused on carefully selected domains could demonstrate significant increases in efficiency and effectiveness. A program could be successful for domains that:

- Have a large base of regulatory law, as opposed to mostly case law. This is because case law is much harder to understand.
- Have a large number of rules: for training and testing purposes, and for comparative analysis.
- Support a high-level of interaction among the rules: for evaluating the effectiveness of a rule engine in determining compliance.
- Be non-critical for compliance, at least in the early stages, given the difficulty of the problem.
- Be able to procure data that could serve as ground truth, to support evaluation.

Whatever the domain that is ultimately chosen, we believe that focusing on improved parsing technology would lead to greater program effectiveness.

7 References

[1] Carl Andersen, Brett Benyo, Miguel Calejo, Mike Dean, Paul Fodor, Benjamin N. Grosof, Michael Kifer, Senlin Liang, Terrance Swift: Understanding Rulelog Computations in Silk. CoRR abs/1308.4125 (2013)

[2] Tara Athan, H arold Boley, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Wyner, OASIS LegalRuleML, ICAIL 2013, 2013.

[3] Olga Babko-Malaya, Greg P. Milette, Michael K. Schneider, Sarah Scogin: Identifying Nuggets of Information in GALE Distillation Evaluation. LREC 2012: 2322-2327

[4] G. Barabucci, L. Cervone, M. Palmirani, S. Peroni, and F. Vitali, Multi-layer Markup and Ontology Structures in Akoma Ntoso, AICOL Workshop, LNCS 6237, Springer, 2009.

[5] Harold Boley and Michael Kifer: RIF Basic Dialect, Second Edition, 2013.

[6] Alan Buabuchachart, Nina Charness, Katherine Metcalf, Leora Morgenstern: Automated Methods for Extracting and Expanding Lists in Regulatory Text. DoCoPe@JURIX 2013

[7] Alan Buabuchachart, Katherine Metcalf, Nina Charness, Leora Morgenstern: Classification of Regulatory Paragraphs by Discourse Structure, Reference Structure, and Regulation Type. JURIX 2013: 59-62

[8] Eugene Charniak and Yorick Wilks, *Computational Semantics*, Elsevier, 1976.

[9] F. Dell'Orletta, S. Marchi, S. Montemagni, B. Plank, and G. Venturi, The SPLeT-2012

[10] Shared Task on Dependency Parsing of Legal Texts, SPLeT 2012, Workshop on Semantic Processing of Legal Text, at LREC 2012, Istanbul.

[11] JBoss Drools http://drools.jboss.org

[12] Rudolf Flesch: *How to Write Plain English: A Book for Consumers and Lawyers*, Harper Collins, 1979.

[13] Rinke Hoekstra, Joost Breuker, Marcello Di Bello, Alexander Boer: LKIF Core: Principled Ontology Development for the Legal Domain. Law, Ontologies and the Semantic Web 2009: 21-52

[14] T. Joachims, *Learning to Classify Text Using Support Vector Machines*, Springer 2002.

LDC, Linguistic Data Consortium Catalog, 2009. At http://www.ldc.upenn.edu/Catalog.

[15] David McClosky, Sebastian Riedel, Mihai Surdeanu, Andrew McCallum, and Christopher D. Manning: Combining joint models for biomedical event extraction. *BMC Bioinformatics* 13(S-11: S9, 2012.

[16] Christopher D. Manning: Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? CICLing (1) 2011: 171-189

[17] John McCarthy: Applications of Circumscription to Common Sense Reasoning, *Artificial Intelligence*, 1986.

[18] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky: Distant supervision for relation extraction without labeled data, ACL/IJCNLP 2009: 1003-1011

[19] Tom Mitchell, *Machine Learning*, McGraw Hill, 1997.

[20] Leora Morgenstern, Chris Welty, Harold Boley, and Gary Hallmark: RIF Primer, Second edition, 2013

[21] Joe Olive, Caitlin Christianson, and John McCary (eds.), Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation

[22] Adam Wyner and Wim Peters, On Rule Extraction from Regulations, JURIX 2011.

[23] Adam Wyner, Problems and Prospects in the Automated Semantic Analysis of Legal Texts,

[24] SPLeT 2012, Workshop on Semantic Processing of Legal Text, at LREC 2012, Istanbul.

[25] Y. Yang and X. Liu, A re-examination of text categorization methods, Proc. of the 22nd annual Intnl. ACM SIGIR Con. on Research and development in information retrieval, New York, ACM, 1999.

APPENDIX

A.1: Midterm Evaluation Plan

1. Overview

The IARPA seedling TAILCM (Toward Automated International Law Compliance Monitoring) seeks to develop methods to automate the translation of regulatory text to executable rules that can be input into a standard rule engine. TAILCM is comprised of two stages: In the first, an intermediate representation, or ontology, will be extracted from regulatory text; in the second, executable rules will be generated using the intermediate representation.

Two evaluations are planned for TAILCM, which runs for a year (April 2013 through March 2014). An evaluation of the first stage, Ontology Extraction, will take place at the six-month mark, in October 2013; an evaluation of the second stage, Rule Generation, will take place one month before the end of the TAILCM seedling, in February 2014. [Note of May 2014: The Midterm evaluation took place in November and December 2013; the Final Evaluation took place in April 2014.]

This document describes the plan for these evaluations. We begin by listing terms and their definitions. For each evaluation, we describe the training and test materials, the form of the test and test rules; the method for creating the human gold standard; and the methods for scoring system output against the human gold standard. Parts of the plan are in draft form and will be updated as we approach the scheduled evaluations. As of this writing (July 15, 2013), the evaluation plan for the Ontology Extraction stage is more detailed than for the Rule Generation stage.

2. Terms and Definitions

The terms below are defined as they are used in this document. They are ordered by their appearance in this document.

Ontology: A description of the concepts and relationships that exist between these concepts, specifically when this description is used for some application.

Annotation: An attachment of a label to a string in a document or an entire document. Two types of annotation will be described in this document:

> *Concept Annotation:* A label attaching a concept from an ontology to a string in a document or an entire document.
>
> *Executable Rule Annotation:* A label attaching an executable rule or set of executable rules to a string in a document or an entire document.

Ruling Document: A document which reports on a court or agency's ruling regarding a specific case. An example of such a document is FINRA Letter of Acceptance, Waiver, and Consent No.

2009020149901 (http://disciplinaryactions.finra.org/viewdocument.aspx?DocNB=31049), which describes the case of an investment firm, AXA Advisors, one of whose members engaged in several Ponzi schemes.

Regulation: A regulation or order issued by an executive authority (such as the U.S. Congress) or regulatory agency of or regulatory authority associated with a government (such as FINRA, the Financial Industry Regulatory Authority) and having the force of law, taken in its entirety. Examples of regulations are the Securities Act of 1933 (http://www.sec.gov/about/laws/sa33.pdf), the Securities Exchange Act of 1934 (http://www.sec.gov/about/laws/sea34.pdf), and the USA Patriot Act of 2001 (http://www.gpo.gov/fdsys/pkg/PLAW-107publ56/pdf/PLAW-107publ56.pdf). Full regulations frequently run over 100 pages; some (e.g., the Securities Exchange Act) are several hundred pages. However, regulations can be also be one or a few pages long: see, e.g., FINRA Rules 9143, 9144, and 9216. We will refer to a *short regulation* as one that is no more than 15 pages in length.

Regulation Portion: The part of a regulation cited in a ruling document, generally cited because it is the specific portion of a regulation relevant to the application of a judgment in a particular case. Regulation portions are typically referred to by the name of the entire regulation followed by the heading of a particular section. Examples of regulation portions are Securities Exchange Act of 1934 Section 10(b) and Section 3(a)(39)(F), Article III, Section 4 of FINRA's By-laws; these regulation portions are cited by , ruling document FINRA Letter of Acceptance, Waiver, and Consent No. 2009020149901. Regulation portions are often cited in a ruling document when the entire regulation is long.

3. Ontology Extraction

3.1 Training Materials for Ontology Extraction

The training materials for the first stage consist of a corpus, an ontology consisting of a merged upper and lower ontology, and a set of concept annotations on a selection of documents in the training corpus, associating concepts from the ontology with strings in the documents.

3.1.1 Corpus

The training corpus comprises several hundred regulatory documents regarding financial regulation, collected as text files. These documents are divided into two classes as defined above:

(1) A set of 100 *ruling documents*
(2) A set of several hundred *regulations* and *regulation portions* that have been cited by at least one of the above ruling documents.

Most regulations and regulation portions collected for the corpus are between 2 and 6 pages, where each page contains about 500 words. On average, each such document is expected to yield at least several dozen concepts for the ontology described below.

Ruling documents, regulations, and regulation portions are all stored in one folder. However, naming conventions make it easy to distinguish ruling documents from others.

The focus, for at least this first stage, is on ontology extraction from regulations and regulation portions as opposed to ruling documents.

3.1.2 Concept Annotations

The training materials will include, for a subset of the regulations and regulation portions in the corpus, a set of concept annotations, indicating elements of an ontology of financial regulation that can be extracted from the documents. Examples of extracted concepts are *financial instrument*, *securities*, *stock exchange*, and *transaction*.

These annotations will be produced by the TAILCM research team.

3.1.3 Ontology

The TAILCM training ontology will consist of the following:

(1) The TAILCM *upper ontology*, formed by merging parts of LKIF (Legal Knowledge Interchange Format, at http://www.estrellaproject.org/?page_id=5) and FIBO (Financial Business Industry Ontology, at http://www.omg.org/hot-topics/fibo.htm) with other concepts proposed by the TAILCM research team. Examples of concepts included in the upper ontology are *foundation*, *corporation*, *mandate*, *violation*, and *penalty*.
(2) The concept annotations described above, organized into the TAILCM *lower ontology*. For example, if *exchange* and *stock* exchange have both been extracted, *stock exchange* might be represented as a subclass of *exchange.*
(3) An ontology formed by the merging of the TAILCM upper and lower ontologies. The merging process may include the creation of new concepts and relation instances in the ontology. For example, if the actions *embezzle*, *steal*, and *purloin* are extracted from a document, the concept *theft*, which subsumes these actions, might be added as a bridge between the lower ontology concepts and the upper ontology concept *illegal activity*.

