

**PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
POLYTECHNIC INSTITUTE
COMPUTER SCIENCE GRADUATE PROGRAM**

**UNDERSTANDING CONTRACTS
IN NATURAL LANGUAGE**

DANIELE ANTUNES PINHEIRO

Thesis submitted to the Pontifical Catholic
University of Rio Grande do Sul in partial
fulfillment of the requirements for the
degree of Master in Computer Science.

Advisor: Prof. Felipe Rech Meneguzzi

**Porto Alegre
2019**

**REPLACE THIS PAGE WITH
THE LIBRARY CATALOG
PAGE**

**REPLACE THIS PAGE WITH
THE COMMITTEE FORMS**

ACKNOWLEDGMENTS

I would like to thank Pontifical Catholic University of Rio Grande do Sul, its professors, and administrators for the opportunity of studying for these last two years with all the support I needed.

I would like to thank my advisor, Felipe Meneguzzi, for choosing me, trusting in me and for his patient and support during the time of our research.

I would like to thank my parents, Roni and Rozane, for the support, for accepting and respecting my absence in some moments, and my grandmother, Nena, for showing me how to be strong no matter the situation.

I would like to thank my friends, Andressa and Aline for always being there, prepared to catch me; Pedro Gyrão for your indescribable loyalty; Ingrid Louise, for your everyday talk; and the ones who are with me even when they don't know, Candice Michelin, Diego Jornada, Matthias Nunes, Ricardo Piccoli, Rodrigo Chamun and Letícia Spezia.

And last, I would like to thank HP, not only for the scholarship but also for all my colleagues that became real friends and make my day happier and lighter, especially Cassio Cons and Vinicius Lafourcade.

INTERPRETANDO CONTRATOS EM LINGUAGEM NATURAL

RESUMO

Contratos são acordos entre pessoas ou organizações, chamados de partes. Geralmente são escritos em linguagem formal e são compostos por um conjunto de regras que devem ser seguidas pelas partes envolvidas nele. No processamento de contratos, é comum assumir uma etapa manual para extrair os componentes do contrato, o que é uma tarefa que exige tempo e geralmente é baseada em domínio específico. Considerando um cenário onde todos os dias há mais pessoas interessadas em processar o trabalho legal, uma ferramenta automatizada para extrair componentes contratuais é extremamente útil. Esta pesquisa definiu um método para extrair e formalizar automaticamente esses componentes, resultando em uma estrutura semântica útil para outros projetos. Para avaliar nosso trabalho, nós criamos um dataset com 15 contratos anotados e medimos a nossa acurácia em diferentes tipos de extração.

Palavras-Chave: Contratos, PLN, aprendizado de máquina, deep learning.

UNDERSTANDING CONTRACTS IN NATURAL LANGUAGE

ABSTRACT

Contracts are agreements between people or organization, called parties. They are usually written in formal language and are composed by a set of rules to be followed by the parties involved in it. In the processing of contracts, it's common to assume a manual step to extract the contract components to work with, which is a task that demands time and usually is domain based. Considering a scenario where every day there are more people interested in processing legal work, an automated tool to extract contractual components is extremely useful. This research defines an approach to automatically extract and formalize these components resulting in a semantic structure useful for other projects. To evaluate our work, we created a dataset containing 15 annotated contracts and measure our accuracy over different types of extractions.

Keywords: Contracts, NLP, machine learning, deep learning.

LIST OF FIGURES

Figure 2.1 – Example of contract structure	25
Figure 2.2 – Natural Language Processing tasks.	27
Figure 2.3 – POS tagging example.	29
Figure 2.4 – Noun, verb and prepositional phrase example.	29
Figure 2.5 – Constituency parse tree example.	31
Figure 2.6 – Dependency tree example.	31
Figure 2.7 – PP attached to the VP.	32
Figure 2.8 – PP attached to the NP.	32
Figure 2.9 – Feed-forward network.	34
Figure 2.10 – Recurrent neural network.	35
Figure 2.11 – Example of SyntaxNet output.	36
Figure 3.1 – Dependency tree example of a norm.	40
Figure 3.2 – The processing of a contract.	40
Figure 3.3 – Example of a graph for fairness assessment.	44
Figure 4.1 – Example of parse tree.	49
Figure 4.2 – Second example of parse tree.	49
Figure 4.3 – Example of a dependency tree.	54

LIST OF TABLES

Table 4.1 – Comparison of norm classification with range of threshold values.	50
Table 4.2 – Measures threshold values comparison for norm classification.	51
Table 4.3 – Threshold values comparison for first contract zone.	51
Table 4.4 – Threshold values comparison for deontic modality classification.	52
Table 4.5 – Threshold values comparison for action and condition classification. . .	52
Table 4.6 – Threshold values comparison for object and subject classification. . . .	52
Table 4.7 – Threshold values comparison for smaller sentences classification. . . .	53
Table 4.8 – Threshold values comparison from Aires et al. (2017a).	53
Table 4.9 – Threshold values comparison with SyntaxNet output.	55
Table 5.1 – Comparisons among related work.	60
Table 5.2 – Granularity table.	61

LIST OF ALGORITHMS

Algorithm 3.1 – Searches for the norm action.	42
Algorithm 3.2 – Calculates the semantic similarity between two norm actions.	42
Algorithm 3.3 – Identifies conflict between two norms.	43

CONTENTS

1	INTRODUCTION	21
2	BACKGROUND	23
2.1	NORMS	23
2.2	CONTRACTS	24
2.3	NATURAL LANGUAGE PROCESSING	26
2.3.1	SENTENCE SEGMENTATION	27
2.3.2	TOKENIZATION	27
2.3.3	LEMMATIZATION AND STEMMING	28
2.3.4	PART OF SPEECH TAGGING	28
2.3.5	NAMED ENTITY RECOGNITION	29
2.3.6	RELATION EXTRACTION	30
2.3.7	PARSE TREE	30
2.4	MACHINE LEARNING	33
2.5	SYNTAXNET	36
3	UNDERSTANDING CONTRACTS IN NATURAL LANGUAGE	39
3.1	EXTRACTING CONTRACT COMPONENTS	39
3.2	APPLICATIONS	41
3.2.1	CONFLICT IDENTIFICATION	41
3.2.2	CONTRACT FAIRNESS EVALUATION	43
4	EXPERIMENTS	47
4.1	EXPERIMENTS	47
4.1.1	DATASET	47
4.1.2	COMPONENT EXTRACTION	48
4.1.3	CONFLICT IDENTIFICATION BY ACTION COMPARISON	53
5	RELATED WORK	57
6	CONCLUSION	63
	REFERENCES	65

ATTACHMENT A – Contract example	71
ATTACHMENT B – JSON output example	77

1. INTRODUCTION

In our society, cooperation between humans is mainly based on rules (Fehr and Fischbacher, 2004). Whether in our personal or professional lives it is common to base these rules upon expectations, values, and behaviors (Axelrod, 1986). These rules exist to regulate and to guarantee expected behaviors from us, sometimes describing punishments when something out of the ordinary happens (Axelrod, 1986). They are called norms and their goal is to avoid conflicts among people. There are social norms about nearly every aspect of human behavior (Sunstein, 1996). Norms can be either informal, when there is no formal regulation of the norm, or they can be formal when they are incorporated in laws or regulations from the institutions that regulate the behavior of the people within the society (Dignum, 2002). When these rules are formalized, they are written in a contract and signed by the people involved in it. That forms the definition of contract: an agreement between people, called parties, with rules to be followed, called norms.

A contract is an essential part of any transaction or business agreement between people or organizations. This written agreement with a combination of norms specifies everything that is expected by the parties and because of that it's important that they are clear and fair to each party involved in the contract. Thus, the study and modeling of norms have attracted the interest of scientists from different disciplines such as sociology, economics, philosophy, and computer science (Dignum, 2002). Within contracts, these norms exist to ensure compliance with specific rules.

The contracts used to be written and signed in paper, but with the advance of our digitized world, they are now electronic. And even though electronic contracts used by agent organizations are already formalized in machine processable mechanisms (Meneguzzi and Luck, 2009; Modgil et al., 2009), contracts in natural language continue to be generated, and its revision is a labor intensive process that requires much human effort. The processing of such contracts needs to have a clear characterization of who the parties are, what are they committed to do, who owes what to whom, what are the actions committed, etc. This processing has generated much research (Prisacariu and Schneider, 2007; Karlapalem et al., 2001). Since representing contract elements to find conflict within its norms, the amount of people working towards solving contracts structure and processing its content is growing every day.

Alongside with the research about contracts is the Natural Language Processing (NLP) area. NLP is a subfield of Artificial Intelligence and its main goal is to make computers understand human language (Russell et al., 1995). For that, there are lots of syntax techniques used as well as semantic ones. The field of NLP is also growing with a large amount of data being produced. It is a challenging area, due to the ambiguity of languages and the huge number of unstructured data to process.

