Chapter 3

Deconstructing Contracts: Contract Analytics and Contract Standards

Kingsley Martin

Contents

Introduction

Contracts are the *language of business.* They document our business relationships and govern the global exchange of trillions of dollars of goods and services each year. Yet, despite their importance and the fact that we live in an increasingly interconnected world, contract drafting and review is relatively unchanged and has failed to keep up with innovation in other business sectors.

The consequences of current drafting practices have recently been quantified. They are alarming. Tim Cummins of the International Association for Contract and Commercial Management observes that "[p]oor Contract Management and contracting process can lead to value leakage of, on average, 9.5% of annual revenue."[1] Similar studies by Ernst & Young report that "[b]y having commercially efficient contracts, effectively managing these throughout their operational life and minimising waste in business activities with suppliers, an organisation can typically save between 5% to 15% of contract spend."[2] Finally, according to research by KPMG, ineffective governance of provider contracts can cause value leakage ranging from 17% to 40%.[3]

Given the significant loss of value, many governments, corporations, universities, and individuals around the world are working on next-generation systems to automate and streamline the contracting process. This chapter outlines some of the key initiatives. It divides the topic into four parts. The first part examines whether technology can truly be harnessed to perform contract-related tasks. The second part describes how theories of complexity can be applied to hierarchical deconstruction and simplification. The third part applies these theories to the practical technology of contract analytics.

The fourth part describes the modularization and simplification of contract language.

Can Technology Perform Human Tasks?

Before examining contract analysis, we must first answer the question whether technology can successfully emulate human tasks. One of the most prominent proponents of Artificial Intelligence, Ray Kurzweil, author of *The Singularity Is Near*, popularized the concept that technology will exceed human capacity in the near future. Technology, Kurzweil prophecies, will help design its successors, at ever-increasing speeds, creating a technology singularity, or an event horizon, beyond which we cannot see. Kurzweil's insight, along with many others, is that technology innovates exponentially, not linearly. This rate of change, which is occurring across all aspects of the technology spectrum, produces explosive growth, sometimes called the "hockey stick effect." One way to think about accelerating innovation is to consider all prior innovations since the birth of the computer and to imagine that all such prior capabilities will be doubled in the next one-to-two years.

> Here's what the exponential curves told [Kurzweil]. We will successfully reverse-engineer the human brain by the mid-2020s. By the end of that decade, computers will be capable of human-level intelligence. Kurzweil puts the date of the Singularity—never say he's not conservative—at 2045. In that year, he estimates, given the vast increases in computing power and the vast reductions in the cost of same, the quantity of artificial intelligence created will be about a billion times the sum of all the human intelligence that exists today.[4]

Of course, there are many who doubt that computers can become truly intelligent. For example, Professors Dreyfus and Dreyfus describe five stages of learning in their book *Mind over Machine* as a process that cannot be emulated by silicon and logic.[5] A *New York Times* opinion post, in response to IBM's Watson computer winning performance on Jeopardy, asserted that "Watson Still Can't Think."[6] But it does not matter whether or not we believe a machine is "thinking." It is the results that count. When we consider any situation from the perspective of outcome or results, what is the difference between intelligence, judgment, brute force, or deep learning?

Others, such as Microsoft cofounder Paul Allen, contend that although the singularity may very well occur; it is a very long way off.[7] Allen points out that achieving the singularity will require enormous developments in software (not just improvements in hardware capacity), and replicating human capacity is unlikely to occur at an accelerating pace. In fact, Allen asserts, we will likely hit a “complexity brake” because “[a]s we go deeper and deeper in our understanding of natural systems, we typically find that we require more and more specialized knowledge to characterize them, and we are forced to continuously expand our scientific theories in more and more complex ways.”

Once again, the core doubt is based on the fact that the human brain is exquisitely complex and therefore cannot be replicated. But, as one comment to Allen’s article states, the doubt whether a computer can perform human tasks is based on the “premise that singularity can only occur when human intelligence can be engineered. It’s far more likely that the inevitable accumulation of thousands of smart objects that don’t even attempt to mimic human cognition will lead to systems that have inhuman intelligence, with the ability to outperform humans on so many different tasks that the intellectual contributions of all but the most creative humans will simply be unnecessary.”

Or, as another comment predicts: the singularity is “the point at which the productivity growth rate permanently surpasses the economic output growth rate. Once that occurs, the economy must continuously shed (human) jobs.”

Even for proponents of advanced automation, the singularity is still many years in the future. Moreover, the progression should not be seen as an event horizon whereby at a certain point in time, machines become hyperintelligent. It is rather an evolution whereby technology advances in skill levels in a series of stages. The likely stages of automation are shown in Figure 3.1. The stages reflect the skills needed to answer a complex question, such as what form of agreement will best serve a client’s needs.

The first step requires us to find relevant material, such as prior examples of an agreement. We can accomplish this task using traditional cataloging techniques or with more modern search tools. Once we have collected relevant material, we must next analyze the data and determine the full range of clause elements (and alternatives) that may be found in each type of agreement. Finally, once we have identified all the elements and reviewed alternative forms, we must determine the optimal configuration.

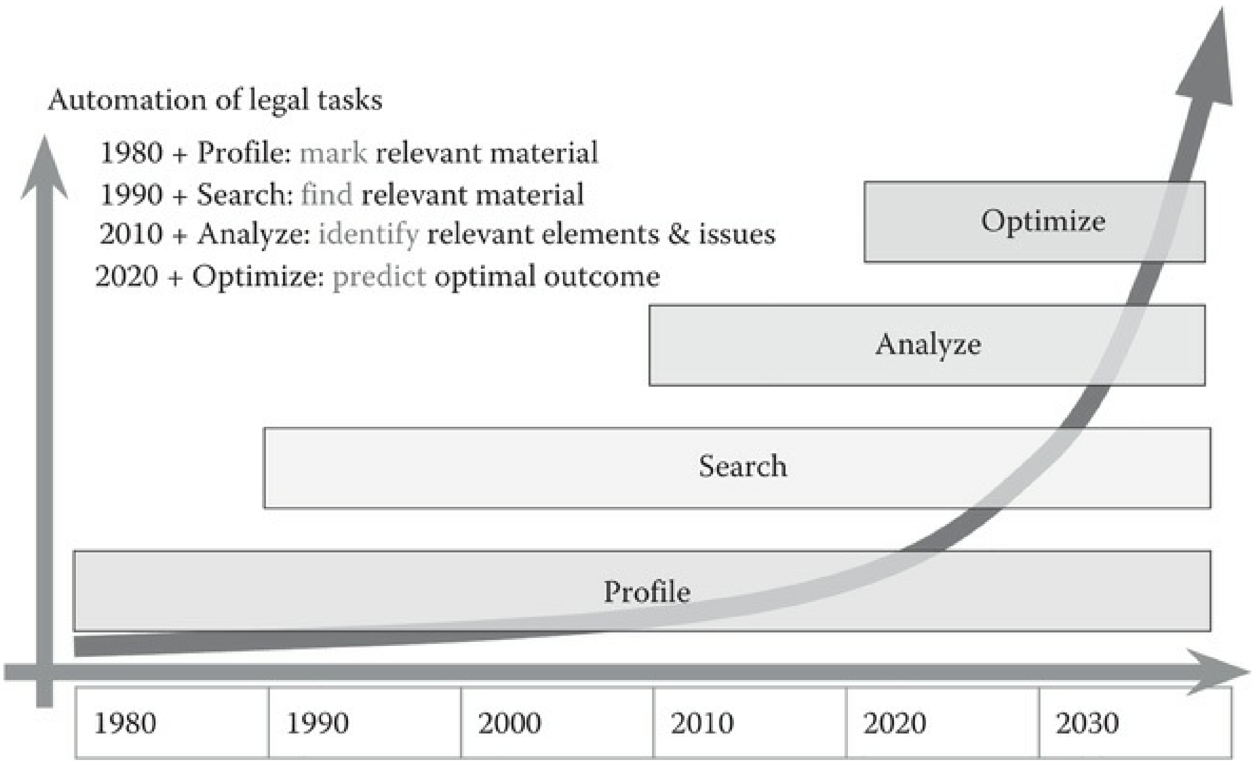

FIGURE 3.1 Stages of automation of legal tasks.