3.2 Test Materials for Ontology Extraction

The test materials will consist of

(1) A corpus of ruling documents and regulations and regulation portions, as discussed above. Full regulations will be used only if they are short regulations, as defined above, that is, no more than 15 pages in length. As with the training corpus, ruling documents will be distinguished from regulations and regulation portions by the form of the file names. There will be no overlap between the training corpus and the test corpus. At this stage, the main function of ruling documents is to ensure that regulations and regulation

portions collected are those that are actually used and referenced. Ontology extraction will proceed only on regulations and regulation portions.

The expected size of the test corpus is 10 ruling documents and 5-10 regulation portions and/or short regulations. We expect that several hundred concepts can be extracted from these 5-10 regulation portions and/or short regulations. Note that 10 ruling documents will typically cite several dozen regulation portions. However, it is likely that many of the regulations and regulation portions cited in the 10 ruling documents will have to be discarded because they will already be in the training corpus, that is, they will have already been cited by ruling documents in the training corpus. Thus, it might happen that 10 ruling documents will yield only around 10 new short regulations and regulation portions. If fewer are yielded, additional ruling documents will be collected until 5-10 short regulations and regulation portions that are not already in the training corpus are obtained.

The size and form of the test corpus is chosen to maximize cost effectiveness. It takes a considerable amount of time both to create a human gold standard and to do adjudication, an essential part of scoring, as described below; thus, using a larger test corpus would significantly more resources. However, we believe that the relatively small size of the corpus will not take away from the significance of the results, because even this small set of documents is very productive in terms of concepts for the ontology.

The test corpus will be created by someone who has not been involved in TAILCM research and who has some expertise in the domain of financial regulation.

(2) The TAILCM ontology, consisting of the merged TAILCM upper and lower ontologies, created during the training phase as described above.

3.3 Form of the Ontology Extraction Test

We are counting the concepts that the system extracts from a set of documents, and measuring the ability of the TAILCM system to organize these concepts within an existing ontology. We compare the behavior of the TAILCM system to that of a human with some expertise in the domain of financial regulation.

Formally, the ontology extraction test is characterized by its input and output, as follows:

The input consists of the test materials described above.

The output will consist of:

(1) For each regulation or regulation portion, a set of concepts extracted from that regulation portion, along with provenance for each extracted concept. (It is likely that there will be

multiple provenances for each extracted concept.) Provenance is given as a snippet of text from the regulation portion, and will be expressed in terms of the character positions of the text snippet. As discussed below, the system will not be penalized for giving the wrong provenance for a concept. However, provenance will be useful for at least two reasons:

a. It will help diagnose incorrect recognition of concepts
b. It will help in adjudication of mismatches between concepts recognized by a human and concepts recognized by the system, as explained below.

(2) The enhanced TAILCM ontology, consisting of the input TAILCM ontology to which extracted concepts have been added, and which has been re-organized as needed.

These results will be compared to the concepts and merged ontology generated by the human, and evaluated by another human when system output and human output do not match, as described below in the discussion of the Adjudicated Gold Standard.

3.4 Ontology Extraction Test Rules

The ontology extraction test will be held in October 2013. Prior to test material distribution, the ontology extraction system will be frozen.

During the test, the system will have access to all of the materials, tools, and websites it used during training. This includes but is not limited to concept repositories such as WordNet, ConceptNet, and FrameNet. Access to search engines for certain purposes, e.g., to determine frequency of word combination usage, in order to determine whether a candidate concept for the ontology passes the bar, will also be permitted. The idea is to create a test environment that is as close as possible to the training environment. A full list of sites permitted for access will be given closer to the test date.

3.5 Creation of the Human Gold Standard for Ontology Extraction

The Human Gold Standard will be created by the person who collects the test corpus, someone who has not been involved in TAILCM research but who has some domain expertise in financial regulation. The person will create a list of concepts in regulations and will mark the text from which each concept has been extracted. In addition, this person will relate extracted concepts to one another as well as to the existing TAILCM ontology.

Ideally, the Human Gold Standard would be created through the efforts of at least two people. This would ensure reasonable inter-annotator agreement (IAA). Resources do not permit this. However, we will try to have a second person, also not involved in TAILCM research, annotate a sample of the test corpus so that we can check on IAA.

3.6 Scoring the Ontology Extraction Test

3.6.1 Scoring concept extraction

3.6.1.1 Precision, Recall, and F1 measured against the Human Gold Standard

The basic metrics that will be measured are Precision (P), Recall (R), and F1, the harmonic mean between Precision and Recall.

Assume n regulation portion documents in the test corpus, and assume a canonical ordering $D_1 \ldots D_n$ on these documents. Let D_i be the i^{th} such document. Let $HC(D_i)$ = the set of concepts extracted by the creator of the human gold standard (HGS) for this document. Let $SC(D_i)$ = the set of concepts extracted by the TAILCM system for this document. We define:

$TP(D_i)$, the true positives for D_i = the set of all concepts in both $SC(D_i)$ and $HC(D_i)$

$FP(D_i)$, the false positives for D_i = the set of concepts in $SC(D_i)$ that are not in $HC(D_i)$

$FN(D_i)$, the false negatives for D_i – the set of concepts in $HC(D_i)$ that are not in $SC(D_i)$

Then

$$P(D_i) = TP(D_i) /(TP(D_i) + FP(D_i))$$

$$R(D_i) = TP(D_i) / (TP(D_i) + FN(D_i))$$

$$F1(D_i) = 2*P(D_i)*R(D_i) /(P(D_i) + R(D_i))$$

These formulas can of course be generalized to the set of all concepts. Let $SC(D) = \Sigma_{i=1..n} SC(D_i)$; that is, the set of all concepts. The above definitions of P, R, and D are, as expected:

$$P(D) = TP(D) / (TP(D) + FP(D))$$

$$R(D) = TP(D) / (TP(D) + FN(D))$$

$$F1(D) = 2*P(D)*R(D) / (P(D) + R(D))$$

Note that whether measuring P, R, and F1 for an individual document or for the set of documents, we measure distinct concepts rather than mentions of concepts. Thus, if the concept *transaction* is extracted 5 times in one document, it is still counted as only one concept; similarly if we are measuring across the corpus and the concept is extracted from several documents.

3.6.1.2 Precision, Recall, and F1 measured against the Adjudicated Gold Standard

A simple comparison against a Human Gold Standard has two drawbacks:

1. First, humans make errors, especially in tasks that are not the standard purview of people. It is not necessarily the case that an individual will find all concepts of importance in a particular document. It is possible that an automated system will find concepts that a particular individual will not find, but which, once discovered, are recognized as valid.

2. Second, humans and machines may recognize identical concepts but name them differently. These differences can be relatively minor ---- e.g., naming a concept "broker dealer" rather than "broker/dealer" --- in which case, having a tolerance threshold characterized by a small edit distance would enable recognizing that both human and system have recognized the concept. However, the differences between named extracted concepts can be too large, even if they refer to the same concept. For example, consider the sentence *The electronic storage media must preserve the records in a non-rewriteable, non-erasable format*. The system might recognize *non-rewriteable, non-erasable format* as a concept in the ontology, while a human might express that concept as *permanent format* or *non-modifiable format.* In this case, the system would be penalized for a false positive as well as a false negative, although it appropriately recognized a concept in the ontology.

To deal with these issues, we plan to have False Positives adjudicated by a second person who has also not been engaged in TAILCM research. This human adjudicator builds a new gold standard, known as the Adjudicated Gold Standard (AGS) by starting out with the HGS and then examining all false positives. If the human adjudicator recognizes as valid a concept which has been marked as a false positive --- that is, the concept appears reasonable to the human adjudicator --- that concept will be added to the AGS. P, R, and F1 are then recalculated in terms of the AGS. *Both sets of scores --- P, R, and F1 calculated in terms of the original Human Gold Standard, and P, R, and F1 calculated in terms of the Adjudicated Gold Standard --- will be reported to IARPA.*

Note that to determine whether two different expressions refer to the same concept, the human adjudicator will need to examine the text snippet from which a concept was extracted. This is one reason that we require provenance for each concept extracted.

3.6.2 Scoring the ontology organization

3.6.2.1 Precision, Recall, and F1 measured against the Human Gold Standard ontology

For a set of documents D, the ontology of D is defined by the set of concepts that can be extracted from D, along with a set of instances of binary relations on elements of this concept set. [2] It is assumed that all relations (sets of ordered pairs) are labeled.

To calculate P, R, and F1, the HGS ontology, HGS(O), is first normalized by deleting false positive concepts and suitably modifying relations. E.g., if HGS(O) has concepts A, B, and C with relations A subclass B and B subclass C; and the system ontology, S(O), has concepts A and C with relation A subclass C, the normalization process removes concept B from HGS(O) and replaces the relations A subclass B and B subclass C with A subclass C. (In this case, NHGS(O), the normalized HGS ontology is identical to S(O).) The ontology normalization process is motivated by the preference to focus on similarity of relations rather than similarity of concepts, which is measured separately.

Then:

TP(O), the set of true positives in S(O), is the union, over all i, of the set of all ordered pairs in relation Ri that are in S(O) and also in HGS(O).

FP(O), the set of false positives in F(O), is the union, over all i, of the sets consisting of ordered pairs in relation Ri that are in S(O) but not in HGS(O).

FN(O), the set of false negatives in O, is the union, over all i, of the sets of all ordered pairs in relation Ri that are in HGS(O) but not in S(O).

P, R, and F1 are then defined in the usual way.

3.6.2.2 Precision, Recall, and F1 measured against the Adjudicated Gold Standard ontology

As is the case for concept extraction, humans may not perfectly determine the set of relation instances, and may recognize as valid an instance of a relation that has been posited by a machine. To deal with this, the human adjudicator in charge of adjudicating false positives for extracted concepts will also adjudicate false positives for relation instances. If the human adjudicator recognizes as valid a relation instance which has been marked as a false positive, that relation instance will be added to the AGS. P, R, and F1 are then recalculated in terms of the AGS.

3.6.3 Partial Credit (under development)

A straightforward comparison of the TAILCM-generated ontology against a gold standard, even against the Adjudicated Gold Standard, may exert too high a penalty for mismatches in ontological organization. For example, if the TAILCM ontology incorrectly places a concept in the ontology, the above scoring system will give it no credit, even if it is placed under a node that is relatively close to the node under which it belongs in the AGS. It would seem more reasonable

[2] The restriction to binary relations is assumed for simplicity. In practice, there are likely to be n-ary relations on ontology elements.

to give partial credit for nearly correct answers. One method of doing this is to calculate the distance between the node under which the concept was placed and the node under which the concept should have been placed, similar to the concept of an n^{th} cousin m-times removed. The amount of partial credit should be higher for low values of n and m than for higher values.