The processing of contracts relies on NLP techniques. To process a contract is to make a series of actions in order to achieve a particular goal. For instance, a contract's norm may establish the opposite condition of previous one, what could generate a conflict inside the contract; or a contract may be more fair to one party than the other. The process of finding these situations requires a mechanism to clearly identify syntax and structural elements that provides means to identify these problems and this the motivation of this work. But to extract a contract's elements it is necessary to understand its structure as well as its context. To be able to find such elements its important to have a clear definition of who the parties are, what are the contract's norms, what is the commitment, what are the actions committed and who owes what to whom. In order to automate the processing of contracts in natural language, we need some a semantic representation of the contract, and a means to formalize this semantic representation from the natural language text.

Although there is much research on deontic logic and reasoning about contracts in formal languages, there are few efforts in bridging the gap between natural language contracts and systems that attempt, for example, to detect conflicts between the norms within a contract. Most of the work in this area assumes a manual step to separate the contract components they want to work with or to create their own extractor, which is a task that also demands time and is usually specific to a domain. We aimed to avoid such inefficient preprocessing steps by developing a semantic representation and an approach to extract information from text into that representation. In a scenario where every day there are more people interested in processing legal work, an automated approach to extract contractual components is extremely useful.

We researched an approach to formalize contracts written in natural language using SyntaxNet, the Google NLP framework. We defined a formalism generic enough to encode most forms of natural language contract, resulting in a semantic structure useful for various reasoning tasks regarding the contract processing. One of the main advantages of our approach is that it will be suitable for other people's research, such as Curtotti *et al.* (Curtotti et al., 2013), Gao and Singh (Gao et al., 2012), Curtotti and McCreath (Curtotti and McCreath, 2010), and Aires (Aires et al., 2017a), since it provides a contract with its element extracted and ready for use. To evaluate our approach we created a dataset of 15 annotated contracts, with 1217 norms. This dataset is available for download and could be used to train algorithms for other approaches.

2. BACKGROUND

This chapter describes the main concepts around contracts and norms, as well as the Natural Language Processing (NLP) area, its applications and techniques used for processing contracts. We introduce NLP approaches that will be the base to understand the processing of contracts and its challenges. Works such as Gao *et al.* (Gao et al., 2012) and Gao (Gao and Singh, 2014) strongly relies on NLP techniques to succeed in their approaches, while Griffo *et al.* (Griffo et al., 2017) defines what are the components of a contract and how to model them. In this chapter, we introduce the main concepts of Machine Learning, with approaches used for NLP.

2.1 Norms

Coexistence in society depends on each human being respecting each other. Our attitudes and actions are often limited by rules, whether they are merely common sense or formalized in a paper. These rules are represented by norms to regulate our social behavior. They describe what is expected under general events and what should happen in case of an unexpected event (Dignum, 2002). Therefore, norms reflect expectations about attitudes and behaviors (Cannon et al., 2000) and they exist to avoid conflicts in social groups, often determining punishments if an individual does not act as expected (Axelrod, 1986).

When a norm is created, it is often related to at least one subject, which is the member who needs to follow the behavior described. This member is called *party* and it is the mainly interested in the rules described in the norms. The parties can be people, organizations, countries, etc.

Norms are classified by the type of restrictions they have. Among the common classifications, there are *prohibition*, *permission* and *obligation*. A norm defined as a prohibition declares a behavior that must not happen. A norm defined as an obligation is the opposite: declares a behavior that must happen. And a norm defined as a permission declares a behavior that can or not happen. These classifications are often represented with Deontic Logic (Dignum, 1999). Also referred as logic of obligation and logic of norms (Føllesdal and Hilpinen, 1970), the Deontic Logic is the field of philosophical logic that studies the obligation, permission, and related concepts as features of action.

A modality is the expression of the quality or state of a given subject. Deontic modality refers to the concepts of Deontic Logic, describing representations of worlds and its violations (Von Wright, 1951). The deontic modality within the norm can be identified by the modal verb present in the norm, such as permission with *may*, obligation with *must* and prohibition with *can not*.

The understanding of a contract relies on the understanding of its norms since they describe the rules of the agreement. They are described in the most part of the contract, leaving the remaining parts responsible for describing the parties' information. To give meaning to norms it is important to identify the parties and the relation between them. According to works (Griffo et al., 2017; Gao et al., 2012; Aires et al., 2017a) it is possible to find other items inside the norms structure that helps to interpret its meaning. They are:

- **Deontic expression:** The Deontic expression categorizes a norm into permission, prohibition, and obligation. For example, the norm "Alara must open and maintain a bank account" expresses an obligation, characterized by the modal verb *must*.
- **Conditions:** Conditions are restrictions to a specified circumstance. In the example: "Alara must open and maintain a bank account unless it is too expensive", the condition is "unless it is too expensive".
- **Object and subject:** The object of a norm refers to the entity that is acted upon by the subject. Example: "Alara must open and maintain a bank account", where "Alara" is the subject and "bank account" is the object.
- **Actions:** The action refers to the entity that represents what the party should do. Example: "Alara must open and maintain a bank account", where "open and maintain a bank account" is the action.
- **Contrary-to-duty:** Contrary-to-duty are norms to repair other norms' violation. They happen in situations where there are two or more obligations in a norm, but the contrary-to-duty only happens if a first norm is violated. Example: "Alara must open and maintain a bank account. If not, she must be paid in cash".

The study and formalization of norms have attracted the interest of scientists from different disciplines (Dignum, 2002). In computer science the goal of this type of research is to provide a more certain decision making and to guarantee less human effort in the process of text. In the next section we introduce the concept of contracts.

2.2 Contracts

Bessone (Bessone, 1987) defines contract as "an agreement between two or more persons to constitute, regulate or extinguish a legal relationship of a patrimonial nature". Contracts serve to regulate and guarantee behaviors, whether of people or organizations. They are formed by parties and norms, as described in Section 2.1.

Contracts formalize what each party should expect from an agreement. They used to be written and signed in paper, but nowadays it is common to have them in a digitized

form. Electronic contracts are modeled, controlled and monitored by a software system (Karlalalem et al., 2001). They are usually written in formal language, and they follow the same purpose as contracts: to regulate an agreement among parties. But the fact that they are digitized means that the access to its content is much easier and faster, which increases the amount of research regarding it (Prisacariu and Schneider, 2007). With the amount of data being produced, it is not suitable to submit humans to validate contracts anymore. Because of that the interest to interpret contracts is growing and tools to analyze and validate contracts are being created more and more.

Although the structure of a contract may change, we show in Figure 2.1 an example, following the Formation Agreement - DreamWorks Animation SKG Inc. and DreamWorks LLC¹ contract from Onecle repository.

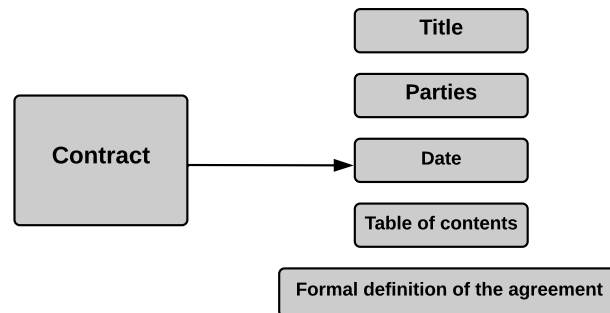


Figure 2.1 – Example of contract structure

The Formation Agreement is attached in file A at the end of this paper as an example. Part of its content was removed in order to better visualize its structure.

¹<http://contracts.onecle.com/dreamworks/dreamworks.form.2004.10.shtml>

2.3 Natural Language Processing

Natural Language Processing (NLP) is a subfield of Artificial Intelligence that studies how a computer interprets and understands human language (Russell et al., 1995). Our choice of words is often unconscious and information such as a person's tone of voice or the knowledge of the place a conversation is happening are intrinsic for humans to process and to interpret. A computer does not have that unless it is categorically given to it, so it needs lots of information to correctly process the meaning of a sentence or text. Words can have different meanings in different contexts (Fellbaum, 1998). Contexts can be implied and changed during a conversation. These are some challenges the Natural Language Processing field addresses with growing researches nowadays more than ever, mainly due to the advance of social media and the use of electronic communication devices (Liu, 2012).

Applications of NLP include a number of fields of studies, such as machine translation, text mining, summarization, information retrieval, sentiment analysis, speech recognition, and so on (Chowdhury, 2003). Machine translation (Somers, 1999) is the area that studies the translation of text for other languages, respecting the grammatical rules. Text Mining (Tan et al., 1999) is the process of structure data and finding patterns in it. It is different from Information Retrieval (Frakes and Baeza-Yates, 1992), which is the study of searching for specific information in a document or a document itself. Summarization is the field (Hu and Liu, 2004), as the name suggests, of summarizing the main content of a text. Speech recognition is the process of translate voice to words (Rabiner and Juang, 1993). Sentiment Analysis (Liu, 2012), also known as Opinion Mining, is the study of the affective side of a text, which deals with subjective information, like irony.