As shown in the illustration, technology does not yet have the capability to predict the best form of agreement. Technology, in itself, therefore, should not be seen as a current threat to take away jobs. Today, the state of the art is more like the magnetic resonance imaging machine used by the surgeon to augment the physician's insight. In the case of contract analytics, today's technology can analyze vast collections of agreements, create a checklist of items for the lawyer to consider, and offer examples of how other lawyers have drafted contract terms. Think of it as similar to a spell or grammar checker. When you draft a document, the checker compares what you wrote to its library of every correctly spelled word and grammar rule. It clearly shows you a comparison between what you wrote and what the standard is, and then allows you choose which one to use. Contracts analysis does something similar, but instead of just comparing simple spelling and grammar, it compares the complex clauses and language of your contract (either one you wrote or one someone else wrote that you need to review) with the clauses and language of a vast library of successfully executed contracts. It clearly shows you how your contract compares to the standard and allows you to quickly and easily make changes to improve the quality and comprehensiveness of your document.

Theory

The theoretical underpinnings of this chapter are founded on two main hypotheses. First: all complex systems can be examined through the process of hierarchical deconstruction, whereby the common building blocks or patterns can be discovered. Second: all complex systems—over time—follow a maturity model from one-off to commodities. In combination, deconstruction and maturity models allow us to "see the forest through the trees" and simplify the overall process, without limiting the capabilities of the resulting system to adapt to the full range of applicable circumstances.

Technology: Complexity and Hierarchical Deconstruction

The insight gained from hierarchical deconstruction draws from the science of complexity first developed by Nobel Laureate Herbert Simon. The goal of deconstruction, which has been applied to science, engineering, art, and literature, is to see through the cloud of complexity and identify the core elements and determine how they piece together. By any measure, contracts are complex documents. Seen as an assemblage of words, each agreement appears to be unique, and custom tailored to each transaction. But are they truly more different than they are similar? How can we see through the variance and complexity and see the fundamental patterns?

The first step is to examine the structure of agreements. Complexity theory "unlike traditional 'cause and effect' or linear thinking … is characterized by nonlinearity."[8] It addresses problems—like contracts—that are dynamic, unpredictable, and multidimensional, consisting of a collection of interconnected relationships and parts where certainty is replaced by probability. The thought leader of complexity, Herbert Simon, observed that all complex systems can be decomposed into a nested hierarchy of subsystems.[9] "Critically, however, not all these subsystems are of equal importance (i.e., centrality). In particular, some subsystems are 'core' to system performance, whereas others are only 'peripheral.' "[10] Contracts have similar attributes. Not all terms are equally important. In many cases just a small number of clauses are critical.[11]

An empirical analysis confirms that at the structural or building block level there is a high degree of similarity. This is not surprising since contracts document business relationships that have been performed millions of times. Over the years the manner and means of buying, selling, licensing, and engaging services has been documented into a knowable set of norms. For example, there are a finite number of ways to deliver goods, perform services, or pay for something. In fact, there is consistency across all agreements, as depicted in by the ContractStandards unified contract framework.[12] Of course, there are the very rare situations where a transaction is performed for the first time, but such rarities should not obscure the general rule. They are simply exceptions.

Content: Automation and Standardization

A Wikipedia page describes maturity as “a measurement of the ability of an organization for continuous improvement in a particular discipline.”[13] Different maturation stages can often be observed through the lens of other industries to better understand the process of commoditization. For example, Henry Ford revolutionized automobile manufacturing by introducing both the technology of assembly line and the introduction of modular, standardized components. Ford would not have met with success if he just introduced the automated assembly line. The line will quickly stall if each component of a car were designed differently without any thought how they will fit together.

In the legal industry, the trajectory is described by Richard Susskind in *The Future of Law*.[14] He categorized the phases as a journey from one-off, to standardized, to systemized, to packaged, and finally to commoditized. Unfortunately, law, for the most part, is still stuck on the bespoke phase, where all matters are treated as unique requiring custom-tailored solutions (Figure 3.2).

Contract Analysis

Evolution of Contract Technology

Over the next few years, technology will likely evolve from point solutions—automating particular tasks—to fully integrated platforms enabling a contract automation assembly line, orchestrating all phases of the contract lifecycle. The overall trend is to introduce more standardized processes and then to integrate such standards across all systems both internal and external. Today, the majority of contract tasks are performed by professionals with limited application of technology. Over time, this pattern is likely to reverse with more tasks handled by standard procedures and fewer tasks performed by skilled individuals undertaking nonstandard tasks. As illustrated in Figure 3.3, the trend line portends a decreasing percentage of tasks will be performed by professionals in a one-off manner and an increasing percentage performed by technology and systems.

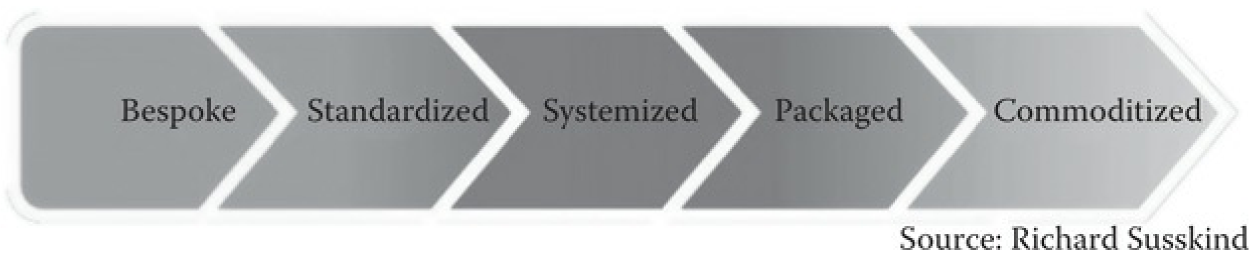

FIGURE 3.2 Evolution of legal services.[15]

Tools of Contract Analysis

Recently, contract analysis has emerged to offer promising tools to give us insights into individual contracts and portfolios of thousands of agreements. At its core, contract analytics is a suite of technologies and processes capable of parsing ones, tens, hundreds, thousands, or even millions of contracts and identify patterns within and across all agreements.

The overall approach is a hierarchical deconstruction. First, the algorithms deconstruct agreements into their component clauses (or building blocks). Second, each clause is broken down into sentences, and the software analyzes the precise language of each provision. Finally, sentences are examined at the word level to identify key contract variables, such as names, places, dates, and amounts. The overall approach is illustrated in Figure 3.4.