In addition, because concepts higher in the ontology are generally considered to be more important than concepts that are deeper in the ontology, we are considering weighting concepts and relation instances by their position in the graph, so that relation instances that occur higher up in the graph are weighted more than those that are lower in the graph.

Schemes for giving partial credit are still in development and will be reported on in more detail in a future version of this document.

4. Rule Generation

The rule generation test is intended to measure the performance of the TAILCM system on constructing a set of executable rules to capture the regulatory force of a regulation or regulation portion.

4.1 Training Materials for Rule Generation

The training materials for the second stage consist of a corpus of documents, an ontology consisting of a merged upper and lower ontology, a set of concept annotations on a selection of documents in the training corpus, labeling parts of the document with concepts from the ontology, and a set of executable rule annotations on the same selection of documents, labeling parts of the document with an executable rule or set of executable rules.

4.1.1 Corpus, Concept Annotations, Ontology

The training corpus, ontology, and concept annotations will include both the training material and the test material (including the Human Gold Standard and the Adjudicated Gold Standard) for the TAILCM ontology extraction stage.

4.1.2 Executable Rule Annotation

The training materials will include, for that subset of the regulations and regulation portions in the corpus for which concept annotations were produced, a set of executable rule annotations, indicating (sets of) executable rules which correspond to portions of these regulations and regulation portions. Each executable rule must be built using standard logical and rule-building operators on only the concepts in the TAILCM ontology.

These annotations will be produced by the TAILCM research team.

4.2 Test Materials for Rule Generation

The test materials will consist of

(1) A corpus of ruling documents and regulations and regulation portions, as discussed above. Full regulations will be used only if they are short regulations, as defined above, that is, no more than 15 pages in length. As with the training corpus, ruling documents will be distinguished from regulations and regulation portions by the form of the file names. There will be no overlap between the training corpus and the test corpus. Rule generation will proceed only on regulations and regulation portions; ruling documents will be used as one of the methods to ensure correctness of the rule generation system.

The expected size of the test corpus is 10 ruling documents and 5-10 regulation portions and/or short regulations. As of this writing (June 15, 2013), we are estimating that each regulation portion will be captured by no fewer than 5 executable rules, but very possibly more. This estimate is rough and preliminary and will probably change during the coming months.

(2) The TAILCM ontology that has been developed in Stage I training and test.

4.3 Form of the Rule Generation Test

We intend to compare the executable rules that TAILCM generates from a set of documents with the executable rules that a human generates from these documents.

The input consists of the test materials described above.

The output will consist of, for each regulation portion, a set of executable rules, along with the provenance for each executable rule. The system will not be penalized for giving wrong provenance for an executable rule. However, provenance will be useful to:

a. help diagnose incorrect generation of rules
b. help adjudicate mismatches between rules generated by humans and rules generated by TAILCM.

It is expected that the TAILCM system will extract new concepts from the test documents and integrate these into the TAILCM ontology and that the executable rules will be constructed from this augmented ontology.

These results will be compared to the human-generated sets of executable rules, and evaluated by another human when system output and human output do not match, as discussed below in the discussion of the Adjudicated Gold Standard.

4.4 Rule Generation Test Rules

The rule generation test will be held in February 2013. Prior to test material distribution, the rule generation system will be frozen.

During the test, the system will have access to all of the materials, tools, and websites it used during training, since the idea is to create a test environment that is as close as possible to the training environment. A full list of sites permitted for access will be given closer to the test date.

4.5 Creation of the Human Gold Standard for Rule Generation

The Human Gold Standard will be created by the person who collects the test corpus, someone who has not been involved in TAILCM research. That person will have some domain expertise in financial regulation as well as expertise in writing executable rules. The person will write sets of executable rules that correspond to regulations or regulation portions and will mark the text from which each executable rule has been generated.

4.6 Scoring the Rule Generation Test

This section will be expanded in the coming months. In general, we plan to measure P, R, and F1 by comparing human and TAILCM system generated output. However, different sets of executable rules may be functionally equivalent, so we need to have some way of measuring functional equivalence. We can reduce the amount of potentially functionally equivalent executable rule sets by placing restrictions on the language used to generate executable rules, as well as on the methods used by both humans and TAILCM to generate the rules. We will explore these approaches during the research for Stage 2.

As with ontology extraction, we plan to compare TAILCM performance against both a Human Gold Standard and an Adjudicated Gold Standard. We will report more details about the construction of the Adjudicated Gold Standard in future versions of this document.

A.2: Midterm Evaluation Results

TAILCM Midterm Evaluation 12/20/2013

The following report shows the results for the performance of Leidos's TAILCM system on the Midterm Evaluation. The Midterm Evaluation task was ontology development: specifically, extracting concepts from a set of financial regulatory documents, organizing them into an ontology, and merging them into an existing upper ontology.

As per discussions with IARPA in October 2013, the TAILCM system ontology was compared to two Human Gold Standards, one developed by Exprentis and one developed by an Leidos ontology expert not involved in TAILCM research. We give metrics for comparisons against both Human Gold Standards.

The metrics reported are Precision (P), Recall (R), and the harmonic mean between Precision and Recall (F1) as defined in the TAILCM Evaluation Plan of July15, 2013, applied to concepts and links.

Definitions:
The following definitions are taken from the TAILCM Evaluation Plan:

Assume n regulation portion documents in the test corpus, and assume a canonical ordering D_1 … D_n on these documents. Let D_i be the i^{th} such document. Let $HC(D_i)$ = the set of concepts extracted by the creator of the human gold standard (HGS) for this document. Let $SC(D_i)$ = the set of concepts extracted by the TAILCM system for this document. We define:
$TP(D_i)$, the true positives for D_i = the set of all concepts in both $SC(D_i)$ and $HC(D_i)$
$FP(D_i)$, the false positives for D_i = the set of concepts in $SC(D_i)$ that are not in $HC(D_i)$
$FN(D_i)$, the false negatives for D_i – the set of concepts in $HC(D_i)$ that are not in $SC(D_i)$
Then
$P(D_i) = TP(D_i) /(TP(D_i) + FP(D_i))$
$R(D_i) = TP(D_i) / (TP(D_i) + FN(D_i))$
$F1(D_i) = 2*P(D_i)*R(D_i) /(P(D_i) + R(D_i))$

False Positive nodes may be *adjudicated* into an Adjudicated Gold Standard by an independent adjudicator --- someone who had not been involved in TAILCM research --- who examines each False Positive node and determines whether the concept belongs in the ontology and can be extracted from the relevant document.

We modified the calculation of P, R, and F for links in a way that resulted in a stricter measurement for TAILCM. Specifically, while the Evaluation Plan allowed for a normalization process that removed False Positive nodes and their links before calculating P, R, and F for links, we keep these links until False Positive nodes are adjudicated. If a False Positive node is adjudicated into the Gold Standard, the links are then adjudicated. If a False Positive node is not adjudicated into the Gold Standard, the links are then removed as discussed in the Evaluation Plan.

Results:
Summary: Recall in general was relatively high, while pre-adjudicated precision was low. Post-adjudicated precision (and thus F1) was much higher that pre-adjudicated precision (and F1). P, R, and F were significantly stronger compared to the Leidos ontology than to the Exprentis ontology.

Pre-Adjudication:

The following table gives pre-adjudication P, R, and F for concepts, compared to the Exprentis HGS.

Document	7 USC 9a	7 USC 13a 2	17 CFR 166 3	FINRA 6460	FINRA 6120	FINRA 5350	FINRA 5220	FINRA 2340
Precision	0.161073826	0.123076923	0.28	0.133928571	0.226804124	0.183333333	0.157894737	0.121621622
Recall	0.615384615	0.324324324	0.368421053	0.428571429	0.458333333	0.423076923	0.405405405	0.317647059
F1	0.255319149	0.178438662	0.318181818	0.204081633	0.303448276	0.255813953	0.227272727	0.175895765

The following table gives pre-adjudication P, R, and F for concepts, compared to the Leidos HGS.

Document	7 USC 9a	7 USC 13a 2	17 CFR 166 3	FINRA 6460	FINRA 6120	FINRA 5350	FINRA 5220	FINRA 2340
Precision	0.362416107	0.395652174	0.592592593	0.426573427	0.467213115	0.452054795	0.491803279	0.511450382
Recall	0.692307692	0.752066116	0.695652174	0.813333333	0.780821918	0.891891892	0.731707317	0.744444444
F1	0.475770925	0.518518519	0.64	0.559633028	0.584615385	0.6	0.588235294	0.606334842

The following table gives pre-adjudication P, R, and F for links, compared to the Exprentis HGS.

Document	7 USC 9a	7 USC 13a 2	17 CFR 166 3	FINRA 6460	FINRA 6120	FINRA 5350	FINRA 5220	FINRA 2340
Precision	0.04040404	0.047169811	0.147058824	0.096153846	0.06097561	0.090909091	0.130434783	0.088785047
Recall	0.307692308	1	1	1	0.714285714	0.833333333	1	0.863636364
F1	0.071428571	0.09009009	0.256410256	0.175438596	0.112359551	0.163934426	0.230769231	0.161016949

The following table gives pre-adjudication P, R, and F for links, compared to the Leidos HGS.

Document	7 USC 9a	7 USC 13a 2	17 CFR 166 3	FINRA 6460	FINRA 6120	FINRA 5350	FINRA 5220	FINRA 2340
Precision	0.346153846	0.424603175	0.657142857	0.473684211	0.477272727	0.441558442	0.536231884	0.474074074
Recall	0.885245902	0.938596491	1	0.972972973	0.913043478	0.971428571	0.986666667	0.920863309
F1	0.497695853	0.584699454	0.793103448	0.637168142	0.626865672	0.607142857	0.694835681	0.625916870

Post-Adjudication:

The following table gives post-adjudication P, R, and F for concepts, compared to the Exprentis HGS.

Document	7 USC 9a	7 USC 13a 2	17 CFR 166 3	FINRA 6460	FINRA 6120	FINRA 5350	FINRA 5220	FINRA 2340
Precision	0.812080537	0.835897436	0.76	0.785714286	0.226804124	0.766666667	0.821052632	0.752252252
Recall	0.889705882	0.765258216	0.612903226	0.814814815	0.458333333	0.754098361	0.78	0.742222222
F1	0.849122807	0.799019608	0.678571429	0.8	0.303448276	0.760330579	0.8	0.747203579

The following table gives post-adjudication P, R, and F for concepts, compared to the Leidos HGS.