The more our society produces data, personally and within organizations, the more the amount of research regarding natural language grows. The increasing number of electronic data brought the need of tools able to process it. Nowadays there is a good deal of tools and APIs with features dedicated only to processing text. Among the most famous ones, there are NLTK (Natural Language Toolkit) (Bird, 2006), a Python tool that provides an interface with several lexical resources, including WordNet (Miller, 1995). The Stanford CoreNLP (Manning et al., 2014) provides a set of NLP techniques, with an API for common programming languages. More recently, Google launched its own NLP framework, called SyntaxNet² and Sling (Ringgaard et al., 2017), a parser for natural language based on semantic frames.

Although the contract is based on a structure, showed in section 2.2, the processing of its contents strongly relies on NLP techniques. Since parsing the text into sentences to find the grammatical class to each word, every technique has its role when trying to give mean to a contract. We introduce in the next sections some of these tasks, as well as NLP

²<https://github.com/tensorflow/models/tree/master/research/syntaxnet>

concepts that will be important for the research proposed in chapter 3. Figure 2.2 shows these tasks as a pipeline.

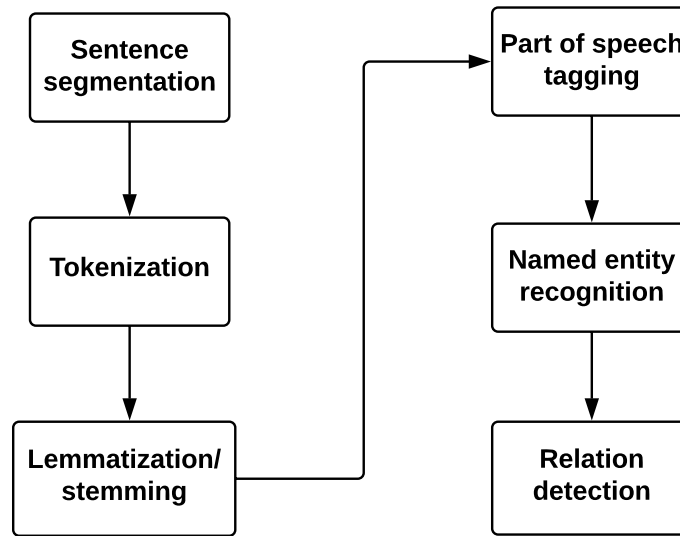


Figure 2.2 – Natural Language Processing tasks.

2.3.1 Sentence segmentation

One of the first steps to process a text is to understand its structure. It is common to find texts without any grammar punctuation and a large amount of unwanted information on it. This is the case of *Twitter* datasets, for example. As it consists of a set of small texts, the *tweets*, these datasets usually need a preprocessing to remove all unwanted information before the start of the work. The sentence segmentation task helps to create a structure for the text since is responsible for extracting the sentences from it. The main challenge is to find the correct location to break the sentence, the boundary, which usually is a punctuation (Cambria et al., 2017).

In contracts, the sentence segmentation task helps to break the contract into pieces, which makes it easier to give the correct meaning to each piece. For example, the sentence segmentation is the first step to find norms in a contract.

2.3.2 Tokenization

In NLP, *token* is any segment of text or symbol that has a meaning, which can be words, numbers, and punctuation. Tokenization is the process of splitting a text into these basic units, the *tokens*, with the objective of finding some pattern that can be used for further

processing. An example of tokenization of the sentence “The contract is an agreement.” will be “The”, “contract”, “is”, “an”, “agreement”, “.”. These *tokens* are useful to find key words in sentences, such as modal verbs or references to subjects.

2.3.3 Lemmatization and Stemming

Given a word w , lexical normalization is the task of removing all abbreviations or spelling errors, transforming the word into its standard form (Han et al., 2013). Lemmatization and Stemming are part of lexical normalization. Together, these techniques eliminate variations of gender, number and verbs forms. Lemmatization is the task of finding the basic form of a word, the *lemma*. For example, the variations “am, are, is” after lemmatized would be just “be”. It transforms nouns and adjectives into their singular and masculine forms and verbs into their infinitive forms. It is most used in Latin languages, such as Portuguese because it has an extensive vocabulary and more distinct forms of words within the same family.

Stemming is the process of reducing a word into its base, the largest common part shared by morphologically related forms of a word (Ingason et al., 2008), which is called *stem*, removing all affixes. For example: the stem of “different” is “differ”.

2.3.4 Part of speech tagging

Part of speech tagging (POS tagging) may be the most important task of Natural Language Processing because it is responsible for attributing a label with grammatical information to each word of a given text. This label is the part of speech of the word, which can be noun, verb, pronoun, preposition, adverb, conjunction, participle and article, for example. The correct label can define if the meaning found in the text is right, once words can have a different meaning when used as nouns and when used as verbs. For example, the word *season*: when used as a noun it means a period of the year (spring, summer, fall, winter), but when used as a verb means to apply spices or flavorings to food.

POS tagging is also known as word classes or syntactic categories (Jurafsky and Martin, 2014), and its use influences other tasks, such as stemming and named entity recognition (explained below). The POS tagging itself gives a large amount of information about a word and its neighbors since there are some rules that are usually followed. For instance nouns are preceded by determiners and adjectives, and verbs by nouns. Figure 2.3 shows an example of POS tagging, where the tag *NNS* is noun (plural), *VBP* is verb (non-3rd person singular), *IN* is a preposition.

Contracts/**NNS** are/**VBP** agreements/**NNS** between/**IN** people/**NNS**

Figure 2.3 – POS tagging example.

A sentence expresses a complete thought. It is composed by a group of words that has a subject and a predicate, which is a verb or verb phrase. The concepts of verb phrase (VP), noun phrase (NP) and prepositional phrase (PP) represent an important role when parsing a sentence, and trying to understand a text by its syntactic structure. This structure can be represented in a parse tree (explained below). A noun phrase is composed by a noun or a determiner followed by a noun while a verb phrase is composed by a verb or a verb followed by a noun phrase (Matthews, 2007). The prepositional phrase starts with a preposition followed by and object, and function as adjectives or adverbs (Pantel and Lin, 2000). Figure 2.4 shows an example of it.

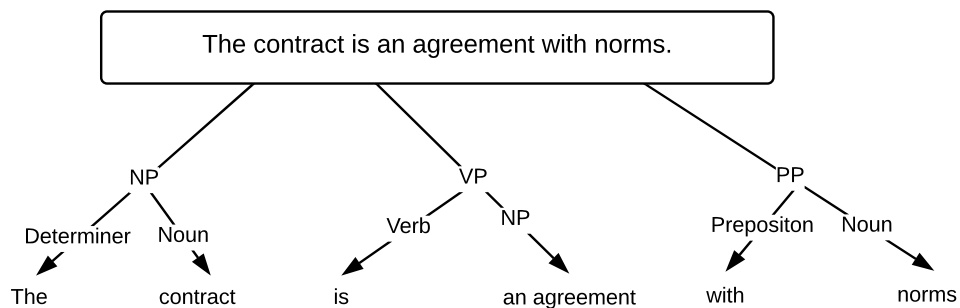


Figure 2.4 – Noun, verb and prepositional phrase example.

2.3.5 Named Entity Recognition

Named Entity Recognition is used to identify the names of things, such as a person, an organization, a country, a location. This task (also called entity extraction) is a tagging task, similar to POS tagging. It identifies the proper nouns of a sentence and separates them into predefined categories. For example, in the sentence “The tree is green”, the entity is tree.

In the context of contracts, this is an extremely important task to find the entities involved in the agreement. Contracts are related to at least one entity, which is the one (or the ones) referenced in the entire contract. If this identification is done wrong, the entire processing of the contract will be wrong.

2.3.6 Relation extraction

Once you have an entity identified, it is time to understand what is the role it is playing. For that, relation detection is used. This task identifies the semantic relations between entities. One of the approaches used for this detection is the anaphora resolution, which studies the linguistic event of an expression pointing back to another (Reinhart, 2016). Finding these expressions references allows to also find the relation between the entities, for example, “employed by”, “part of”, “married to”.

2.3.7 Parse tree

Parse tree or syntax tree is the representation of a string in a structure according to a grammar. A grammar is a set of structural rules usually divided into two groups: *morphology*, the study of how words are formed by smaller units, and *syntax*, the study of how words form larger units such as phrases and clauses (Dixson, 2002). In this dissertation we use the English grammar and its definitions of verb, noun, verb phrase, noun phrase, prepositions, etc.

The use of parse trees is an important task in NLP because the structure of words and its relations helps to fill the gap between linguistic expressions and the meaning of a sentence (Socher et al., 2013). Parse trees built considering the words in the sentence, from left to right, each word class and with the sentence itself being the root of the tree are called constituency parse tree (Covington, 2001).