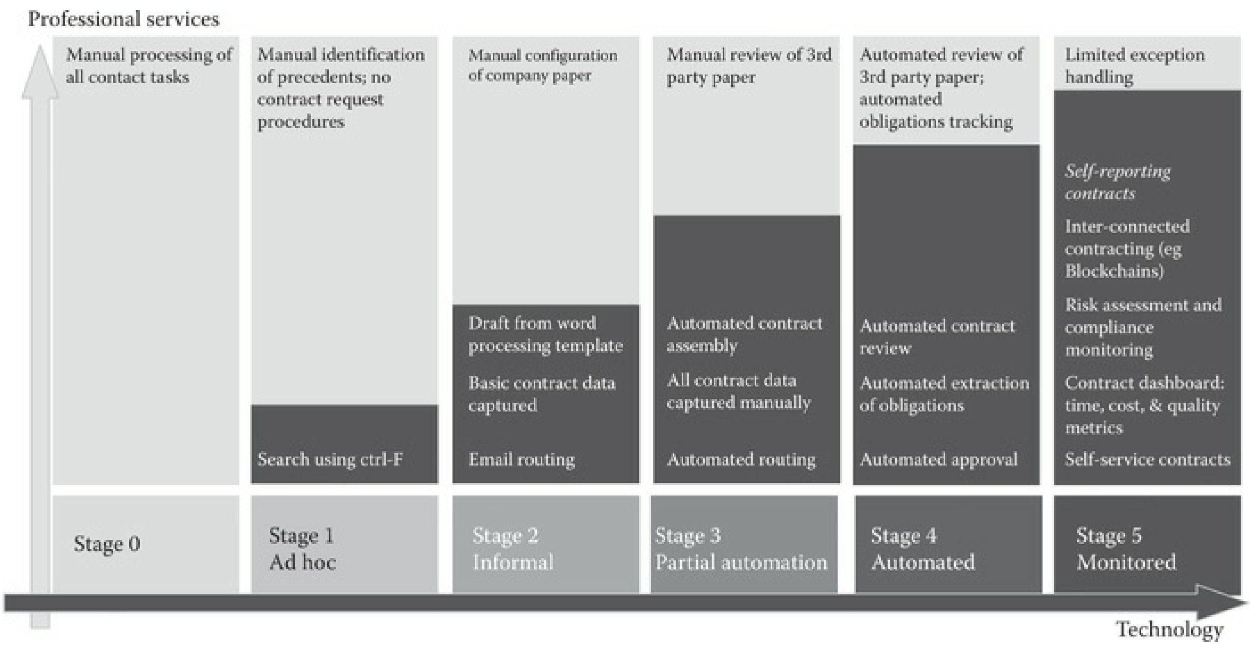

FIGURE 3.3 Contract technology maturity model.[16]

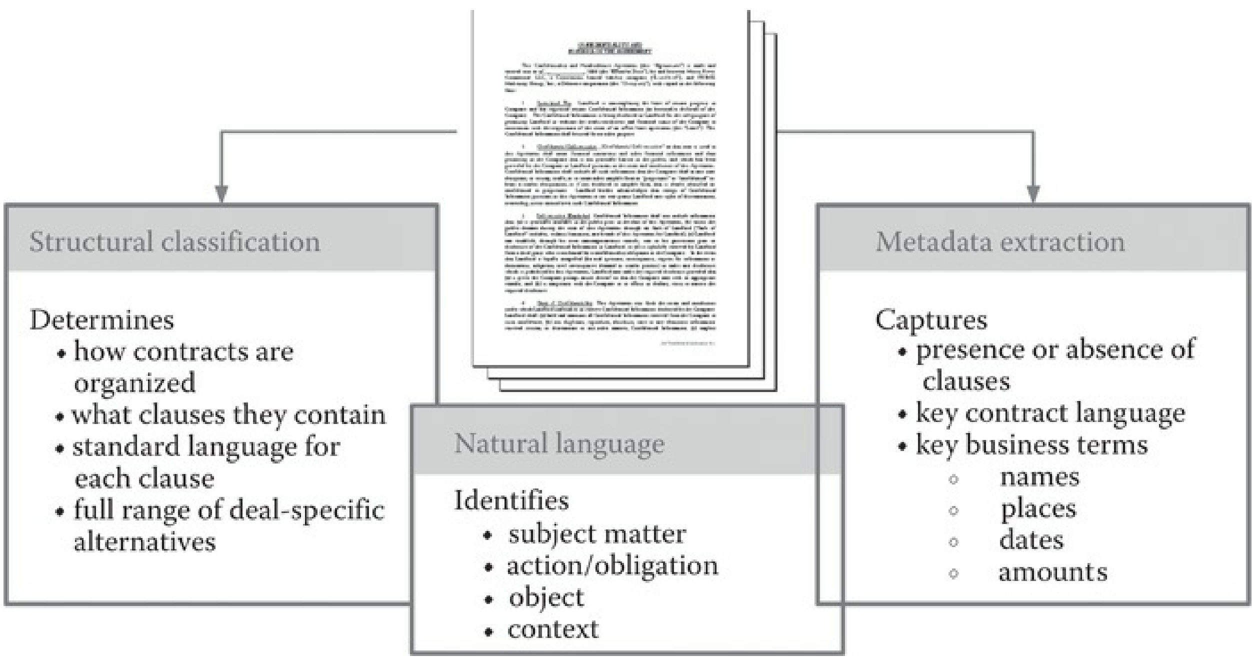

FIGURE 3.4 Suite of contract analysis tools.

Structural Classification

Contract analysis applies inductive reasoning techniques in order to identify the elements of a particular type of agreement. Algorithms first deconstruct a sample set of agreements into base components, such as clauses and sentences. Next, the software looks for similar components in other documents. Finally, the tools aggregate the components in a single organizing framework and capture key statistical information.

Deconstruction

Deconstruction may generate a flat list of clause paragraphs in the case of a simple agreement or a hierarchical outline of articles, sections, clauses, and subclauses in the case of longer and more complex agreements. One key advantage of deconstruction by contract sections is that the output can take advantage of captions that serve as outline headings and group-related concepts. For example, the software can group all representations, warranties, and covenants into sections of related concepts.

Matching

For each clause block or sentence, software next finds the closest match to such block or sentence in all other agreements. Typically, closest match algorithms apply some form of term frequency (TF)–inverse document frequency (IDF). The formula is composed of two elements. First, TF measures how frequently a term occurs in a document or a text block in a set of documents, using the formula:

$$\mathrm{TF}(t) = \left(\text{Number of times term } t \text{ appears in a document}\right) / \left(\text{Total number of terms in the document}\right).$$

However, TF will identify common (or nondistinguishing) words, such as “the” or “and.” The second element, IDF, measures the importance or the distinguishing nature of a term, using the formula:

$$\mathrm{IDF}(t) = \log\left(\text{Total number of documents/Number of documents with term } t \text{ in it}\right).$$

When applied to a set of clauses (as opposed to documents), the formula will yield a high IDF score for the words “governed,” “construed,” and “interpreted” in the case of the governing law clause. These are the words that appear with high frequency in the clause compared with any other clause and therefore can be used as search patterns to find matches in other agreements.

TF-IDF can be refined with adjusting word weights using additional statistical measures (such as word proximities and n-grams[17]). Figure 3.5 shows an example of matching text block groups from a set of resumes to identify a section captioned “hobbies.”

Aggregation

The process of aggregation reassembles all the matched elements into a single, common outline. A method of matching blocks is shown in in Figure 3.6. The resulting outline is shown in Figure 3.7. The process is typically iterative. It is similar to the game MasterMind™, whereby the players attempt to decode a pattern sequence by building up their knowledge through a series of questions and binary answers. At each pass, the software organizes and sequences the clauses in a particular order and determines how many documents in the sample set will it match. It then reorders and resequences the evolving outline and determines whether the structure matches more (or less) of the samples. Each time it matches more, it moves closer to the aggregate standard. The result is an outline of the hierarchical building blocks that best match the greatest number of instances in the sample provided.