Document	7 USC 9a	7 USC 13a 2	17 CFR 166 3	FINRA 6460	FINRA 6120	FINRA 5350	FINRA 5220	FINRA 2340
Precision	0.859060403	0.817391304	1	0.741258741	0.885245902	0.794520548	0.877049180	0.824427481
Recall	0.842105263	0.862385321	0.794117647	0.883333333	0.870967742	0.935483871	0.829457364	0.824427481
F1	0.850498339	0.839285714	0.885245902	0.806083650	0.878048780	0.859259259	0.852589641	0.824427481

The following table gives post-adjudication P, R, and F for links, compared to the Exprentis HGS.

Document	7 USC 9a	7 USC 13a 2	17 CFR 166 3	FINRA 6460	FINRA 6120	FINRA 5350	FINRA 5220	FINRA 2340
Precision	0.297979798	0.367924528	0.382352941	0.586538462	0.060975612	0.581818182	0.663043478	0.556074766
Recall	0.766233766	1	1	1	0.714285714	0.969696971	1	0.975409836
F1	0.429090909	0.537931034	0.553191489	0.739393939	0.112359551	0.727272727	0.797385621	0.708333333

The following table gives post-adjudication P, R, and F for links, compared to the Leidos HGS.

Document	7 USC 9a	7 USC 13a 2	17 CFR 166 3	FINRA 6460	FINRA 6120	FINRA 5350	FINRA 5220	FINRA 2340
Precision	0.631410256	0.734126984	0.914285714	0.700657895	0.727272727	0.681818182	0.793478261	0.724074074
Recall	0.933649289	0.963541667	1	0.98156682	0.941176471	0.981308411	0.990950226	0.946731235
F1	0.75334608	0.833333333	0.955223881	0.817658349	0.820512821	0.804597701	0.881287726	0.820566632

A.3: Final Evaluation Plan

1. Overview

The IARPA seedling TAILCM (Toward Automated International Law Compliance Monitoring) has sought to develop methods to automate the translation of regulatory text to executable rules that can be input into a standard rule engine. TAILCM comprised two stages. In the first stage, an intermediate representation, or ontology, was extracted from regulatory texts. In the second stage, principal components of rules --- *slot fillers* for rule templates --- were extracted from regulatory texts.

This document describes the evaluation plan for TAILCM's second stage, referred to here as the final evaluation, as discussed with and agreed to by IARPA in January 2014, and as subsequently modified, and as discussed with IARPA.

The midterm evaluation plan and results are available in separate documents.

2. Set-up of Test

As was the case with the TAILCM midterm evaluation, the target domain was U.S. financial regulation, and in particular, regulations restricting insider trading and money laundering.

The test was designed to measure the ability of TAILCM software to extract fundamental rule components from regulatory sentences. Examples of regulatory sentences are:

- Alternatively, a member may report a riskless principal transaction by submitting the following (FINRA 6380A)
- If the bank files a SAR pursuant to paragraph c of this section and the suspect is a director or executive officer the bank may not notify the suspect pursuant to 31 U.S.C . 5318 but shall notify all directors who are not suspects. (CFR-2013-title12-vol1-sec21-11)

For each regulatory sentence, TAILCM software tries to determine whether the sentence describes an obligation, prohibition, or permission. Sometimes a sentence contains more than one obligation, prohibition, or permission. TAILCM software tries to extract as many as are contained in the sentence. For each obligation, prohibition, or permission, it attempts to extract:

- The **Type** of regulation: whether it is an obligation, prohibition, or permission
- The **Action** that is obligated, prohibited, or permitted
- The **Agent** of the action, understood to be the cause or initiator of the action
- The **Patient** of the action, understood to be the participant on whom the action is carried out
- The **Conditions** attached to the Action, Agent, or Patient, understood to be circumstances that must be true when the action is carried out, or other conditions attached to the agent or patient
- The **Exceptions** attached to the Action, Agent, or Patient.

In general, not all this information may be present in a regulatory sentence. Following are some examples of the sorts of output that TAILCM extracts:

Example 1: Basic example
FINRA Rule 6380A Ln96

Input: Alternatively, a member may report a riskless principal transaction by submitting the following.

Output:
Type: Permission
Action: report
Agent: member
Patient: riskless principal transaction

This simple example discusses a permitted action, to report. The agent is a member; the patient is a riskless principal transaction. Note that there are no conditions or exceptions. The last four words of the sentence, "by submitting the following", indicate the manner of reporting that is required; information which this evaluation does not measure.

Example 2: Inanimate agent
NFA Rule 2-48 Ln1
Input: The report must contain the data and be in the format prescribed by NFA.

Output:
Type: Obligation
Action: contain
Agent: report
Patient: data

Note that
- Actions need not be all that active: "contain" is considered an action.
- However, the verb "be" is not considered an action.
- An agent is not necessarily human or even animate.

Example 3: Multiple obligations I
CFR-2013-title12-vol1-sec21-11 Ln45-c
Input: Supporting documentation shall be identified and maintained by the bank as such and shall be deemed to have been filed with the SAR.

Output:

Type: Obligation
Action: identified
Agent: Bank
Patient: Supporting documentation

Type: Obligation
Action: maintained
Agent: Bank
Patient: Supporting documentation

Note that
- There are two actions that are obligated: identifying and maintaining
- The agents for the two obligations are identical; the patients for the two obligations are identical
- Both the agent and patient are inanimate
- "Deemed" is not counted as an action for purposes of this evaluation. It generally refers to meta-legal activity, rather than a basic obligation, prohibition, or permission.

Example 4: Multiple Obligations II, Conditions
CFR-2013-title12-vol1-sec21-11 Ln50

Input: If the bank files a SAR pursuant to paragraph c of this section and the suspect is a director or executive officer the bank may not notify the suspect pursuant to 31 U.S.C . 5318 but shall notify all directors who are not suspects.

Note that

- There are two regulations mentioned in this sentence: a prohibition against notifying certain suspects and an obligation to notify other suspects.
- This sentence contains a condition. In general, conditions are restrictions; however, not all restrictions are conditions.
- The condition restricts the action (in this case, notify). In other cases, the condition can depend on the agent or the patient of the action.

Output:

Type: Prohibition
Action: notify
Condition of notify: If the bank files a SAR pursuant to paragraph c of this section and the suspect is a director or executive officer
Agent: bank
Patient: suspect

Type: Obligation
Action: notify
Agent: bank
Patient: directors

Example 5: Exceptions

CFR-2013-title12-vol1-sec21-11 Ln56-c

Input: SAR and any information that would reveal the existence of a SAR are confidential and shall not be disclosed except as authorized in this paragraph k.

Output:

Type: Prohibition
Action: disclosed
Patient: SAR and any information
Exception of disclosed: except as authorized in this paragraph k

Note that as with conditions, exceptions may apply to actions, agents, or patients.

INCORRECT: Example 6

The following is an example of a regulatory sentence where some of the extracted material has been assigned to the wrong category.

FINRA Rule 6380A Ln119
Input: For any transaction in an order for which a member has recording and reporting obligations under Rules 7440 and 7450 the trade report must include an order identifier meeting such parameters as may be prescribed by FINRA assigned to the order that uniquely identifies the order for the date it was received.

Output:

Type: Permission
Action: include
Condition of include: For any transaction in an order for which a member has recording and reporting obligations under Rules 7440 and 7450
Agent: trade report
Patient: order identifier

Type: Permission
Action: prescribed
Patient: such parameters

Note that

- There is one regulation here, an obligation: the trade report must include certain pieces of information. However, another "regulation" has been identified.
- The obligation for the trade order to include an order identifier has been misidentified as a permission.

INCOMPLETE: Example 7

The following is an example of a regulatory sentence where only some of the requested information is extracted. The text from which information has not been extracted is bolded.

MSRB Rule-G-37 Ln14

Input: No broker dealer or municipal securities dealer or any individual designated as a municipal finance professional of the broker dealer or municipal securities dealer pursuant to subparagraphs or of paragraph g of this rule shall solicit any person including but not limited to any affiliated entity of the broker dealer or municipal securities dealer or political action committee to make any payment or shall coordinate any payments to a political party of a state or locality where the broker dealer or municipal securities dealer is engaging or is seeking to engage in municipal securities business .

Output:
Type: Prohibition
Action: Coordinate
Patient: Payments

Note that:

- There are two regulations in this sentence, one a prohibition against soliciting, the other a prohibition against coordinating payments. Only the second prohibition has been extracted.
- The agent of the coordination action (also the agent of the unrecognized solicitation action) has not been extracted.

3. Test sentence selection and processing

3.1 Preprocessing and Selection

The final eval test sentences were chosen from the 250 regulation units originally collected for the TAILCM training corpus as well as the 8 regulation units that comprised the midterm evaluation. Each regulation unit contained multiple sentences. Each regulation unit was first preprocessed using our method for expanding bulleted text (Buabuchachart et al., 2013).

Before designing or developing any software, we randomly divided the set of regulation units of the original TAILCM training corpus into six unequally sized parts. (Five were of roughly equivalent size, and consisted of 10-12 regulation units each; the sixth consisted of all the remaining (nearly 200) regulations.) We restricted ourselves to training on two of the five sets, approximately 10% of the original TAILCM training corpus, leaving three of the five sets, and the sixth (large) set, entirely alone. This was done to ensure that we would not inadvertently train on test data.

Selection of test sentences was performed by a Leidos intern who had not been part of TAILCM's software development team. Selcctions were first made from the Midterm Corpus, then from three of the segregated sets of 10-12 documents each.

A sentence was a candidate for being included in the test corpus if our chosen parser, the LTH semantic parser, did not hang while parsing the sentence --- that is, if it could terminate parsing within a few minutes. Since we knew that the parser (indeed any parser) would not parse most of the sentences correctly, delivering a correct parse was not necessary for being a candidate; just being a correct parse. Sentences were chosen to ensure a representative distribution among our various sources (FINRA, NFA, MSRB, USC, CFR). Sentences were also chosen to have sufficient examples of all types of regulations (obligations, permissions, prohibitions), and of agents, actions, patients, conditions, and exceptions.

Finally, once a large set of sentences was collected, substantial-duplicate sentences were eliminated. Substantial-duplicate sentences occur frequently when bulleted text is expanded, because (possibly multiple) preambles are repeated. For example, the distributed text of CFR 103.19 (Title 31) contains the following sentences which are substantial duplicates of one another. The duplicated part is italicized in both sentences.