Figure 2.5 illustrates a constituency parse tree for the sentence “I booked a flight from LA” following the English grammar. In this tree VP stands for verb phrase, NP for noun phrase, Nom for Nominal PP and PP for preposition phrase. S is the root of the tree, the sentence itself.

A parse tree built considering the relationship between words and how they connect within the sentence is called dependency tree (De Marneffe et al., 2006). Figure 2.6 illustrates an example of it. With a dependency tree we can find the piece of text that are the subject of a sentence and how it relates to the other words. In this example, booked is the verb connected to the subject “I” while “flight” is the object connected to “booked”. “From LA” is the prepositional phrase dependent of the object “flight”.

After parsing a sentence, we can obtain many dependency trees because the sentence can have many structural ambiguities (Kim et al., 2001). As we mentioned in Section 2.3.4, a word can have a different meaning when used as a noun and when used as a verb, and this definition interferes with the tree composition. There is a common problem called prepositional phrase attachment ambiguity (Nadh and Huyck, 2009), which happens

Felipe

Citation

Felipe

How can the use of something be a task?

Felipe

This sentence is ungrammatical, revise it.

Felipe

I want to see this grammar, with the production rules and all..

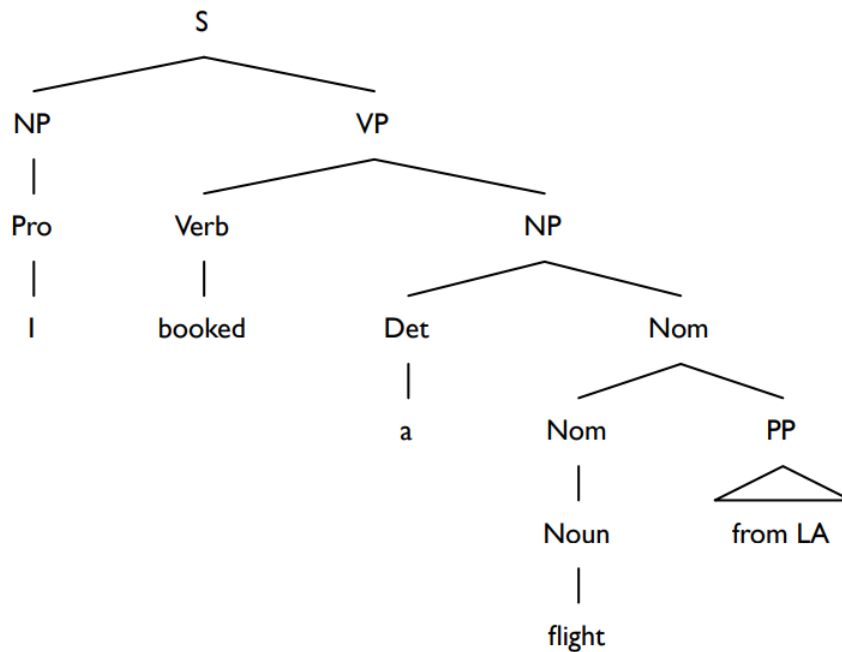


Figure 2.5 – Constituency parse tree example.

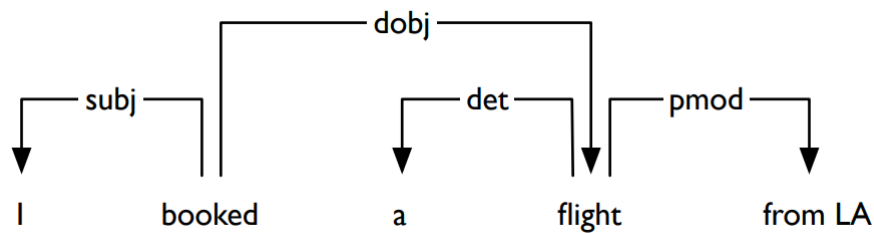


Figure 2.6 – Dependency tree example.

when the prepositional phrase (PP) is attached directly in a verb phrase (VB) of a sentence instead of the noun phrase (NP) or vice versa. As an example, the sentence “I saw the girl with the telescope.”, illustrated in Figure 2.7, from (Nadh and Huyck, 2009), attaches the PP in the VP, which transforms the telescope in the instrument of the sentence. Another possible dependency parsing of this sentence (illustrated in Figure 2.8) is to attach the PP into the NP, which makes the telescope modifies the girl and the meaning to “the girl possesses the telescope”

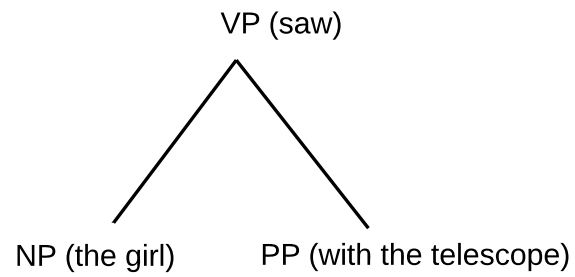


Figure 2.7 – PP attached to the VP.

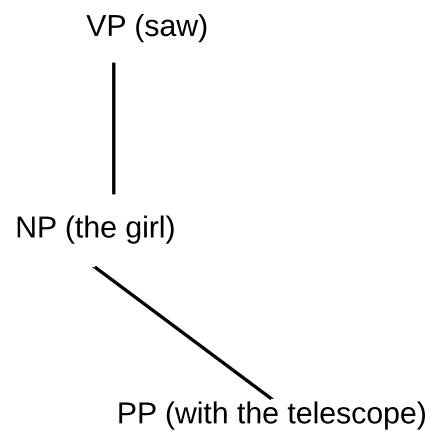


Figure 2.8 – PP attached to the NP.

2.4 Machine Learning

Machine Learning is the field of Artificial Intelligence responsible for transform amounts of data into information. While NLP tries to make computers understand human language, Machine Learning tries to make computers learn new information without human intervention. It is used to automate tasks and to discover insights and connections from data that a person would take a long time to discover or not discover at all. The use of machine learning systems is important to help decision making and to automatically apply new actions. Nowadays they are commonly found in recommendations systems, self-driving cars, filter for content in social media or to understand what customers are saying about a product. This area of research is growing due to the increasing volume of data being produced, the advance of computational processing and the affordable data storage (Minelli et al., 2012).

Machine learning is often divided into three parts: supervised, unsupervised and reinforcement learning (Hyvärinen et al., 2010). Supervised learning approaches are the ones where the process of learning is based on a set of rules from instances (Kotsiantis et al., 2007). These instances can be called training sets, they are examples with the correct output labels for a further classification. This approach requires a large dataset, so the classifier has enough examples to classify new instances. On the contrary, in the unsupervised learning, this training set does not exist. This approach is closely related to data mining, where there is no labels, just a set of information with connections that need to be discovered (Hyvärinen et al., 2010). Lastly, reinforcement learning relies on a trial and error system and it is often used for robotics, gaming, and navigation. This type of learning has three primary components: the agent (the decision maker), the environment (everything the agent interacts with) and actions (what the agent decides to do). The goal is to take actions in order to maximize the reward (Kaelbling et al., 1996).

Although machine learning approaches have achieved plenty of accomplishments (Pang et al., 2002; Ikonomakis et al., 2005; Nigam et al., 2000), they were limited in their ability to process natural data in their raw form due to the increasing number of features that needed to be evaluated (LeCun et al., 2015). Represent the knowledge of features to feed a classifier became an expensive task, performance wise. Therefore, another machine learning approach emerged to solve the problem of knowledge representation and processing: deep-learning. Deep-learning technique is a representation-learning method with multiple levels of features representation (LeCun et al., 2015) applied to large amounts of data. This allows for a more complex set of features to be represented from one raw input since each level of representation is responsible for a piece of knowledge. In the context of text processing, word embedding helped the performance of deep learning methods achieve a new level due to its form of representation. A word embedding is a learned representation for text where words that have the same meaning have a similar representation. Each word is asso-

ciated with a feature vector that represents the different aspects of the word, which makes the number of features smaller than the size of the vocabulary that is being used (Bengio et al., 2003).

Deep learning technique uses a model inspired in the human brain (Gupta, 2013; Goodfellow et al., 2016) for processing information: artificial neural networks. They allow the computer to have massive parallelism, distributed computation and inherent contextual information processing (Jain et al., 1996) while dealing with data inputs. Among the classifications for artificial neural networks there are *Recurrent Neural Networks* (RNN) and *Feed-forward Neural Network* (FNN). They are named after the way they process information through their nodes, with a series of mathematical operations among them. A standard neural network (NN) consists of many simple, connected nodes called neurons, each producing a sequence of activations (Schmidhuber, 2015), which are responses to the input they receive. The neurons have weight, defining the importance of the data they process, and they are separated in a given number of layers. In the case of feed-forward network the input is received by the network and processed, transforming it into an output, with no recurrent process or need to understand the information from the previous node (Fine, 2006). Figure 2.9 shows an example of a feed-forward network architecture³. Convolutional Neural Networks (CNN) are a feed-forward network with a specific layer for convolving filters that are applied to local features (Kim, 2014). They are most used in the processing of images (Krizhevsky et al., 2012), but the NLP field has used for text processing as well (Kalchbrenner et al., 2014; Kim, 2014).