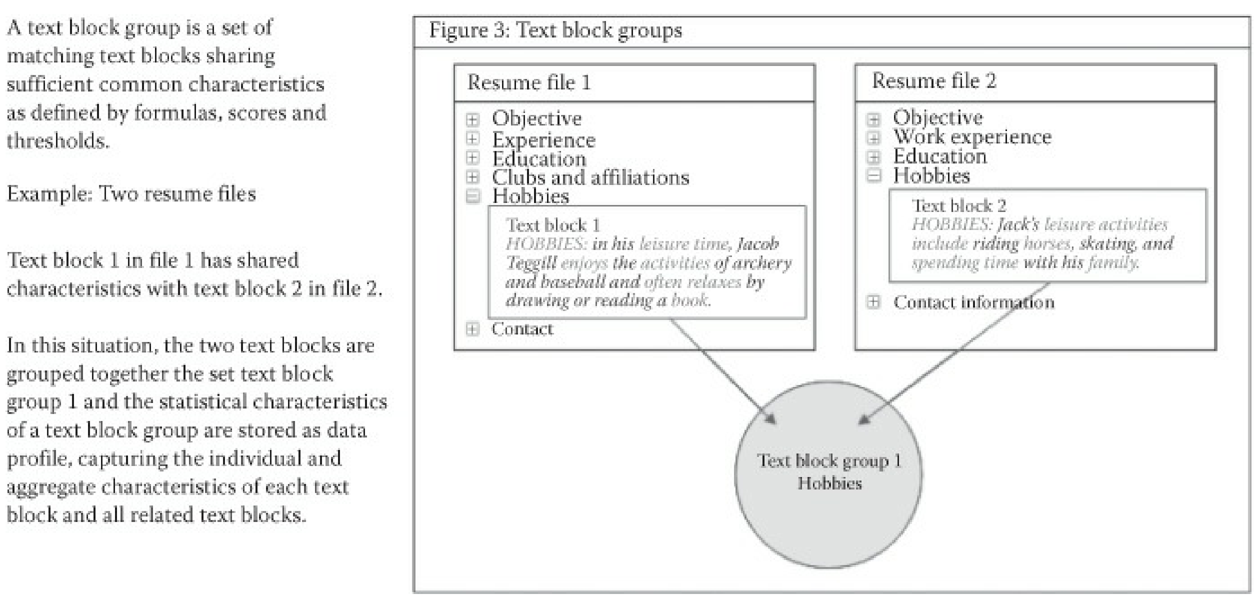

FIGURE 3.5 Matching text block groups.[18]

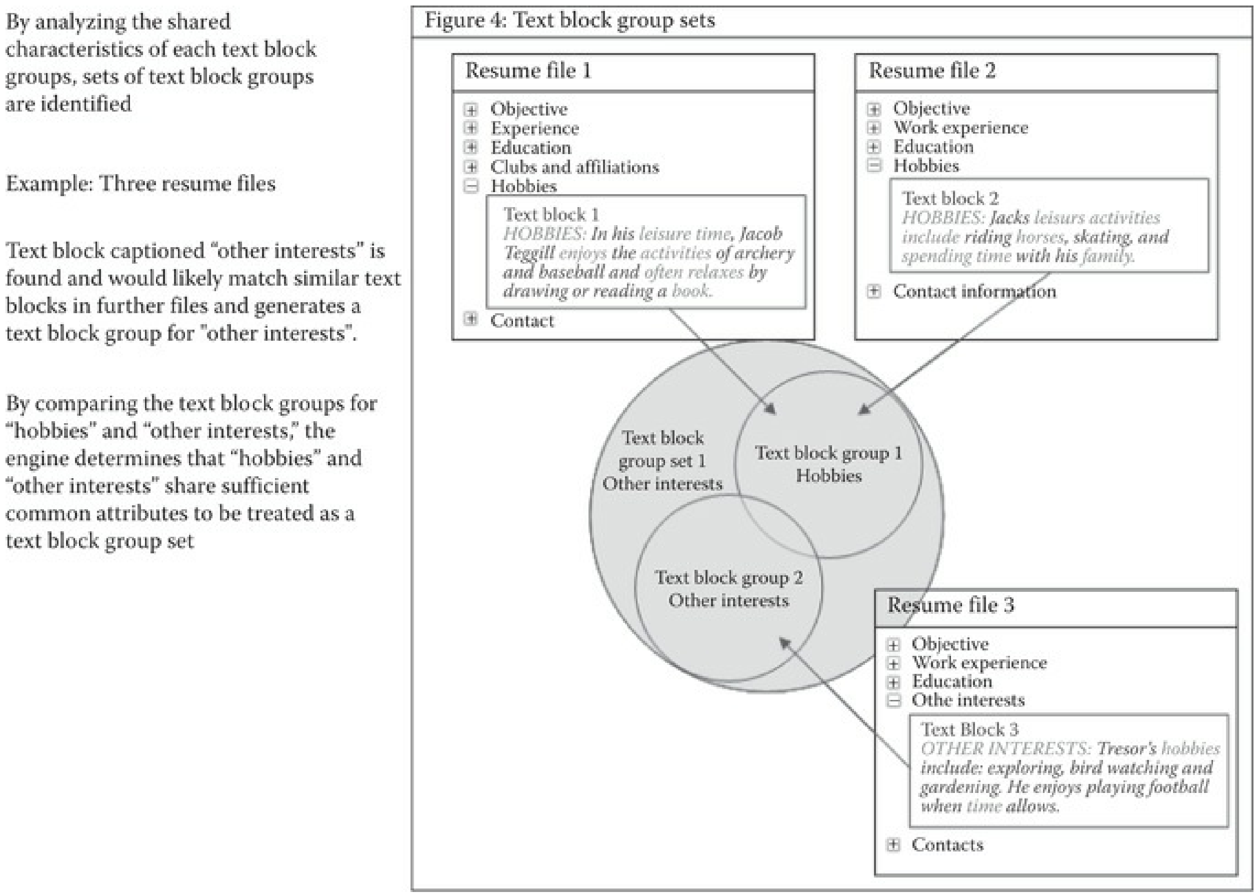

FIGURE 3.6 Matching text block groups.[19]

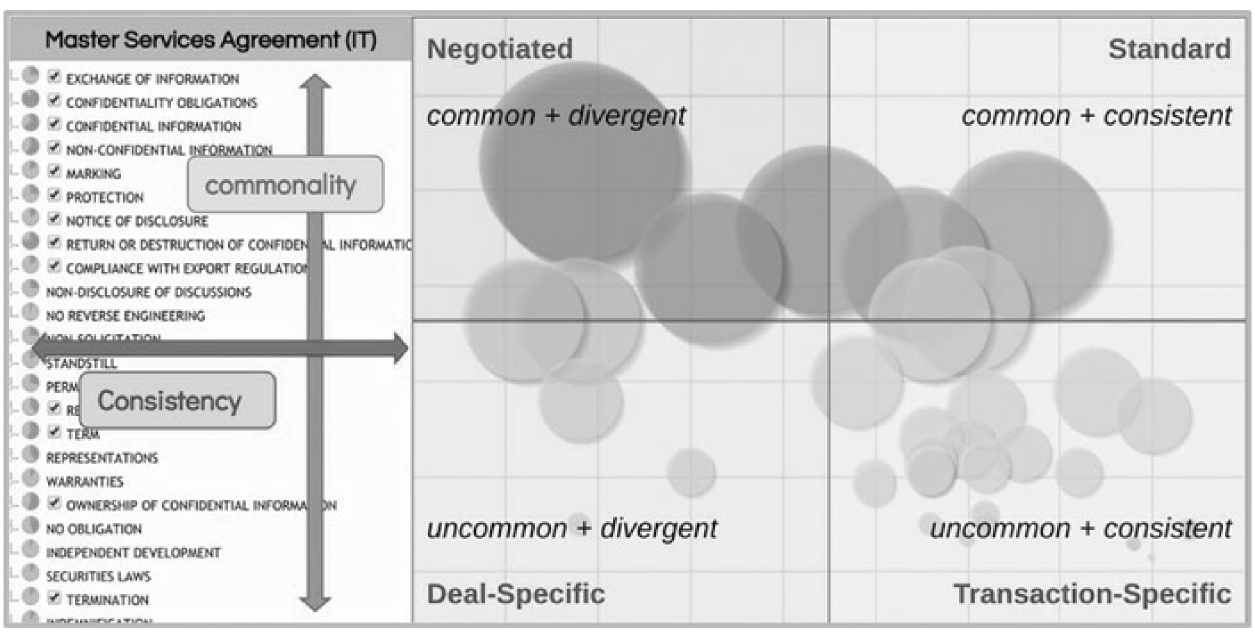

FIGURE 3.7 Contract analysis visualization.

Reference Standard—A Statistical Benchmark

In the aggregation process, the software also captures key statistics for each branch in the common outline. The software measures the frequency of each clause across the set. This commonality number gives insights into the use of the provision. Does it appear in most agreements in the sample? If so, we can make the assumption that the clause as required. Alternatively, does it appear in some, but all the sample agreements? If so, we can assume that clause is optional to be used in specific circumstances. In addition, the software can measure the consistency of the language across the sample, providing further insights. Where the language is generally consistent, we can consider the term standard (or perhaps boilerplate). Where the term is highly variant, we can classify the term as highly negotiated or one to be configured to particularly deal circumstances.