(a) General. (2) A transaction requires reporting under the terms of this section if it is conducted or attempted by, at, or through a broker-dealer, it involves or aggregates funds or other assets of at least $5,000, and the broker-dealer knows, suspects, or has reason to suspect that the transaction (or a pattern of transactions of which the transaction is a part): (i) Involves funds derived from illegal activity or is

intended or conducted in order to hide or disguise funds or assets derived from illegal activity (including, without limitation, the ownership, nature, source, location, or control of such funds or assets) as part of a plan to violate or evade any federal law or regulation or to avoid any transaction reporting requirement under federal law or regulation;

a*) General. (2) A transaction requires reporting under the terms of this section if it is conducted or attempted by, at, or through a broker-dealer, it involves or aggregates funds or other assets of at least $5,000, and the broker-dealer knows, suspects, or has reason to suspect that the transaction (or a pattern of transactions of which the transaction is a part):* (ii) Is designed, whether through structuring or other means, to evade any requirements of this part or of any other regulations promulgated under the Bank Secrecy Act, Public Law 91-508, as amended, codified at 12 U.S.C. 1829b, 12 U.S.C. 1951-1959, and 31 U.S.C. 5311-533 2;

We aimed to minimize substantial duplicate sentences because they could skew test results.

3.2 Processing

Any sentence selected for potential inclusion was run through the LTH semantic parser. Two sets of parses were created: the set of parses as returned by LTH, and a set of parses where some light manual corrections were made to correct for some of the mistakes introduced by LTH's underlying dependency parser.

Both sets of parses were then input to TAILCM software, in order to return results for both the corrected and uncorrected parses. (As discussed in the final report, results for corrected and uncorrected parses turned out to be very close.)

4. Form of Test, Evaluation process, and Scoring

4.1 Form of Test

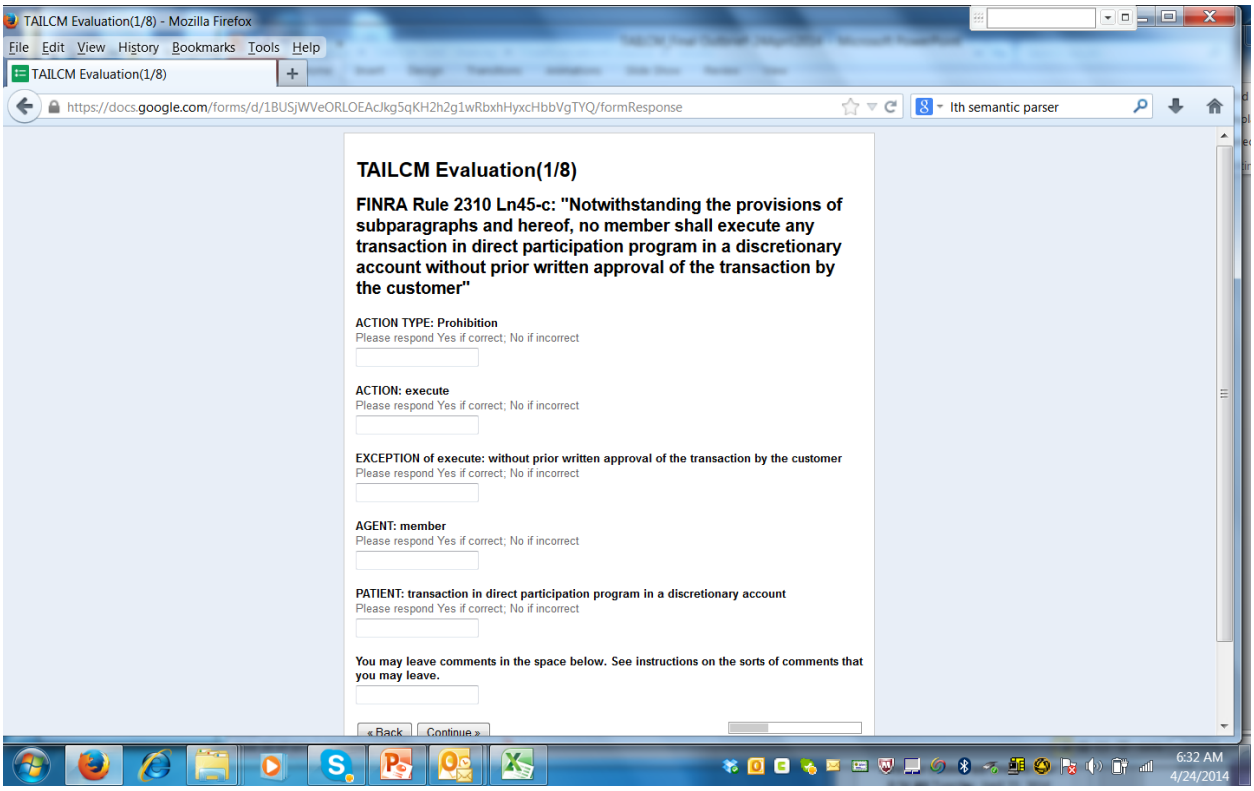
The test consisted of parses of 135 sentences. There were 135 LTH-generated parses, and 128 lightly manually corrected parses of these sentences, for a total of 263 parses. TAILCM software generated 263 sets of output, one for each parse, generating as many regulation types, actions, agents, patients, exceptions, and conditions as possible.

This output was then processed so it could be evaluated by human adjudicators. The 263 sets of output were divided among 8 Google surveys, one set of output per screen. The division was done randomly but according to the following constraint: No survey contained two parses (the corrected and uncorrected parses) for any sentence.

4.2 Evaluation Process

Each screen contained a set of output for an individual parse. Under each generated rule component (regulation type, action, agent, patient, condition, exception) was a box under which an adjudicator could write Yes if the generated rule component was correct and No if it was

incorrect. The adjudicators also had a space --- an extra box --- on the screen to write comments, including supplying any missing components or correcting incorrect components.



Adjudicators had five days to complete the adjudication process, from April 10, 2014 to April 15, 2015. They were asked to complete and submit a survey before going on to the next of the surveys.

There were three adjudicators, two from Leidos, and one from AFRL. Neither Leidos adjudicator was on the TAILCM team. One adjudicator was from outside the division and unknown to anyone on the TAILCM team. Each adjudicator had a training session on a preliminary version of the Google surveys as well as written instructions. Due to an inaccuracy in the written instructions, there was wide variation in how the extra box was used.

4.3 Scoring

4.3.1 Basic Definitions

In a standard evaluation, we draw on the following definitions:

Gold standard: A benchmark set that is taken to be the set of all and only all correct responses on a test, used for evaluating a performance of a system.

True Positive: An element of a test set that matches an element of a predefined gold standard.

False Positive: An element of a test set that does not matches any of the elements of a predefined gold standard.

False Negative: An element of a predefined gold standard that is not matched by any of the elements in a test set.

Precision: A number measuring the accuracy / specificity of a system's output. If TP is the cardinality of the set of True Positives and FP is the cardinality of the set of False Positives, Precision is defined as TP / (TP + FP)

Recall: A number measuring the sensitivity of a system's output. If TP is the cardinality of the set of True Positives and FN is the cardinality of the set of False negatives, Recall is defined as TP / (TP + FN).

F1: The single measure commonly used to measure and compare system performance against some gold standard, defined as the harmonic mean between Precision and Recall. F1 = (2*Precision*Recall) / (Precision + Recall).

4.3.2 Adjudicator's Scoring and the Gold Standard.

These definitions were carried over to the final evaluation plan with two changes:

(1) Instead of using a predefined gold standard, a gold standard was to be defined by the missing rule components provided by the adjudicators.

(2) Precision was calculated both at the individual adjudicator level and by majority vote. E.g., if two of the three adjudicators agreed with TAILCM output, that was considered a True Positive.

Regarding (1): Since the adjudicators did not consistently specify missing rule components, the Leidos TAILCM team instead created a gold standard for about 25% of the sentences.

Precision, Recall, and F1 were then calculated as defined above. Results are reported in the final

A.4: Final Evaluation Results

1. Overview and Results Summary

The TAILCM final evaluation consisted of determining the correctness of extracting rule components from sentences of regulatory text, slot fillers for rule templates. The rule components that could be extracted for each regulatory sentence were:

- The **Type** of regulation: whether it is an obligation, prohibition, or permission
- The **Action** that is obligated, prohibited, or permitted
- The **Agent** of the action, understood to be the cause or initiator of the action
- The **Patient** of the action, understood to be the participant on whom the action is carried out
- The **Conditions** attached to the Action, Agent, or Patient, understood to be circumstances that must be true when the action is carried out, or other conditions attached to the agent or patient
- The **Exceptions** attached to the Action, Agent, or Patient.

The metrics reported are Precision (P), Recall (R), and the harmonic mean between Precision and Recall (F1) as defined in the TAILCM Final Evaluation Plan.

Results in general were quite strong. Moreover, there was virtually no difference in performance between corrected and uncorrected parses. Average precision ranged from .88 (uncorrected parses) to .90 (corrected parses). Recall computed on a random set of sentences (not computed on the entire set due to resource limitations) was .76. Extrapolated F1 (based on Precision over all sentences and the Recall on the random set) was .82.

2. Detailed Breakdown of Test Sentences

The evaluation was conducted on 263 sets of output corresponding to 135 regulatory sentences from the financial regulatory domain, concerning insider trading and money laundering, were evaluated by three adjudicators. These sets of output were based on 135 sets of uncorrected parses of the regulatory sentences, produced by the LTH semantic parser and 128 sets of parses in which light corrections, generally concerning the dependency of prepositional phrases, were manually made on the 135 parses.

Altogether, the TAILCM software generated 1355 rule components for these 263 parses. Often a parse was missing some rule component (e.g., condition, exception, patient); in a few cases, the software did not generate any rule component for a parse.

Each rule component was considered a question: adjudicators had to determine whether or not the rule component as extracted was correct for that sentence.

The question distribution is shown below

Question type	**Percentage of total questions**
Regulation Type	25
Action	25
Agent	19
Patient	19
Condition	8
Exception	4

The identical numbers for regulation type and action is not coincidental. The software did not consider a sentence to state an actual rule unless it could determine the deontic mood (obligated, permitted, or prohibited) and the action so obligated, or permitted, or prohibited. The identical numbers for agent and patient is coincidental. Often, both agent and patient were present, but sometimes one or the other was missing. A different set of sentences would likely not have an identical percentage for agent and patient.