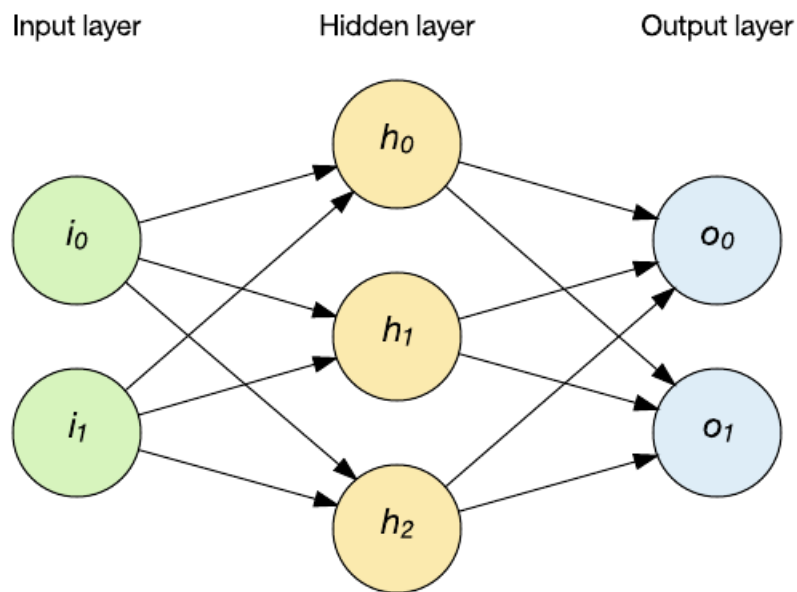


Figure 2.9 – Feed-forward network.

³Image from: <https://goo.gl/GCEh9d>

Recurrent Neural Network (RNN) are considered the deepest of all neural networks (Schmidhuber, 2015), as they are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations. Figure 2.10 represent an example of RNN, with the hidden layer output applied back into the hidden layer⁴. Long Short-Term Memory (LSTM) is a specific RNN architecture that was designed to model temporal sequences and their long-range dependencies more accurately than conventional RNNs (Sak et al., 2014). LSTMs help preserve the error that can be propagated through time and layers. In the NLP research, RNNs have been found to be very good at predicting the next character in the text or the next word in a sequence (Schmidhuber, 2015).

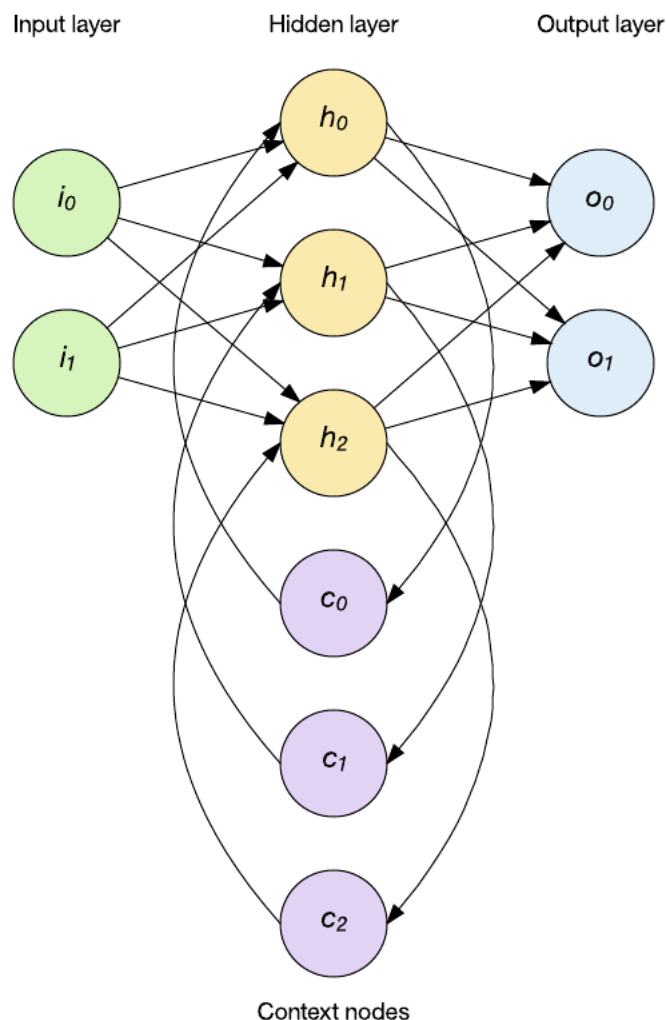


Figure 2.10 – Recurrent neural network.

⁴Image from: <https://goo.gl/GCEh9d>

2.5 SyntaxNet

The processing of contracts relies on NLP techniques when it comes to giving meaning to its sentences. To extract a contract content it is necessary to understand its structure as well as its context, such as who are the parties involved and what they are committed to do. In order to automate the processing of contracts in natural language, we need a formalization that captures the semantic of a contract in a computer processable way. To start this process, we choose to use Google's NLP Framework, SyntaxNet, due to its reliable structure of parsing a sentence according to the meaning of each word.

SyntaxNet is a syntactic parser released by Google in 2016 (Petrov, 2016). It is an open-source neural network framework implemented in TensorFlow⁵ that provides a foundation for Natural Language Processing systems. From an input sentence SyntaxNet analyses the linguistic structure of a given language and describes the syntactic function of each word in the sentence. Then, it infers the relationships between words, representing them in a syntactic dependency tree, which allows the detection of the underlying meaning of the sentence that is being processed. For example, take the sentence: "Alice drove down the street in her car"⁶. Figure 2.11 shows the output from SyntaxNet, characterizing each word part of speech and how it relates to each other.

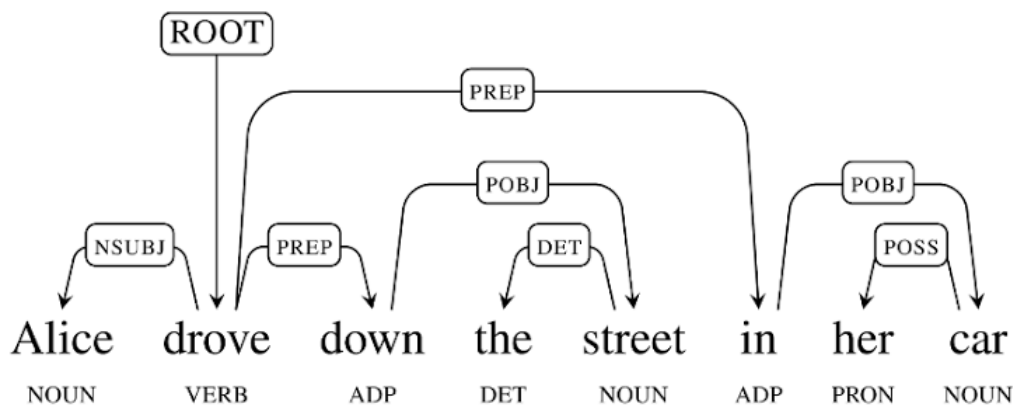


Figure 2.11 – Example of SyntaxNet output.

SyntaxNet processes the words in input sentences from left to right, incrementally adding all the possible dependencies between the words as each word in the sentence is processed (Andor et al., 2016). The same words could have different meaning depending on the context they are in, which is the ambiguity problem, a common challenge when dealing with text. SyntaxNet deals with it using neural networks to consider all the possible dependencies a sentence could have and the meaning a word could have. The combinations of words and its meaning and the order they are used, are considered as possible

⁵<https://www.tensorflow.org>

⁶Example from: <https://ai.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html>

dependencies trees. SyntaxNet computes a score for each possible dependency choosing the dependency structure with the highest score after testing all combinations. This score is calculated based on the plausibility of the combination of words in the input compared with the previously trained data. SyntaxNet is already trained with a model in the English language and this model is responsible for defining the score each possibility gets. After all the scores are calculated, the parse tree is generated for a given sentence. Although the sentence from the example in Figure 2.11 can be parsed into more than one structure, it is possible to see that the parser correctly defines “in her car” as the preposition object of “drove down the street”.

Within a contract, its components can be structured in different forms. In this way, norm processing benefits from using a parser such as SyntaxNet that consider all possible parse trees a sentence can have. We can have a more reliable result about what are the objects of a sentence and how they relate to the party or parties in it. We explain in the next section the work we develop using SyntaxNet to process contract components.