The analysis can be displayed as an outline of all the clauses found in all the sample, organized in the manner that is most representative of all the samples. The clauses can also be plotted on a graph showing negotiated, standard, transaction, and deal-specific clauses as depicted in Figure 3.7.

- The upper-right quadrant—*standard clauses*—contains frequently occurring, consistent clauses. Examples of such clauses should include so-called "boilerplate" provisions. However, as the analysis can also show, such boilerplate provisions may, in fact, exhibit wide variation in language.
- The upper-left quadrant—*negotiated clauses*—contains commonly occurring, divergent clauses. These are clauses that appear in most agreements but contain different language. Such language variation may occur due to negotiation between the parties creating different terms or it can be attributable to different drafting customs and personal preferences. For example, the purchase price clause in an acquisition agreement will likely exhibit wide variation across a set of documents. However, similar degrees of variation can also be found in a severability clause.
- The lower-right quadrant—*transaction-specific clauses*—contains clauses that do not occur frequently, but when they are found contain consistent language. A good example of a transaction-specific clause is "The Offer" term found in a set of merger agreements. This term may appear in about 20% of the agreements—indicating that such transactions are structured as a tender offer. In these cases, the language will likely be consistent.

- Finally, the lower-left quadrant—*deal-specific clauses*—contains infrequently occurring clauses that display wide language variation. These are clauses that are custom-tailored to a particular transaction.

However, not all clauses fit neatly into this schema. Clauses that should contain consistent language are often, in practice, highly variable. In large part, this background divergence can be attributed to custom and personal drafting preferences. Indeed, this is very evident when analyzing documents from a large number of different organizations, compared with a set from a single law firm or corporate legal department. The more varied the source of the documents, the more diverse the clauses and the language. Of course, the fact that divergence increases with the diversity of the source set should not be surprising because they will contain a broader range of drafting customs and personal preferences. The key point is that as divergence increases above a certain threshold, the usefulness of the statistics is reduced.

Language (Legal Terms)

After identifying the structural clause blocks, technology can next examine the language of each provision, capture context, and some understanding of its meaning.

Supervised Deconstruction (Break-It-Down Checklists)

Just as the software analyzes entire agreements and identifies the clause building blocks, algorithms investigate the common elements of complex provisions in a manner similar to deconstructing entire agreements into clauses, but now at a more granular level. For example, an indemnity clause can be broken down into the core, highlighted elements: who indemnifies whom, for what, and under what circumstances. The approach can be applied to a recent merger agreement, where the highlighted text shows the common elements.

Once again, an empirical analysis confirms that the core concepts in each provision are highly consistent in substance, while the precise words may change semantically from one clause example to another.

Programmatic Deconstruction

Today, “break-it-down” analysis is done in a semiautomated manner whereby experts review program output and craft comprehensive checklists identifying the core and optional elements for an indemnity and all other clauses. At the same time, technologists are working on further automating the process to create detailed checklists for all agreement types. Most of this work builds on Natural Language Processing (or NLP). NLP is the ability of a computer program to understand human communications in written and oral forms. It is composed of numerous subspecialties.[20] At its core, NLP parses sentences into their lexical components by tagging words with parts of speech, finding the relationships between the words, and resolving ambiguities.

> “ X. Indemnification. *Subject to the terms and conditions of this Article…, upon the Closing of the Transactions,* **Parent shall indemnify and hold harmless each of the Company Securityholders** and each of their respective Affiliates, and the representatives, Affiliates, successors, and assigns of each of the foregoing Persons (each, a “Seller Indemnified Party”), **from**, against and in respect of any and **all Damages** incurred or suffered by the Seller Indemnified Parties **as a result of**, arising out of or relating to, directly or indirectly:
>
> (a) **any breach or inaccuracy of any representation or warranty** of Parent or Merger Sub set forth in this Agreement or the certificate of Parent or Merger Sub delivered at the Closing pursuant to Section… (or the assertion by any third party of claims which, if successful, would give rise to any of the foregoing); and
>
> (b) **the breach of any covenant or agreement of Parent or Merger Sub** in this Agreement or any other agreement contemplated by this Agreement to which Parent or Merger Sub is party (or the assertion by any third party of claims which, if successful, would give rise to any of the foregoing) and the breach by the Company of the covenant set forth in Section… hereof.

One of the best known—and widely used—NLP platforms is Stanford’s parser. For example, the Stanford parser can tag the words with parts of speech and identify word relationships, as shown in Figure 3.8.

Fortunately, contracts typically use a narrow lexicon compared with common parlance. They are written as a series of declarative statements in the form of subject, verb, and object. This gives NLP a framework to determine who is the actor (or subject), what is the action (or verb), and what is the nature of the action (the object). Of course, this can be complicated (and open to greater interpretation or ambiguity) when attempting to parse long, compound, or very complex sentences. However, it should also be said that humans will equally struggle to find definitive meaning in sentences exceeding a few hundred words.

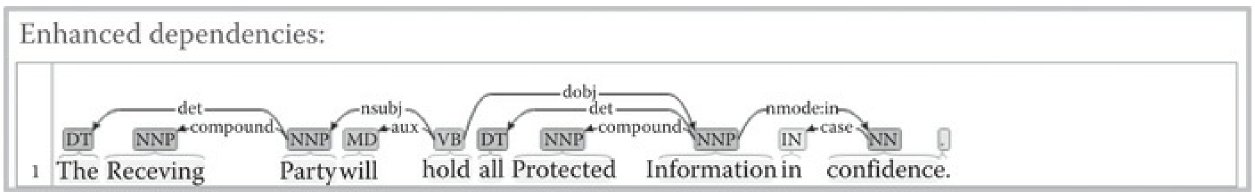

FIGURE 3.8 Stanford NLP parser.[21]

Data (Business Terms)

Identifying Business Terms

Finally, within each sentence, the software can examine the words and identify key variable terms—often representing the business terms of the agreement—such as names, places, dates, values, etc. For example, in the following sentence, key terms are highlighted.

“	This Master Services Agreement is made on January 1, 2017, between ABC Inc., Delaware corporation (the “Seller”) and XYZ, Inc., a California limited liability partnership (the “Buyer”).

The extractions can be summarized in a term sheet capturing key variables such as the type of agreement, names of parties and law firms, consideration, governing jurisdiction, and notice periods.

Tools for Metadata Extraction

The programmatic tools used for automated data extraction are typically a combination of technologies including regular expressions and NLP. A regular expression or regex for short is a sequence of characters that define a search pattern. In some cases, they can be words or parts of words. But they can also be character patterns, such as three numbers followed by a dash, followed by two numbers, followed by a dash, and followed by four numbers: a pattern that describes a social security number.

This pattern can be expressed in regex as: ^\d{3}-\d{2}-\d{4}$. Where \d represents a number from 0 to 9 and {3} means repeat the last instruction three times.

But even relatively predictable patterns such as social security numbers can get complex when, for example, dashes are replaced with spaces or dots. In the case of something like an address, significant variability requires programmatic solutions to detect likely parts of an address and apply probabilistic solutions to predict how likely the text is, in fact, an address.

Despite the complexity, significant advances are made each year with many resources, such as the Stanford NLP parser being shared with a global community of developers. Such open sourcing is fueling increased innovation. For example, Figure 3.9 shows what the Stanford named entity parser captures in the sample opening sentence.