3. Detailed Results

The following table gives average precision for each of the question categories, for both uncorrected and corrected parses:

CORRECTED	Precision (majority vote)	Precision (averaged over adjudicators)
Regulation Type	0.96	0.94
Action	0.96	0.9
Agent	0.86	0.84
Patient	0.82	0.72
Condition	0.96	0.75
Exception	0.86	0.9
Total	0.9	0.85
UNCORRECTED		
Regulation Type	0.93	0.95
Action	0.94	0.94
Agent	0.92	0.86
Patient	0.83	0.76
Condition	0.84	0.73
Exception	0.79	0.84
Total	0.9	0.88

The following table gives precision for each of the adjudicators and each of the categories:

CORRECTED	A1 Correct	A1 Incorrect	Precision	A3 Correct	A3 Incorrect	Precision	A2 Correct	A2 Incorrect	Precision
Regulation Type	170	4	**0.977011**	159	15	**0.913793**	163	11	**0.936782**
Action	167	7	**0.95977**	141	33	**0.810345**	163	11	**0.936782**
Agent	109	18	**0.858268**	103	24	**0.811024**	109	18	**0.858268**
Patient	85	45	**0.653846**	88	42	**0.676923**	107	23	**0.823077**
Condition	40	15	**0.727273**	38	17	**0.690909**	46	9	**0.836364**
Exception	21	3	**0.875**	23	1	**0.958333**	21	3	**0.875**
Total	592	92	**0.865497**	552	132	**0.807018**	609	75	**0.890351**

UNCORRECTED			Precision			Precision			Precision
Regulation Type	167	9	**0.948864**	156	20	**0.886364**	164	12	**0.931818**
Action	164	11	**0.937143**	129	47	**0.732955**	165	11	**0.9375**
Agent	109	16	**0.872**	97	28	**0.776**	115	10	**0.92**
Patient	81	50	**0.618321**	86	46	**0.651515**	110	22	**0.833333**
Condition	31	14	**0.688889**	30	16	**0.652174**	37	7	**0.840909**
Exception	16	3	**0.842105**	13	6	**0.684211**	15	4	**0.789474**
Total	568	103	**0.846498**	511	163	**0.75816**	606	66	**0.901786**

The following table gives Recall computed on Q1 and extrapolated F1, based on average precision, for each of the categories

	Number in gold standard (Q1)	Found by software (Q1)	Recall (Q1)	Precision over all questionnaires (Q1-Q8) (averaged corrected and uncorrected)	Estimated F1 (Q1 – Q8) (extrapolating recall from Q1 to all)
Actions	**51**	**46**	**.90**	**.95**	**.925**
Agents	**47**	**40**	**.85**	**.86**	**.86**
Patients	**47**	**31**	**.66**	**.79**	**.72**
Conditions	**18**	**6**	**.33**	**.77**	**.47**
Exceptions	**8**	**7**	**.87**	**.90**	**.89**
Total items	**171**	**130**	**.76**	**.89**	**.82**

The following table gives the breakdown of answer for each adjudicator:

	A1	A1	A1	A3	A3	A3	A2	A2	A2
File (-c is corrected)	Correct	Incorrect	Total Answered	Correct	Incorrect	Total Answered	Correct	Incorrect	Total Answered
17 CFR 166 3 Ln0	5	1	6	1	5	6	5	1	6
17 CFR 166 3 Ln0-c	4	1	5	4	1	5	5	0	5
7 USC 13a 2 Ln1	3	1	4	1	3	4	3	1	4
7 USC 13a 2 Ln10	4	0	4	4	0	4	4	0	4
7 USC 13a 2 Ln10-c	8	0	8	6	2	8	8	0	8
7 USC 13a 2 Ln11	9	0	9	9	0	9	8	1	9
7 USC 13a 2 Ln11-c	5	0	5	5	0	5	5	0	5
7 USC 13a 2 Ln12	5	1	6	6	0	6	6	0	6
7 USC 13a 2 Ln12-c			0			0			0
7 USC 13a 2 Ln13	3	1	4	1	3	4	4	0	4
7 USC 13a 2 Ln13-c	4	2	6	6	0	6	6	0	6
7 USC 13a 2 Ln14	4	0	4	3	1	4	4	0	4
7 USC 13a 2 Ln14-c	7	1	8	8	0	8	8	0	8
7 USC 13a 2 Ln1-c	4	0	4	4	0	4	4	0	4
7 USC 13a 2 Ln2			0			0			0
7 USC 13a 2 Ln2-c			0			0			0
7 USC 13a 2 Ln5	8	2	10	8	2	10	9	1	10

7 USC 13a 2 Ln5-c	4	1	5	5	0	5	5	0	5
7 USC 13a 2 Ln6	5	0	5	5	0	5	5	0	5
7 USC 13a 2 Ln6-c	7	0	7	3	4	7	7	0	7
7 USC 9a Ln3	7	3	10	10	0	10	5	5	10
7 USC 9a Ln3-c	7	1	8	4	4	8	8	0	8
CFR 103.176 Ln1	6	0	6	5	1	6	6	0	6
CFR 103.176 Ln1-c	4	0	4	4	0	4	4	0	4
CFR 103.176 Ln28	2	0	2	1	1	2	2	0	2
CFR 103.176 Ln28-c	3	2	5	0	5	5	5	0	5
CFR 103.176 Ln47			0			0			0
CFR 103.176 Ln47-c	5	0	5	5	0	5	5	0	5
CFR 12 21.11 Ln56	4	0	4	4	0	4	4	0	4
CFR 12 21.11 Ln56-c	5	2	7	1	6	7	6	1	7
CFR 12 21.11 Ln75	4	0	4	4	0	4	4	0	4
CFR 12 21.11 Ln75-c	13	0	13	13	0	13	13	0	13
FINRA By-law Article V sec 3 Ln0	12	1	13	11	2	13	13	0	13
FINRA By-law Article V sec 3 Ln0-c	7	0	7	7	0	7	6	1	7
FINRA By-law Article V sec 3 Ln3	3	0	3	3	0	3	3	0	3
FINRA By-law Article V sec 3 Ln3-c	5	0	5	5	0	5	5	0	5
FINRA Rule 2310 Ln116	3	1	4	4	0	4	4	0	4
FINRA Rule 2310 Ln116-c	3	1	4	4	0	4	4	0	4
FINRA Rule 2310 Ln118	3	1	4	3	1	4	4	0	4
FINRA Rule 2310 Ln118-c	4	0	4	4	0	4	4	0	4
FINRA Rule 2310 Ln122	5	0	5	5	0	5	5	0	5
FINRA Rule 2310 Ln122-c			0			0			0
FINRA Rule 2310 Ln126	5	0	5	5	0	5	5	0	5
FINRA Rule 2310 Ln126-c	2	0	2	2	0	2	2	0	2
FINRA Rule 2310 Ln221	3	0	3	3	0	3	3	0	3
FINRA Rule 2310 Ln221-c			0			0			0
FINRA Rule 2310 Ln234	4	0	4	4	0	4	4	0	4
FINRA Rule 2310 Ln234-c	4	0	4	4	0	4	4	0	4
FINRA Rule 2310 Ln240	5	0	5	5	0	5	5	0	5
FINRA Rule 2310 Ln240-c	0	4	4	1	3	4	0	4	4
FINRA Rule 2310 Ln36	6	1	7	4	3	7	6	1	7
FINRA Rule 2310 Ln36-c	5	1	6	6	0	6	3	3	6
FINRA Rule 2310 Ln38	10	0	10	10	0	10	10	0	10
FINRA Rule 2310 Ln38-c	4	0	4	1	3	4	4	0	4
FINRA Rule 2310 Ln43	7	1	8	8	0	8	7	1	8
FINRA Rule 2310 Ln43-c			0			0			0
FINRA Rule 2310 Ln45	6	0	6	5	1	6	6	0	6
FINRA Rule 2310 Ln45-c			0			0			0

FINRA Rule 2310 Ln50	5	0	5	5	0	5	5	0	5
FINRA Rule 2310 Ln50-c	3	2	5	4	1	5	4	1	5
FINRA Rule 2310 Ln57	6	0	6	6	0	6	6	0	6
FINRA Rule 2310 Ln57-c	3	0	3	3	0	3	3	0	3
FINRA Rule 2310 Ln61	3	1	4	3	1	4	4	0	4
FINRA Rule 2310 Ln61-c	4	2	6	5	1	6	6	0	6
FINRA Rule 2310 Ln63	4	0	4	4	0	4	4	0	4
FINRA Rule 2310 Ln63-c	11	0	11	9	2	11	11	0	11
FINRA Rule 2310 Ln66	2	2	4	4	0	4	4	0	4
FINRA Rule 2310 Ln66-c	3	0	3	3	0	3	3	0	3
FINRA Rule 2310 Ln7	11	0	11	6	5	11	7	4	11
FINRA Rule 2310 Ln7-c	3	2	5	5	0	5	5	0	5
FINRA Rule 2360 Ln281			0			0			0
FINRA Rule 2360 Ln281-c	2	2	4	2	2	4	3	1	4
FINRA Rule 2360 Ln343	6	0	6	6	0	6	6	0	6
FINRA Rule 2360 Ln343-c	10	0	10	10	0	10	10	0	10
FINRA Rule 5220 Ln0	2	3	5	5	0	5	5	0	5
FINRA Rule 5220 Ln0-c	2	0	2	2	0	2	2	0	2
FINRA Rule 5350 Ln0	11	1	12	10	2	12	10	2	12
FINRA Rule 5350 Ln0-c	6	0	6	6	0	6	6	0	6
FINRA Rule 6120 Ln2	3	0	3	3	0	3	3	0	3
FINRA Rule 6120 Ln2-c	5	0	5	5	0	5	5	0	5
FINRA Rule 6120 Ln4			0			0			0
FINRA Rule 6120 Ln4-c	3	2	5	5	0	5	5	0	5
FINRA Rule 6120 Ln8	4	0	4	4	0	4	4	0	4
FINRA Rule 6120 Ln8-c	2	1	3	1	2	3	2	1	3
FINRA Rule 6250 Ln0	2	1	3	1	2	3	2	1	3
FINRA Rule 6250 Ln0-c	4	0	4	1	3	4	4	0	4
FINRA Rule 6250 Ln103			0			0			0
FINRA Rule 6250 Ln103-c	11	1	12	11	1	12	12	0	12
FINRA Rule 6250 Ln113	2	1	3	2	1	3	2	1	3
FINRA Rule 6250 Ln113-c	2	0	2	1	1	2	2	0	2
FINRA Rule 6250 Ln118	3	4	7	4	3	7	4	3	7
FINRA Rule 6250 Ln118-c	7	1	8	8	0	8	8	0	8
FINRA Rule 6250 Ln123	3	1	4	4	0	4	4	0	4
FINRA Rule 6250 Ln123-c	7	0	7	6	1	7	6	1	7
FINRA Rule 6250 Ln128	3	1	4	4	0	4	4	0	4
FINRA Rule 6250 Ln128-c	7	0	7	6	1	7	6	1	7
FINRA Rule 6250 Ln131	5	5	10	9	1	10	6	4	10
FINRA Rule 6250 Ln131-c	12	0	12	10	2	12	12	0	12
FINRA Rule 6250 Ln142			0			0			0
FINRA Rule 6250 Ln142-c	4	0	4	4	0	4	4	0	4
FINRA Rule 6250 Ln145	4	0	4	4	0	4	4	0	4