3. UNDERSTANDING CONTRACTS IN NATURAL LANGUAGE

In this chapter we describe our research contributions. We start with the formal structure we aim to extract from contracts in Section 3.1, which we design to be as generic as possible allowing multiple contract processing tasks to be carried out; and follow, in Section 3.2, with two use cases that show the practical applications with which we can process documents structured according to our formalization.

3.1 Extracting contract components

Most work on processing natural language contracts assumes a manual step to separate contract components or need to create their own extractor (Chalkidis et al., 2017; Aires et al., 2017a). This step is time-consuming and is usually domain based. In this dissertation we develop an approach to automatically extract and formalize these components to avoid such inefficient preprocessing steps. Our resulting approach defines a mapping from natural language parse trees into the norm formalization of Section 4.1, which allows us to process contracts based on their language structure.

Our approach extracts semantic components by dividing the contract into two zones. The first zone (the *title zone*) corresponds to the beginning of the contract that usually contains its title and its parties. The second corresponds to the contract's content that contains all the norms. We divided the zones into sentences, using the NLTK Sentence Tokenizer¹ and analyzed the parse tree of each sentence. The parse tree (Section 2.3) represents the syntactic structure of a sentence. The type of parse trees we chose to work with is dependency trees because it provides us with information about how a word relates to other in a sentence (Nivre, 2005).

Using a dependency tree, we map the structure of a norm as follows:

- **Party:** proper nouns (*NNP*) that match the ones coming from the first zone;
- **Action:** subject (*NSUBJ*, *DOBJ*) connected to the root verb of the sentence;
- **Condition:** prepositional phrases (*PPS*, *PS*, *PCOMP*) outside the action;
- **Subject:** party outside the action;
- **Object:** party inside the action.
- **Deontic modality:** the word classified as modal verb *MD*.

¹<https://www.nltk.org/api/nltk.tokenize.html#module-nltk.tokenize.punkt>

Figure 3.1 illustrates a dependency tree from the norm “From now on, the Company shall provide to the Customer written notification of any changes to the fees.”, where the word “provide” is the root of the sentence and dependent of “Company”. “Provide” is the beginning of text classified as the nominal subject of the norm. The prepositional complement “From now on” is connected to the root verb, indicating a condition statement.



Figure 3.1 – Dependency tree example of a norm.

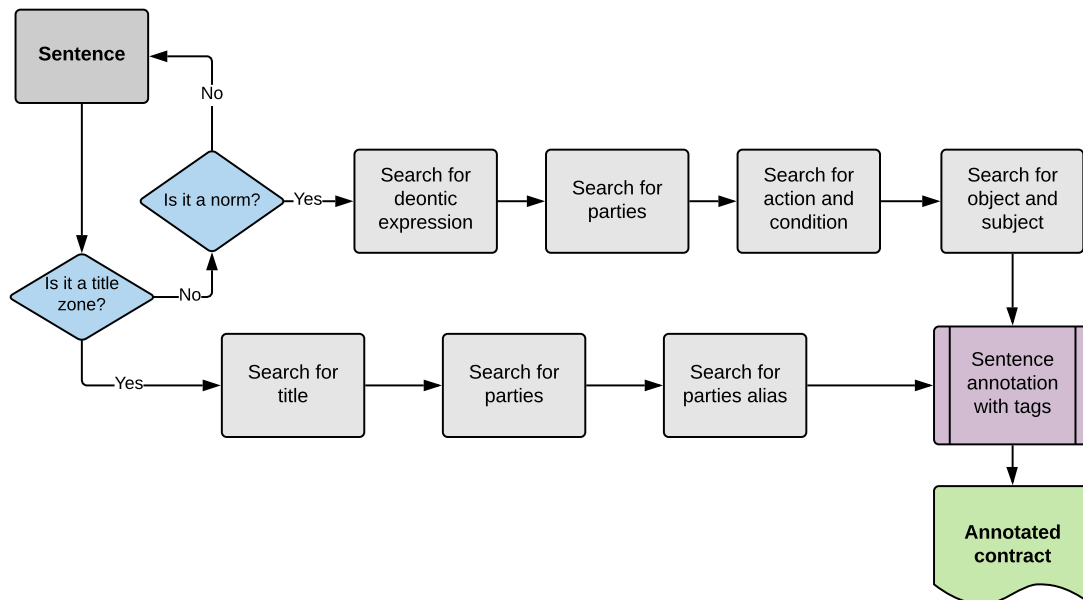


Figure 3.2 – The processing of a contract.

Figure 3.2 illustrates our approach’s process. As the contract is already divided into sentences, we start by checking if the sentence belongs to the title zone. We decided to define the title zone to be in the first 5 sentences of the contract. From the 15 contracts we had as our sample, all of them had the title and the parties explained in the first sentence. The decision to count the first 5 sentences was to have a range to work with in the case of a new pattern of contract. If the sentence is in the title zone we search for the title, the parties, and the parties aliases. For this we used a rule-based algorithm that searches for keywords such as ‘agreement’ to find the title and for key characters such as ‘(" ’)’ to find the party aliases. We chose for this method because most of the contracts follows a pattern, for example: *“CUSTOM MANUFACTURING AGREEMENT This Custom Manufacturing Agreement (this "Agreement") is entered into between Johnson Matthey Inc., having an office at 2003*

Nolte Drive, West Deptford, NJ 08066 ("JMI"), and Celgene Corporation, having an office at 7 Powder Horn Drive, Warren, New Jersey 07059 ("Celgene")".

If the sentence is not in the title zone, we check if it is a norm by searching for a modal verb. If we find the modal verb, we save it as the deontic expression of the norm and start the search for other components. The next search tries to identify if there is a party in the norm. If we find NNPs (proper nouns), we match them with the parties and the aliases found in the title zone to define if they are parties or not. The action and the condition are the following components to be searched; and last, we check if there is a party inside the action to be the object and if there is one in the component related to the action, to be the subject.

We followed this order of processing due to the contract and the norm structure. To process a norm and search for its parties, we must have that information coming from the title zone. And as we defined that the subject and the object of a norm must be a party, to search for them, we must have already found the parties in the previous step of our process.

This process yields a structured document that marks the contractual and normative functions of all textual sequences from the input document. As we show in Section 3.2, such structure allows us to perform a number of contract-processing tasks that rely on the relations between the semantic components of the contract. It is possible to process conflict identification between norms through the comparison of the actions in the sentence. The definition of norm components can be the base to a mechanism to process the fairness of a contract. A structured document with the contract components can serve as a training set for a supervised machine learn algorithm or to train new models for a deep learning approach.

3.2 Applications

3.2.1 Conflict identification

The first application of our structured document format is on the detection of norm conflicts. The key challenge for this task is identifying the components of each norm in a contract so that one can compare with the approach proposed by Aires et al. (2017a).

Aires et al. (2017a) assumes that two norms are conflicting if they have opposite modalities, the same parties, and the same norm action. The comparison of norm elements uses concepts of deontic logic and language similarity to identify corresponding information in norm pairs that may produce a conflict. In their approach, the action is considered to be all the words after the modal verb.

The comparison of norm actions is made through the calculation of a degree of semantic similarity between them, which considers a threshold that indicates when the se-

Algorithm 3.1 – Searches for the norm action.

Require: *dependency_tree*

```

1: procedure SEARCH_ACTION(dependency_tree)
2:   for node in dependency_tree do                                ▷ Iterate over tree nodes
3:     if node=subj then  $\sigma_1 \leftarrow \text{node\_children}$ 
4:   return  $\sigma_1$ 

```

Algorithm 3.2 – Calculates the semantic similarity between two norm actions.

Require: $|\sigma_1| \leq |\sigma_2|$

```

1: procedure COMPUTE_SIMILARITY( $\sigma_1, \sigma_2$ )
2:   for ind1 in  $\sigma_1$  do                                          ▷ Iterate over the indexes of  $\sigma_1$ 
3:     if  $\sigma_1[\text{ind}_1] = \sigma_2[\text{ind}_1]$  then sim  $\leftarrow \text{sim} + 1$ 
4:     else
5:        $\text{sim}_{\text{max}} \leftarrow -\infty$ 
6:       for ind2 in  $\sigma_2$  do
7:         if  $\sigma_1[\text{ind}_1] = \sigma_2[\text{ind}_2]$  then
8:            $\text{sim}_{\text{max}} \leftarrow 0.7$ 
9:           break
10:        else
11:          s1  $\leftarrow$  synonyms of  $\sigma_1$ 
12:          s2  $\leftarrow$  synonyms of  $\sigma_2$ 
13:           $\text{sim}_{1,2} \leftarrow \max_{s_1, s_2}(\text{SIMILARITY}(s_1, s_2))$ 
14:           $\text{sim}_{\text{max}} \leftarrow \max(\text{sim}_{\text{max}}, \text{sim}_{1,2})$ 
15:        sim  $\leftarrow \text{sim} + \max(0, \text{sim}_{\text{max}})$ 
16:   return sim/MEAN(len( $\sigma_1$ ), len( $\sigma_2$ ))

```

semantic distance is high enough to be considered similar or not. Using our approach, the actions are acquired as showed in Algorithm 3.1. From a dependency tree of a norm, every node is compared to be a “NSUBJ” or “DOBJ” (line 3). When the node is found, all its children are saved into σ_1 . The result (line 4) is the text considered as the action of the norm.