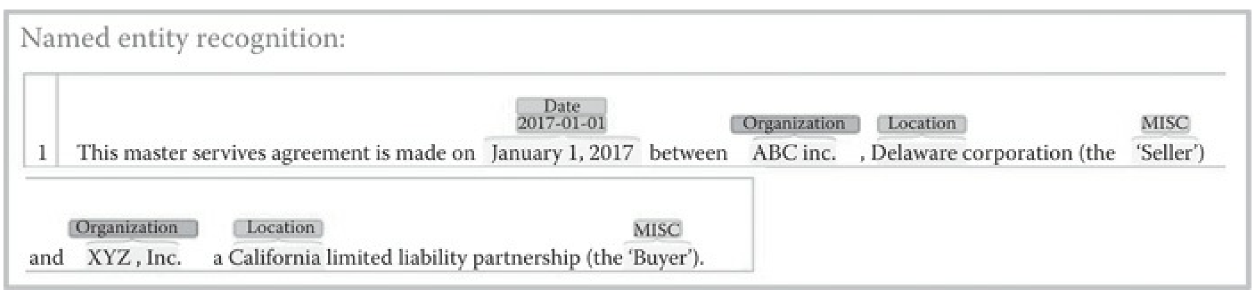

FIGURE 3.9 Results from the Stanford named entity parser.

Training Approaches

The most significant developments in contract automation have been in the area of machine training. This work, as shown in the table, focuses on creating the elements of a particular type of contract and matching similar elements in other agreements. It has progressed from a largely manual set of procedures to one that is increasingly automated.

Approach	Outline	Matching
Manual	Manual identification of clauses	Rule-based matching
Supervised	Structural matching	Machine learning/TF-IDF
Automated	Deep learning	Machine learning/TF-IDF

Manual: Rule-Based Learning

In the early days, contract analysis was mostly a manual undertaking. Experts, often with the use of search tools, manually created a checklist of terms. Then for each term, programmers developed if-else-then, rule-based scripts to find sample clauses. For example, using a regular expression, a search pattern, such as the governing law example shown in the table later, can attempt to find matching sentences.

<table>
<tr><td></></td><td>th(?:e|is)\s+{AGREEMENT}.*?(?:governed|construed).*?laws.*?of.*?({STATE})
Where {AGREEMENT} and {STATE} are macros that expand out to a list of alternatives.</td></tr>
</table>

The challenge for such an approach is that an expert created, manually constructed pattern is unlikely to find all instances of a governing law clause. In technical terms, it may yield high precision, but its recall may be low due to high variation found in contract language. Moreover, the patterns will need to be manually created for all languages.

Supervised: Machine Learning

To overcome the limitations inherent in rule-based techniques, machine learning tools (such as TF-IDF) train the computer to identify the matching characteristics. Experts define the relevant blocks, provide exemplars or guidance, and the machine identifies the relevant characteristics that distinguish one text block from another.

> “ Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.[22]

Machine learning is explained by Tom Mitchell as “[a] computer is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T as measured by P improves with experience E.”[23]

Automated: Deep Learning

Deep learning takes this process one step further. Algorithms automatically identify the building blocks and which features are important for classification (Figure 3.10)

> “ Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones.[24]

> “ The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones. If we draw a graph showing how these concepts are built on top of each other, the graph is deep, with many layers. For this reason, we call this approach to AI deep learning.[25]

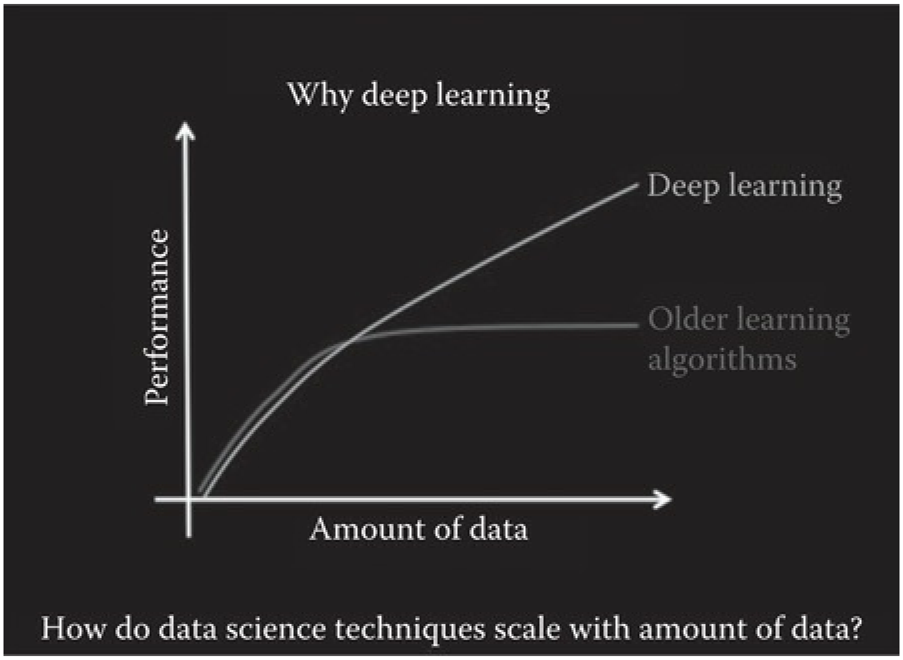

FIGURE 3.10 Comparison of deep learning.

Deep learning is best performed holistically, giving the software the entire agreement as a basis for examination. As a result, deep learning systems (as shown in Figure 3.9)[26] take much longer to train and require far more computing power. However, once trained, testing can be performed faster compared with traditional machine learning techniques as testing time increases with the amount of data.

The challenge for the analysis of transactional agreements is often a lack of high volumes of samples and a high variability in content. Indeed, this is the main reason why training in contract analytics has been based on supervised, machine learning techniques.

Contract Standards—Modularization, Standardization, and Simplification

Evolution of Content Standards

As Richard Susskind observed, the journey from one-off contracts to systematized and packaged systems starts with establishing standards.[27] This trend toward standards is a path followed by all businesses. Of course, law presents a more challenging technical exercise compared to automated manufacturing, where the subject matter is the full breadth of human and business interactions expressed in language. Nonetheless, as Figure 3.11 shows, the trend line evolves from one-off contracting—where every agreement is slightly different—to portfolios of contracts built from standard, modular and reusable components, configured by input variables.

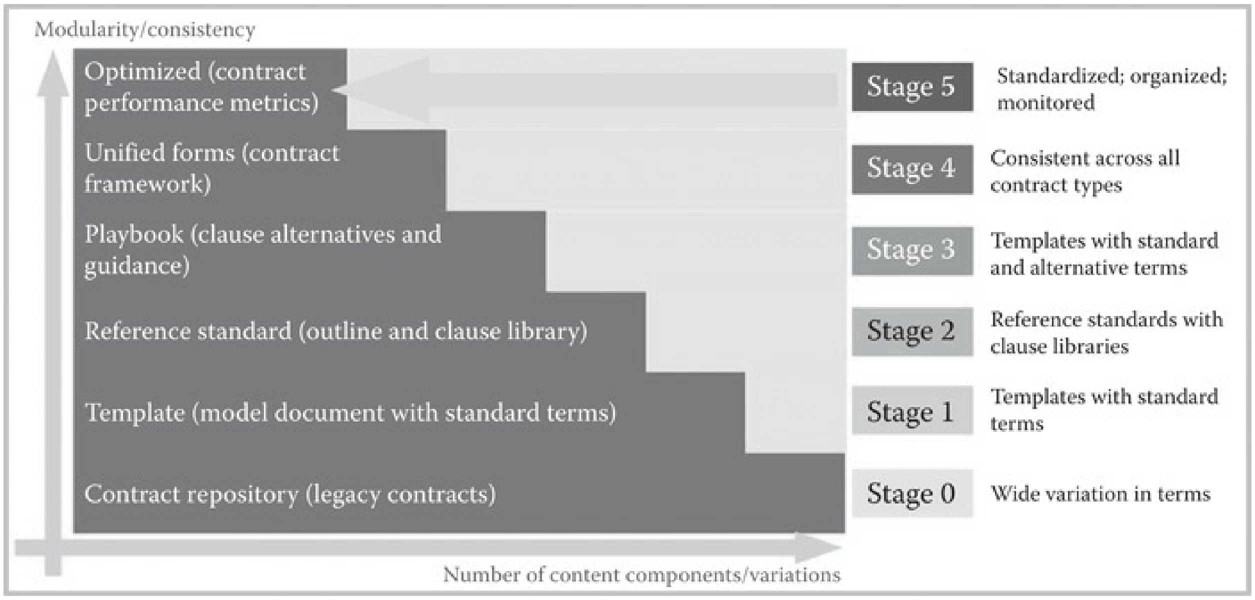

FIGURE 3.11 Evolution of content from one-offs to modular components.[28]

The methodology of content standardization and modularization follows a similar course as the approaches to technology. It also applies the technique of hierarchical deconstruction. We first organize all contract types into modular classes, then standardize the clause building blocks,[29] and finally simplify the language.[30]

Consistent Organization—Unified Forms Library

The first step towards a unified forms library is to collect and organize all contract types in a single taxonomy or complementary set of taxonomies. This can be done by collecting the titles of all agreements filed on, for example, the U.S. Securities and Exchange Commission's Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system or other publicly available resource. In such an exercise, an analysis of EDGAR filings yielded approximately 750 agreement types (although this list can be further refined to remove overlapping types and limit the list to about 500). These agreements can then be organized into a taxonomy organized by the nature of the agreement, the bargain, and the property. At ContractStandards, we categorize agreements in the following manner.