FINRA Rule 6250 Ln145-c	4	0	4	4	0	4	4	0	4
FINRA Rule 6250 Ln4	5	0	5	5	0	5	2	3	5
FINRA Rule 6250 Ln40	4	1	5	4	1	5	5	0	5
FINRA Rule 6250 Ln40-c	4	1	5	5	0	5	4	1	5
FINRA Rule 6250 Ln44	4	0	4	4	0	4	4	0	4
FINRA Rule 6250 Ln44-c	8	3	11	9	2	11	11	0	11
FINRA Rule 6250 Ln4-c			0			0			0
FINRA Rule 6250 Ln54			0			0			0
FINRA Rule 6250 Ln54-c	7	1	8	5	3	8	6	2	8
FINRA Rule 6250 Ln55	2	1	3	3	0	3	3	0	3
FINRA Rule 6250 Ln55-c	5	0	5	5	0	5	5	0	5
FINRA Rule 6250 Ln56	11	0	11	11	0	11	10	1	11
FINRA Rule 6250 Ln56-c	3	0	3	3	0	3	3	0	3
FINRA Rule 6250 Ln58	3	1	4	3	1	4	4	0	4
FINRA Rule 6250 Ln58-c	9	1	10	9	1	10	10	0	10
FINRA Rule 6250 Ln59	4	0	4	4	0	4	4	0	4
FINRA Rule 6250 Ln59-c	3	1	4	4	0	4	4	0	4
FINRA Rule 6250 Ln6	8	0	8	1	7	8	8	0	8
FINRA Rule 6250 Ln64	4	1	5	4	1	5	5	0	5
FINRA Rule 6250 Ln64-c	4	0	4	4	0	4	4	0	4
FINRA Rule 6250 Ln6-c	4	0	4	4	0	4	4	0	4
FINRA Rule 6250 Ln70	2	1	3	1	2	3	2	1	3
FINRA Rule 6250 Ln70-c	3	4	7	5	2	7	3	4	7
FINRA Rule 6250 Ln71			0			0			0
FINRA Rule 6250 Ln71-c	3	0	3	3	0	3	3	0	3
FINRA Rule 6250 Ln79	3	0	3	3	0	3	3	0	3
FINRA Rule 6250 Ln79-c	4	0	4	0	4	4	4	0	4
FINRA Rule 6250 Ln83	5	0	5	5	0	5	5	0	5
FINRA Rule 6250 Ln83-c	2	1	3	1	2	3	2	1	3
FINRA Rule 6250 Ln84	5	1	6	5	1	6	6	0	6
FINRA Rule 6250 Ln84-c	9	1	10	4	6	10	9	1	10
FINRA Rule 6250 Ln85	3	2	5	5	0	5	5	0	5
FINRA Rule 6250 Ln85-c	4	0	4	4	0	4	4	0	4
FINRA Rule 6250 Ln90	5	0	5	5	0	5	5	0	5
FINRA Rule 6250 Ln90-c	6	1	7	7	0	7	6	1	7
FINRA Rule 6250 Ln96	7	0	7	7	0	7	7	0	7
FINRA Rule 6250 Ln96-c	4	0	4	4	0	4	4	0	4
FINRA Rule 6624 Ln0			0			0			0
FINRA Rule 6624 Ln0-c			0			0			0
FINRA Rule 9348 Ln0	10	0	10	10	0	10	10	0	10
FINRA Rule 9348 Ln0-c	5	0	5	5	0	5	5	0	5
FINRA Rule 9348 Ln1			0			0			0
FINRA Rule 9348 Ln1-c	3	0	3	3	0	3	3	0	3

MSRB Rule G-27 Ln117	5	0	5	5	0	5	5	0	5
MSRB Rule G-27 Ln117-c	8	1	9	3	6	9	4	5	9
MSRB Rule G-27 Ln120	2	3	5	3	2	5	5	0	5
MSRB Rule G-27 Ln120-c			0			0			0
MSRB Rule G-27 Ln122	6	1	7	2	5	7	7	0	7
MSRB Rule G-27 Ln122-c	5	2	7	7	0	7	6	1	7
MSRB Rule G-27 Ln129	5	4	9	6	3	9	8	1	9
MSRB Rule G-27 Ln129-c	4	2	6	5	1	6	6	0	6
MSRB Rule G-27 Ln133	3	1	4	4	0	4	4	0	4
MSRB Rule G-27 Ln133-c			0			0			0
MSRB Rule G-27 Ln134	8	1	9	3	6	9	5	4	9
MSRB Rule G-27 Ln134-c	8	0	8	5	3	8	8	0	8
MSRB Rule G-27 Ln140	3	0	3	3	0	3	2	1	3
MSRB Rule G-27 Ln140-c	6	0	6	6	0	6	6	0	6
MSRB Rule G-27 Ln143	2	2	4	4	0	4	4	0	4
MSRB Rule G-27 Ln143-c	2	3	5	2	3	5	2	3	5
MSRB Rule G-27 Ln145	7	0	7	5	2	7	7	0	7
MSRB Rule G-27 Ln145-c	4	0	4	3	1	4	4	0	4
MSRB Rule G-27 Ln151	4	1	5	5	0	5	2	3	5
MSRB Rule G-27 Ln151-c	2	0	2	2	1	3	3	0	3
MSRB Rule G-27 Ln152			0			0			0
MSRB Rule G-27 Ln152-c	2	0	2	1	1	2	2	0	2
MSRB Rule G-27 Ln155	6	0	6	6	0	6	6	0	6
MSRB Rule G-27 Ln155-c	4	1	5	3	2	5	5	0	5
MSRB Rule G-27 Ln156	3	1	4	3	1	4	3	1	4
MSRB Rule G-27 Ln156-c	6	1	7	3	4	7	3	4	7
MSRB Rule G-27 Ln158	4	0	4	3	1	4	4	0	4
MSRB Rule G-27 Ln158-c	3	1	4	4	0	4	4	0	4
MSRB Rule G-27 Ln159	4	1	5	4	1	5	5	0	5
MSRB Rule G-27 Ln159-c	5	0	5	5	0	5	5	0	5
MSRB Rule G-27 Ln160	8	4	12	5	7	12	10	2	12
MSRB Rule G-27 Ln160-c	4	0	4	4	0	4	4	0	4
MSRB Rule G-27 Ln165	5	0	5	4	1	5	5	0	5
MSRB Rule G-27 Ln165-c	6	1	7	6	1	7	7	0	7
MSRB Rule G-27 Ln172	3	0	3	3	0	3	3	0	3
MSRB Rule G-27 Ln172-c	8	2	10	0	10	10	5	5	10
MSRB Rule G-27 Ln178	4	1	5	3	2	5	5	0	5
MSRB Rule G-27 Ln178-c			0			0			0
MSRB Rule G-27 Ln183	5	0	5	5	0	5	4	1	5
MSRB Rule G-27 Ln183-c	3	1	4	3	1	4	4	0	4
MSRB Rule G-27 Ln19	4	0	4	4	0	4	4	0	4
MSRB Rule G-27 Ln191	3	1	4	4	0	4	4	0	4
MSRB Rule G-27 Ln191-c	5	0	5	5	0	5	5	0	5

MSRB Rule G-27 Ln192	8	0	8	4	4	8	8	0	8
MSRB Rule G-27 Ln192-c	3	3	6	5	1	6	6	0	6
MSRB Rule G-27 Ln193	5	0	5	5	0	5	5	0	5
MSRB Rule G-27 Ln193-c			0			0			0
MSRB Rule G-27 Ln194	5	0	5	5	0	5	4	1	5
MSRB Rule G-27 Ln194-c	4	2	6	6	0	6	6	0	6
MSRB Rule G-27 Ln196			0			0			0
MSRB Rule G-27 Ln196-c	2	1	3	2	1	3	2	1	3
MSRB Rule G-27 Ln197	5	0	5	5	0	5	5	0	5
MSRB Rule G-27 Ln197-c	3	1	4	3	1	4	3	1	4
MSRB Rule G-27 Ln19-c	5	0	5	5	0	5	5	0	5
MSRB Rule G-27 Ln200			0			0			0
MSRB Rule G-27 Ln200-c	5	0	5	5	0	5	5	0	5
MSRB Rule G-27 Ln202	5	1	6	2	4	6	5	1	6
MSRB Rule G-27 Ln202-c	4	0	4	3	1	4	4	0	4
MSRB Rule G-27 Ln205	8	2	10	6	4	10	9	1	10
MSRB Rule G-27 Ln205-c	5	0	5	4	1	5	4	1	5
MSRB Rule G-27 Ln210	7	0	7	7	0	7	7	0	7
MSRB Rule G-27 Ln210-c	4	0	4	4	0	4	4	0	4
MSRB Rule G-27 Ln213	7	0	7	6	1	7	6	1	7
MSRB Rule G-27 Ln213-c	5	0	5	5	0	5	5	0	5
MSRB Rule G-27 Ln214	3	1	4	4	0	4	4	0	4
MSRB Rule G-27 Ln214-c	7	0	7	7	0	7	7	0	7
MSRB Rule G-27 Ln218	5	0	5	5	0	5	5	0	5
MSRB Rule G-27 Ln218-c	4	0	4	4	0	4	4	0	4
MSRB Rule G-27 Ln219	1	2	3	0	3	3	2	1	3
MSRB Rule G-27 Ln219-c	7	0	7	6	1	7	7	0	7
MSRB Rule G-27 Ln227	3	0	3	0	3	3	2	1	3
MSRB Rule G-27 Ln227-c	3	1	4	2	2	4	4	0	4
MSRB Rule G-27 Ln243	2	1	3	2	1	3	3	0	3
MSRB Rule G-27 Ln243-c			0			0			0
MSRB Rule G-27 Ln26	5	0	5	3	2	5	5	0	5
MSRB Rule G-27 Ln26-c	3	0	3	3	0	3	3	0	3
MSRB Rule G-27 Ln3	4	0	4	4	0	4	4	0	4
MSRB Rule G-27 Ln33	6	4	10	9	1	10	6	4	10
MSRB Rule G-27 Ln33-c	15	1	16	1	15	16	15	1	16
MSRB Rule G-27 Ln3-c	3	1	4	4	0	4	4	0	4
MSRB Rule G-27 Ln40	3	0	3	1	2	3	3	0	3
MSRB Rule G-27 Ln40-c	3	0	3	3	0	3	3	0	3
MSRB Rule G-27 Ln5	4	0	4	4	0	4	4	0	4
MSRB Rule G-27 Ln59	8	1	9	6	3	9	9	0	9
MSRB Rule G-27 Ln59-c	8	3	11	2	9	11	8	3	11
MSRB Rule G-27 Ln5-c	7	0	7	7	0	7	7	0	7