The action returned by Algorithm 3.1 is the input of Algorithm 3.2. The semantic similarity algorithm takes two parameters σ_1 and σ_2 , the word-vector corresponding to the actions of two norms. The comparison of these actions consists of iterating over the indexes of each word in both word-vectors (line 2). If both vectors have the exact same word in the same position it adds 1 to the final score (line 3). Otherwise, it compares the word in the first vector to every other word in σ_2 (line 6). If the same word is found in a different position, it adds 0.7 to the final similarity score (line 8) representing a penalty for the different positions of the same word. If the same word is not found in a different position, the algorithm tries to compare the similarity between the synonyms of both words being compared (lines 11 and 12) keeping the highest semantic similarity between them. The semantic similarity for individual words is calculated using the Wu-Palmer (WUP) (?) measure provided by WordNet, which generates a score that represents how semantic similar two word senses are (the

Algorithm 3.3 – Identifies conflict between two norms.

Require: $|\sigma_1|, |\sigma_2|, |\sigma_3|$

```

1: procedure IDENTIFY_CONFLICT( $\sigma_1, \sigma_2, \sigma_3$ )
2:   if  $\sigma_1[ind_1] = \sigma_1[ind_2]$  and  $\sigma_2[ind_1] \neq \sigma_2[ind_2]$  then
3:      $sim \leftarrow \text{compute\_similarity}(\sigma_3[ind_1], \sigma_3[ind_2])$ 
4:     if  $sim > \text{threshold}$  then
5:       return True
6:     else
7:       return False

```

SIMILARITY function in Line 13). After iterating over all words, it adds the highest value to the final score (line 15). Finally, the result is the similarity score normalised by the mean length of both sentences (line 16).

The conflict identifier algorithm (Algorithm 3.3) receives as input three parameters: a pair of parties, a pair of modal verbs and a pair of norm actions (line 1)), which belongs to the two norms they want to identify the conflict. If the parties are the same, but the modal verbs are not (2), the similarity of the actions is calculated (line 3). If the similarity is higher than the defined threshold, the norm is a conflict (line 5); if it is not, the norm is not a conflict (line 7). To evaluate the algorithm, they created a dataset with conflicts manually inserted by two volunteers as the gold standard containing 121 norm conflicts out of the 11,928 norm pairs. We used our approach for the extraction of norm components and compare to the Aires *et al.* in Section 4.1.2.

3.2.2 Contract Fairness Evaluation

The concept of fairness in a contract can be understood in different ways. The argument of what is fair can be applied to the selling of a product, where the price could be considered not fair (Poppo and Zhou, 2014). Or fairness can be validated by the simple fact that the parties strictly follow what was agreed in the contract norms (Oakley, 2005). In this dissertation, we define fairness to be violated when a contract demands more of one party than another. In other words, fairness is the measure of the balance or impartiality of the contract. For example:

- Party A should pay Party B for the product.
- Party B must deliver the product to Party A within one year.

In this example we have two norms: in the first one, Party A is committed to Party B. And in the second one, Party B is the one committed to Party A. In a scenario where a contract only has these two norms, we consider that the contract is fair to both parties,

since the obligations to both parties are balanced out. This aspect of fairness is important to understand if a contract is balanced between its parties. For this end we defined in the annotation of our dataset that a norm has a *subject* and a *object* only if they are both parties of the contract. We use this annotation to indicate the direction of the obligation. The *object* is an entity that is acted upon by the *subject*. In this way, we are able to count how many times one party is obligated to the other. A mechanism that automatically does this processing and makes it viewable would save a lot of time and effort towards understanding the parties commitments to each other. Since we can identify the direction of obligations within each norm in a contract, we can use these directions to generate a graph indicating how balanced an overall contract is, and visualize this. As an example of visualization, Figure 3.3 shows an output directly from our algorithm of counting subjects and objects of a norm. In this example, we see that Party 2 has more obligations to Party 1.

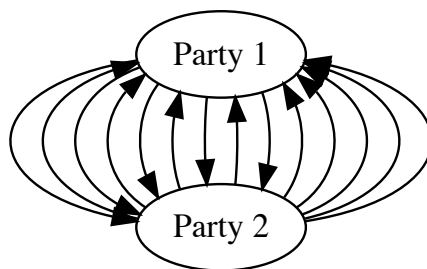


Figure 3.3 – Example of a graph for fairness assessment.

For the example in Figure 3.3 we used Contract 1 from our dataset with the following norms:

- Party 1 (JMI):
 - JMI will ship Material to Celgene in accordance with Articles 6 and 11
 - JMI will not provide Celgene with the results of all assays required to be run under the Specifications.
 - JMI supplied the Pilot Phase Material to Celgene for evaluation and regulatory filing purposes only and Celgene covenants that such Pilot Phase Material shall not be used for human consumption.
 - JMI shall file a Drug Master File and provide Celgene with a reference letter for Material upon completion of the Validation Phase Material and three (3) months of stability testing.
 - JMI shall give Celgene such information and assistance as may be reasonably necessary in the conduct of such defense.

- Party 2 (Celgene):

- Celgene shall provide JMI with a firm order for the next three (3) months and a revised nine (9) month forecast
- During the term of this Agreement, Celgene agrees to buy from JMI Material in quantities that shall equal at least fifty percent (50%) of Celgene's requirements for all formulations of Material from any bulk manufacturer thereof in each calendar year, including NPC;
- Celgene will provide all Resolving Agent needed by JMI to manufacture the Pilot Phase Material, Validation Phase Material and Material at no cost to JMI.
- Celgene shall ensure that all Resolving Agent delivered to JMI shall include instructions on proper handling requirements, including a Material Safety Data Sheet, and shall be packaged, labeled and transported in accordance with all applicable rules, regulations, tariffs, ordinances and statutes.
- Celgene assumes and will bear the expenses of, and will hold JMI harmless against, any loss, suit, claim or damage arising from or out of any intellectual property liability for Material manufactured to Celgene's Specifications
- Celgene shall provide a firm written order for Material approximately three (3) months prior to the date of the anticipated FDA approval of Celgene's NDA and shall also provide to JMI a written forecast of its requirements for Material for the next succeeding nine (9) months.
- Celgene agrees to promptly notify JMI in writing of all claims and threatened claims against Celgene for which Celgene may be entitled to indemnity hereunder.
- Celgene shall to buy from JMI all of Celgene's requirements for all formulations of Material.
- Celgene shall pay to JMI, upon the delivery by JMI to Celgene of the first lot of Pilot Phase Material pursuant to Section 3.3.

Although we describe here a possibility at a high level, we believe that the processing of the balance of a contract is a possible application of our work. A mechanism with the purpose of fairness validation could influence the design of a contract (Fehr et al., 2007) and clearly state the parties obligations to each other.

4. EXPERIMENTS

We describe in this chapter the experiments we made using our approach to extract contract components. We divided Section 4.1 in three parts: we start with the definition of our dataset in Section 4.1.1 and follow with the result we found using the dataset as the gold standard in Section 4.1.2. In 4.1.3 we evaluate our approach comparing it with Aires et al. (2017a) results.

4.1 Experiments

Our approach to extract contract components resulted in a JSON file with the components annotated (see the example attachment file B). To generate the output file, we ran experiments using SyntaxNet in a single core of a 24 core Intel Xeon CPU E5-2620 @2.00Ghz with 48GB of RAM, with a 2-minute time limit and a 2GB memory limit.

4.1.1 Dataset

Our research depended on a dataset with a clear annotation of contracts components to be the gold standard of our results. For this end, we searched for other works that already had a set of contracts annotated, but we could not find one that worked for us. Thus, we had to create a new dataset which we manually annotated for to train the probabilistic parser used to generate the syntax and dependency trees.

For the examples and for the development of this research, we used the contracts from the OneCle¹ repository. This repository contains a set of Business Contracts and it is been used for other researches, such as Gao (Gao et al., 2012; Gao and Singh, 2014) and Aires(201, 2015; Aires et al., 2017a). The work of Aires(201, 2015) already created a gold standard for classification norms (Aires et al., 2017b). Their gold standard contains 92 annotated contracts, divided into two sets: a norm sentence set with 9864 norms and a common sentence (non-norm) set with 10554 sentences. In our research we chose to increase Aires's annotation, using the same contracts he used for his work.