- Nature of the agreement
  - – unilateral agreements (such as wills, trusts, insurance agreements, share certificates, etc.)
  - – exchange agreements between two or more parties
  - – organization agreements (such as bylaws, operating agreements, etc.)
- Nature of the exchange
  - – purchase and sale agreements
  - – lease or license agreements
  - – Service or performance agreements
- Nature of the asset, interest, right, or restriction
  - – Property (including real property, tangible property, intellectual property)
  - – Interest or right
- Temporal nature of the asset, interest, right, or restriction
  - – Existing rights
  - – Future rights
  - – Contingent rights

Consistent Contract Terms—Unified Contract Framework

Within each contact type, software and experts can identify the common clauses. Some may be unique to a particular agreement type, others may be found in many different agreements. While a definitive empirical study has yet to be completed, the total number of distinct clauses can be estimated with the following formula.

$$\text{Estimated number of exchange agreements}(500)$$
$$* \text{ average number of clauses per agreement }(50)$$
$$* \text{ percentage of unique clauses per agreement }(0.3) = 7{,}500.$$

Accordingly, it is estimated that with a library of 7,500 clauses (a relatively small number for computers) any agreement can be assembled.

The clauses can be further organized into a common or unified framework, as shown in Figure 3.12. The exchange framework is composed of a grid of three rows and three columns. The first row of terms describes the business agreement. It details the nature of the transaction (describing the value given and received by each party), the mechanics of the exchange (detailing how the consideration will be exchanged or received), and the period of time during which the parties are bound by the terms of the agreement. The second row describes the statements, actions, and circumstances that each party requires to assure that they receive the benefits of their bargain. Finally, the third row describes what happens in the event of a breach of the agreement or other circumstances preventing the parties from realizing the benefits of the bargain and how the agreement is interpreted.

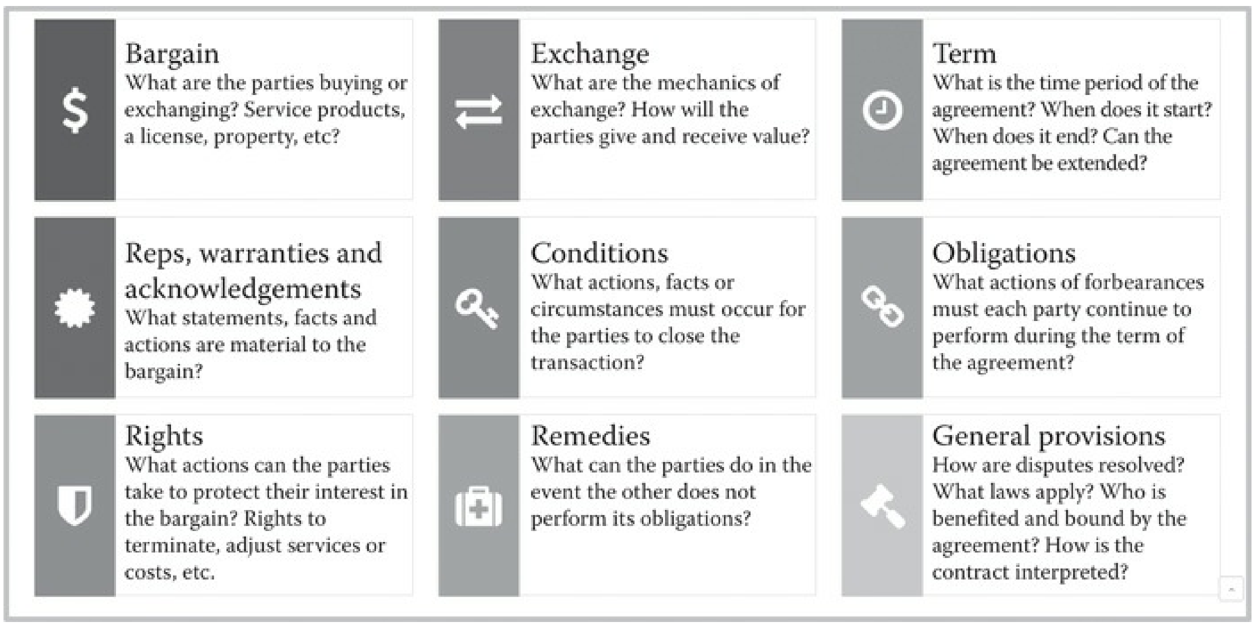

FIGURE 3.12 ContractStandards unified contract framework.[31]

Clause Variables: Playbooks

For each clause in the agreement, the software further identifies variations. In general, such clause variations are driven by three main factors: the nature of the transaction, the degree to which the clause favors one party or the other, and the requirements of the governing jurisdiction (in that order of importance). It is commonly believed that contract terms must be carefully crafted to the meet the mandates of each governing jurisdiction. In truth, this is significantly overstated. There are relatively few contract terms that must be specifically worded to comply with local laws. A contract is a private bargain, enforceable by law. The parties are free to enter into any terms of agreement, provided they are not illegal or contrary to public policy. Terms that are clear, fair, and balanced are more likely to be enforced by the courts —and less likely to be litigated in the first place.

Clause variations can be provided in the form of a playbook, which provides contract professionals with guidance on the use of each clause alternative, marking them in terms of preference. For example, some businesses use a PADU system, tagging clauses as Preferred, Acceptable, Discouraged, and Unacceptable.

Deal Variables: Term Sheets

In addition to clause variations, text variables data (identified through metadata extraction tools) capture key contract or business terms. As noted earlier, these variables identify names of parties, dates, amounts, and places. In combination with all clause variations, software can outline all the choices that need to be made to configure a contract to the needs of a transaction. The process can be viewed as reverse engineering legal logic from a sample set of agreements.[32] While this may sound like an intrusion or even a usurpation of human intellect, it simply mirrors that way we learn. The main difference is that we can learn faster by training machine with inductive reasoning methods.

Monster Matrix

The ultimate goal is to create a resource for all contract types and a library of modular clauses (together with their deal-specific variations) that can, in combination, create every contract type and tailor each instance to the particular needs of any transaction. I sometimes refer to the final construct as the "monster matrix." It captures every contract type in the horizontal dimension and each contract clause the vertical (Figure 3.13).

Consistent Language—Style Guide

Contact consistency is perhaps best achieved through a style guide, ensuring that all agreements are written in a consistent manner and written in a form that can be read by both humans and machines.[33] The style guide is used to draft contract sentences in a consistent manner. Applying a consistent approach to drafting, contractual sentences can be classified into three basic forms of sentences, each written in the form of subject–verb–object.

1. Organization: Follow a consistent, logical organizational structure
2. Headings: Use informative headings that serve as a summary of the contents
3. Clauses: Break text into smaller units with one substantive topic per clause
4. Sentences: Draft in short, declarative sentences
5. Words: Use standard language, without jargon or legalese
6. Punctuation: Use punctuation to aid readability

FIGURE 3.13 ContractStandards drafting principles.