MSRB Rule G-27 Ln62	2	0	2	1	1	2	2	0	2
MSRB Rule G-27 Ln62-c	5	1	6	5	1	6	6	0	6
MSRB Rule G-27 Ln92	8	0	8	8	0	8	7	1	8
MSRB Rule G-27 Ln92-c	3	1	4	4	0	4	4	0	4
MSRB Rule G-27 Ln95	5	3	8	3	7	10	6	4	10
MSRB Rule G-27 Ln95-c	4	0	4	3	1	4	4	0	4
MSRB Rule G-27 Ln99	2	1	3	2	1	3	0	3	3
MSRB Rule G-27 Ln99-c	3	0	3	3	0	3	3	0	3
MSRB Rule G-3 Ln151	2	3	5	5	0	5	5	0	5
MSRB Rule G-3 Ln151-c			0			0			0
MSRB Rule G-3 Ln68	3	1	4	1	3	4	3	1	4
MSRB Rule G-3 Ln68-c	2	2	4	2	2	4	2	2	4
MSRB Rule-G-2 Ln0	3	4	7	3	4	7	4	3	7
MSRB Rule-G-2 Ln0-c	4	0	4	4	0	4	4	0	4
MSRB Rule-G-41 Ln0	4	0	4	4	0	4	4	0	4
MSRB Rule-G-41 Ln0-c	4	0	4	4	0	4	4	0	4
NASD Rule 2340 Ln0	4	1	5	2	3	5	5	0	5
NASD Rule 2340 Ln0-c	3	1	4	2	2	4	3	1	4
NASD Rule 2340 Ln10	6	1	7	4	3	7	6	1	7
NASD Rule 2340 Ln10-c	5	3	8	2	6	8	6	2	8
NASD Rule 2340 Ln13			0			0			0
NASD Rule 2340 Ln13-c	4	0	4	4	0	4	4	0	4
NASD Rule 2340 Ln27			0			0			0
NASD Rule 2340 Ln27-c	4	0	4	4	0	4	4	0	4
NFA Rule 2-25 Ln0	7	1	8	5	3	8	5	3	8
NFA Rule 2-25 Ln0-c	4	0	4	4	0	4	4	0	4
NFA Rule 2-25 Ln1	3	0	3	1	2	3	2	1	3
NFA Rule 2-25 Ln1-c	2	1	3	2	1	3	2	1	3
NFA Rule 2-4 Ln0	4	1	5	5	0	5	5	0	5
NFA Rule 2-4 Ln0-c	1	3	4	4	0	4	3	0	3
NFA Rule 2-6 Ln0			0			0			0
NFA Rule 2-6 Ln0-c	3	1	4	2	2	4	3	1	4
NFA Rule 2-6 Ln1	3	1	4	4	0	4	4	0	4
NFA Rule 2-6 Ln1-c			0			0			0
NFA Rule 3-18 Ln0	4	2	6	6	0	6	6	0	6
NFA Rule 3-18 Ln0-c	4	1	5	5	0	5	5	0	5
NFA Rule 3-18 Ln3			0			0			0
NFA Rule 3-18 Ln3-c	4	4	8	7	1	8	4	4	8
NFA Rule 3-18 Ln6	4	0	4	4	0	4	4	0	4
NFA Rule 3-18 Ln6-c	4	0	4	4	0	4	4	0	4
TOTAL:	1124	183	1307	1026	284	1310	1178	131	1309

Glossary

Agent: The individual or organization that performs an action (grammatical role in sentence)

Annotation: An attachment of a label to a string in a document or an entire document.

Captcha: A program, generally used for protecting websites against bots, that generates and grades tests that humans usually can pass and bots currently cannot pass. These tests often involve reading of partially distorted images of letters.

Deontic: expressing obligation, or more generally, obligation, permission, and prohibition. Examples of deontic operators include should, must, may, cannot and shall not.

Executable rule: A rule that can be processed by a rule engine.

F1: The single measure commonly used to measure and compare system performance against some gold standard, defined as the harmonic mean between Precision and Recall. F1 = (2*Precision*Recall) / (Precision + Recall).

False Negative: An element of a predefined gold standard that is not matched by any of the elements in a test set.

False Positive: An element of a test set that does not matches any of the elements of a predefined gold standard.

Gold standard: A benchmark set that is taken to be the set of all and only all correct responses on a test, used for evaluating a performance of a system.

Irrealis: Characteristic of a grammatic mood, indicating that what is being written or said is not believed to be true or have happened. Regulatory text is typically in the irrealis mood.

Natural language processing: A set of techniques for understanding spoken and written human language; the act of applying these techniques to some text.

Ontology: A description of the concepts and relationships that exist between these concepts, specifically when this description is used for some application.

Patient: The entity on which an action is performed (grammatical role in sentence)

Precision: A number measuring the accuracy / specificity of a system's output. If TP is the cardinality of the set of True Positives and FP is the cardinality of the set of False Positives, Precision is defined as TP / (TP + FP)

Recall: A number measuring the sensitivity of a system's output. If TP is the cardinality of the set of True Positives and FN is the cardinality of the set of False negatives, Recall is defined as TP / (TP + FN).

Regular expression: A pattern of words or other symbols defined using the Kleene operators of concatenation, repetition, and disjunction, frequently used for searching and/or classification.

Regulation: A regulation or order issued by an executive authority (such as the U.S. Congress) or regulatory agency of or regulatory authority associated with a government (such as FINRA, the Financial Industry Regulatory Authority) and having the force of law, taken in its entirety. Examples of regulations are the Securities Act of 1933 (http://www.sec.gov/about/laws/sa33.pdf), the Securities Exchange Act of 1934 **(**http://www.sec.gov/about/laws/sea34.pdf**),** and the USA Patriot Act of 2001 (http://www.gpo.gov/fdsys/pkg/PLAW-107publ56/pdf/PLAW-107publ56.pdf**).**

Regulation unit: A short regulation (generally less than ten pages) or portion of a longer regulation

Rule: sentence in formal logic that has an antecedent (IF part) and consequent (THEN part).

Rule engine: A software system that processes rules written in a formal language of the form IF <condition> THEN <action>, generally by matching the <condition> with a set of facts describing some situation and executing the corresponding <action>

Ruling Document: Document which reports on a court or agency's ruling regarding a specific case. An example of such a document is FINRA Letter of Acceptance, Waiver, and Consent No. 2009020149901 **(**http://disciplinaryactions.finra.org/viewdocument.aspx?DocNB=31049), which describes the case of an investment firm, AXA Advisors, one of whose members engaged in several Ponzi schemes.

Semantic Parser: Parsing program intended to recognize semantic roles of sentence constituents and in particular to recognize actions, agents, and patients of a sentence.

Support vector machines: A set of supervised learning methods for recognizing patterns, used for classification of data.

True Positive: An element of a test set that matches an element of a predefined gold standard.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AI: Artificial Intelligence

CAPTCHA, captcha: Completely Automated Public Turing test to tell Computers and Humans Apart)

CFR: Code of Federal Regulations

CIC: Complete-information-critical

CID: Complete-information-desirable

CMO: collateralized mortgage obligations

CRU: Corpus of Regulation Units

CUSIP: Committee on Uniform Security Identification Procedures

DARPA: Defense Advanced Research Projects Agency

Def., Defn., Def'n: definition

Doc: document

DT: determiner (Penn Treebank tag)

FINCEN: Financial Crimes Enforcement Network

FINRA: Financial Industry Regulatory Authority

GALE: Global Autonomous Language Exploitation (a former DARPA Program)

HGS: Human Gold Standard

HTML: HyperText Markup Language

IARPA: Intelligence Advanced Research Projects Activity

IC: Intelligence Community

IN: Preposition or subordinating conjunction (Penn Treebank tag)

JJ: Adjective (Penn Treebank tag)

LDC: Linguistic Data Consortium

LKIF: Legal Knowledge Interchange Format

LTH: Semantic parser based at University of Lund, Sweden

MD: Modal (Penn Treebank Tag)

ML: Machine Learning

MSRB: Municipal Securities Rulemaking Board

NASD: National Association of Securities Dealers

NFA: National Futures Association

NL: Natural Language

NLP: Natural Language Processing

NN: Noun, singular or mass (Penn Treebank tag)

NNP: Proper noun, singular (Penn Treebank tag)

NNPS: Proper noun, plural (Penn Treebank tag)

NNS: Noun, plural (Penn Treebank tag)

OASIS: Organization for the Advancement of Structured Information Standards

PDF: Portable Document Format

RB: Adverb (Penn Treebank tag)

Ref: reference

Reg: regulation

Regex: regular expression

REMIC: Real Estate Mortgage Investment Conduit

RIF: Rule Interchange Format

SAIC: Science Applications International Corporation

SVM: Support Vector Machines

TAILCM: Toward Automated International Law Compliance Monitoring

TRACE: Trade Reporting and Compliance Engine

TTO: TAILCM-generated Test Ontology

USC: United States Code

VBG: Verb, gerund or present participle (Penn Treebank tag)

VBN: Verb, past participle (Penn Treebank tag)

VP: Verb phrase

W3C: World Wide Web Consortium

XML: Extensible Markup Language