- Statements of the Parties
  - – Obligations (what the parties must do): Party…will…verb…object.
  - – Restrictions (what the parties can't do): Party…will not…verb…object.
  - – Permissions (what the parties can do): Party…may…verb…object.
  - – Statements (what the parties agree to): Party…[copula]…verb…object.
- Statements of the Agreement
  - – Positive = This agreement will be…verb…object.
  - – Negative = This agreement may not be…verb…object
- Definitions
  - – Term… ("means"/"includes")…definition

Numerous agreements contain a fourth type of sentence, namely sentences about other subject matter topics. However, many of these sentence forms are the form of passive statements and should be avoided.

Obligations, restrictions, and permissions can be further refined and classified by NLP as being

- Dated (such as an obligation to deliver or pay on a particular date),
- Ongoing (such as a monthly payment obligation), or
- Contingent (such as an obligation conditional on another action or event).

In addition, contractual statements can be subject to qualifications (such as requirements, exceptions, minimums, maximums, or other restrictions).

Consistent Wording—Controlled Contract Language

The final level of consistency may be achieved through the creation of a controlled contract vocabulary, where contractual statements are crafted from a limited lexicon of words. Key contractual statements are generally expressed through a modal verb and a main verb. In the English language, there are 12 modal verbs of which just five commonly occur in agreements (can, may, must, shall, and will). The process of generating a controlled contract language can be aided with technology to examine the frequency of verb usage for each clause type and use the generated list to select the clearest verbs for each contractual obligation.

Many will doubt the feasibility of such narrowing of contract terms to a limited and controlled list. Lawyers may view it as a severe restriction on the need to custom-craft languages to nuances of each transaction. However, a recent exercise shows how the approach can offer real value. The study analyzed a large number of nondisclosure agreements and identified the core confidentiality obligations. The analysis found wide variations in wording empirically measured through Levenshtein distance vectors, which measures the similarity between two blocks of text. However, when captured as a checklist of concepts, the software found four main confidentiality obligations:

- To use the information solely for the purpose of the disclosure (87%)
- To keep the information confidential (93%)
- To protect the information from loss or unauthorized disclosure (42%)

- To notify the disclosing party in the event of loss or unauthorized disclosure (12%)

The software further captured how frequently each obligation appeared in the set (shown in parentheticals). With this information, a new simplified standard can be proposed using a controlled contract language with the benefit of significantly reducing contract length, while increasing readability. A sample set of obligations is illustrated as follows.

Use of Information. The receiving party will use the Protected Information solely for the [Purpose].
Confidentiality Obligation. The receiving party will hold the Protected Information in confidence.

Protection of Information. The receiving party will exercise reasonable care to protect the Protected Information from any loss or unauthorized disclosure.
Notification of Disclosure. The receiving party will immediately notify the disclosing party upon discovery of any loss or unauthorized disclosure of Protected Information.

Conclusion: The Future of Contracts

The world of contracts, which has been largely unchanged for hundreds of years, is about to enter a period of rapid transition fueled by a global economy. Agreements that in the past were solely rendered in printed form may now be prepared, executed, and delivered in electronic form. Indeed, some terms may be linked to information on websites that may be amended with or without notice.

The combination of contract automation and contract standards will likely trigger the rise of contract apps. Not just applications to assemble and review contract terms, but rather technologies to link contract terms to applications. The best known is the blockchain designed to securely manage the ledger of obligations and payment terms. We will also see the integration of insurance applications to protect against loss (such as title insurance or reps and warranty insurance), to manage assurances (such as escrows), compliance applications to manage interparty obligations, and to ensure conformity with rapidly changing regulatory standards.

[1] Tim Cummins, Poor Contract Management Costs Companies 9%—Bottom Line, Oct. 23, 2012, Commitment Matters blog, https://commitmentmatters.com/2012/10/23/poor-contract-management-costs-companies-9-bottom-line/ (last visited Dec. 9, 2017).

[2] Ernst & Young, Supporting Local Public Services Through Contract Optimization, 2016, www.ey.com/Publication/vwLUAssets/Supporting_local_public_services_th _Contracts_optimisation/$FILE/EY_Contracts_optimisation.pdf (last visited Dec. 9, 2017).

[3] Sourcing Focus, How to Stop the Value Leakage, 2016, www.sourcingfocus.com/site/featurescomments/how_to_stop_the_value_lea (last visited Dec. 9, 2017).

[4] Lev Grossman, 2045: The Year Man Becomes Immortal. We're fast approaching the moment when humans and machines merge. Welcome to the Singularity movement, TIME, Feb. 10, 2011.

[5] Hubert L. Dreyfus, Stuart E. Dreyfus & Tom Athanasiou, Mind over machine: The Power of Human Intuition and Expertise in the Era of the Computer (1988).

[6] Stanley Fish, Watson Still Can't Think, *The New York Times*, Feb. 28, 2011.

[7] Paul Allen, The Singularity Isn't Near, *MIT Technology Review*, Oct. 12, 2011.

[8] Complexity Science in Brief, 2012 (www.uvic.ca/research/groups/cphfri/assets/docs/Complexity_Science_in_B (last visited Dec. 8, 2017).

[9] Herbert A. Simon, The Architecture of Complexity, Proceedings of the American Philosophical Society, Vol. 106, No. 6. (Dec. 12, 1962), pp. 467–482.

[)] Alan MacCormack, Carliss Baldwin, and John Rusnak, The Architecture of Complex Systems: Do Core-periphery Structures Dominate? Harvard Business School Working Paper 10-059, Jan. 19, 2010.

[1] Kingsley Martin, Some Observations on the Nature of Contract Drafting, Feb. 28, 2011 (http://contractanalysis.blogspot.com/2011/02/some-

observations-on-nature-of-contract.html, last visited Dec. 8, 2017).

[2] www.contractstandards.com/resources/csframework.

[3] https://en.wikipedia.org/wiki/Maturity_model (last visited Dec. 8, 2017).

[4] Richard Susskind, *The Future of Law*, Clarendon Press Publication, 1998.

[5] Richard Susskind, "Susskind on the Evolution of Legal Services," The Am Law Daily, Oct. 10, 2017

[6] Kingsley Martin, Contract Maturity Model (Part 2): Technology Assembly Line—from Active to Passive Systems, June 16, 2016, http://legalexecutiveinstitute.com/contract-maturity-technology-assembly-line/ (last visited Dec. 8, 2017).

[7] An n-gram is a contiguous sequence of n items from a given sequence of text or speech, such as "hold harmless."

[8] Illustrations from US Patent 8606796: Method and system for creating a data profile engine, tool creation engines and product interfaces for identifying and analyzing files and sections of files.

[9] *Id* US Patent 8606796.

[0] Introduction to Natural Language Processing (NLP), Algorithmia Blog, Aug. 11, 2016, https://blog.algorithmia.com/introduction-natural-language-processing-nlp/ (last visited Dec. 8, 2017).

[1] Visualization provided using brat visualization/annotation software.

[2] What is Machine Learning? A definition, Expert System blog, www.expertsystem.com/machine-learning-definition/ (last visited Dec. 8, 2017).

[3] Tom M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[4] Faizan Shaik, Deep Learning vs. Machine Learning – the essential differences you need to know! Analytics Vidhya Blog, April 8, 2017, www.analyticsvidhya.com/blog/2017/04/ comparison-between-deep-learning-machine-learning/ (last visited Dec. 9, 2017).

[5] Ian Goodfellow, Yoshua Bengio and Aaron Courville, *Deep Learning*, MIT Press, 2016.

[6] Andrew Ng (www.slideshare.net/ExtractConf).

[7] Richard Susskind, *The End of Lawyers?* 2008.

[8] Kingsley Martin, *Contract Maturity Model (Part 3): Evolution of Content from One-Offs to Modular Components*, Legal Executive Institute Blog, July 20, 2016 (http://legalexecutiveinstitute.com/contract-maturity-modular-components/ (last visited Dec. 9, 2017).