Our annotation was made using Webanno(Yimam et al., 2013), a tool that enables annotations of text files with a friendly interface. Our annotation was made in collaboration with Pradeep Murukannaiah². After analyzing the related works and the ends we wanted to achieve, we decided to annotate the following contract's components:

¹<http://www.onecle.com>

²<http://www.se.rit.edu/~pkm/>

- Title
- Parties
- Parties alias
- Norm
 - Subject
 - Object
 - Condition
 - Deontic modality

The annotation of our dataset took a long time to be finished since we needed very specific components annotated. For this reason, we were able to annotate 15 contracts. Within these contracts, we have 1217 norms and 3855 sentences.

4.1.2 Component extraction

Using SyntaxNet we are able to generate the parse tree for each sentence of the contract. The tree provides us with the information necessary to extract the components, such as the classification of modal verbs, which indicates the deontic modality of a norm, or the nominal subject, which indicates the action of a norm. For classification purposes, we decide to identify the object and subject exclusively as parties. Figure 4.1 shows an example of parse tree output from the sentence *JMI will ship Material to Celgene in accordance with Articles 6 and 11*. We highlighted in green the classification for *nominal subject*, which we classify as the action of the norm. Since *Celgene* is inside this part of the tree, it is classified as the subject and, therefor, *JMI* is the object.

The output of this sentence is:

- **Action:** ship Material to Celgene in accordance with Articles 6 and 11.
- **Deontic modality:** obligation
- **Modal verb:** will
- **Party 1:** JMI
- **Party 2:** Celgene
- **Object:** JMI
- **Subject:** Celgene

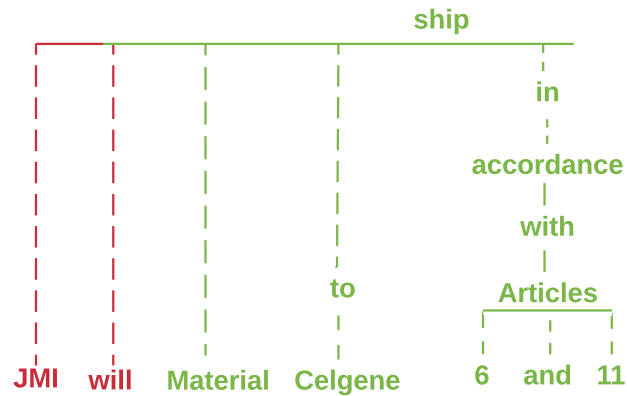


Figure 4.1 – Example of parse tree.

Although the norm above is correctly classified, we faced many issues with longer sentences. As an example, Figure 4.2 shows the parse tree of the sentence *Any present or future duty use whether Federal applicable to this transaction are not included in the price herein stated and when due shall be paid by Celgene without cost or charge to JMI.* . We highlighted in green the part of the tree classified as the *nominal subject*, which contains more *nominal subjects* inside of it. The algorithm does not know which is the right one and captures the one in the higher level of the tree. For this reason, in this example, our algorithm is not able to categorize correctly the action of the norm and the object and subject are miss placed. The output is:

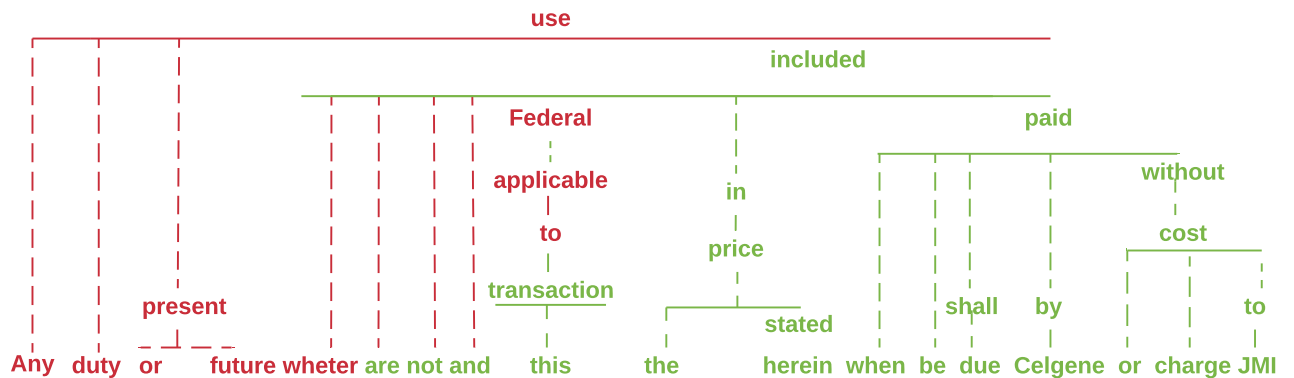


Figure 4.2 – Second example of parse tree.

- **Action:** transaction are not included in the price herein stated and when due shall be paid by Celgene without cost or charge to JMI.
- **Deontic modality:** permission
- **Modal verb:** shall
- **Party 1:** JMI

- **Party 2:** Celgene
- **Object:** JMI
- **Subject:** Celgene

In order to evaluate our results, we used four measures: accuracy, precision and recall, F-score(Powers, 2011). To evaluate the norm classifier we consider as true positives the norms that are both in the annotated contract and in the output of our algorithm; true negatives are the sentences that are classified as not norms for both; false positives are the sentences classified as norms for our algorithm, but as not norms for the annotation; and false negatives are the sentences classified as not norms for the algorithm but classified as norms for the annotation.

Table 4.1 shows the results for the norm classification. We used the *Sequence-Matcher*³ module to measure the semantic similarity between strings. We used a simple threshold rule to compare the strings from the norm candidates from each contract because the comparison of the actual string was returning too many false negatives, which are sentences wrongly classified as not norms. We highlighted the best result for each threshold. We can see that the range of 0.5 has the best results for the true norms and for the mistaken classification (false positives and false negatives) as well. *Contract 9* shows a specific behavior due to the structure of most of the norms: they are written in long sentences which makes it confusing for our algorithm.

Threshold	0.5	.07	0.8	0.5	0.7	0.8	0.5	0.7	0.8	0.5	0.7	0.8
Contract	True Positives			True Negatives			False Positives			False Negatives		
1	39	32	26	56	59	62	44	52	59	8	15	21
2	50	37	34	31	45	49	53	72	77	24	37	40
3	50	40	36	81	83	85	53	67	74	16	26	30
4	74	60	52	142	153	156	87	107	127	39	53	61
5	98	75	65	193	238	244	153	185	190	65	90	98
6	54	39	30	152	159	162	63	79	89	18	33	42
7	67	53	45	143	174	186	63	83	94	16	30	38
8	65	52	41	122	140	151	88	106	116	26	39	50
9	9	0	0	171	181	184	147	163	164	82	91	91
10	63	47	42	50	54	55	127	142	150	5	21	26
11	32	27	21	42	51	54	42	49	52	19	24	30
12	72	59	48	137	150	152	89	107	117	18	31	42
13	57	45	40	80	83	84	53	69	78	15	27	32
14	28	22	20	72	79	79	30	42	45	18	24	26
15	75	59	47	132	137	137	54	73	83	15	31	43

Table 4.1 – Comparison of norm classification with range of threshold values.

³<https://docs.python.org/2/library/difflib.html>

In Table 4.2 we show the accuracy, recall, precision and F-score of each of the contracts tested, based in the values from Table 4.1. We highlighted the best results for each threshold. Our highest result for accuracy was in *Contract 15* with 77%.

Threshold	0.5	.07	0.8	0.5	0.7	0.8	0.5	0.7	0.8	0.5	0.7	0.8
Contract	Accuracy			Recall			Precision			F-Score		
1	0.66	0.64	0.62	0.83	0.68	0.55	0.47	0.38	0.31	0.6	0.49	0.4
2	0.57	0.57	0.58	0.68	0.5	0.46	0.49	0.34	0.31	0.57	0.4	0.37
3	0.66	0.62	0.61	0.76	0.61	0.55	0.49	0.37	0.33	0.6	0.46	0.41
4	0.64	0.64	0.62	0.65	0.53	0.46	0.46	0.36	0.29	0.54	0.43	0.36
5	0.58	0.62	0.61	0.6	0.45	0.4	0.39	0.28	0.25	0.47	0.35	0.31
6	0.74	0.71	0.69	0.75	0.54	0.42	0.46	0.33	0.25	0.57	0.41	0.31
7	0.69	0.75	0.76	0.81	0.64	0.54	0.52	0.39	0.32	0.63	0.48	0.4
8	0.62	0.64	0.64	0.71	0.57	0.45	0.42	0.33	0.26	0.42	0.48	0.33
9	0.52	0.53	0.54	0.1	0	0	0.06	0	0.06	0.07	0.5	0.5
10	0.47	0.42	0.4	0.93	0.69	0.62	0.33	0.25	0.22	0.49	0.37	0.32
11	0.57	0.6	0.58	0.63	0.53	0.41	0.43	0.36	0.29	0.51	0.43	0.34
12	0.66	0.66	0.63	0.8	0.66	0.53	0.45	0.36	0.29	0.58	0.47	0.37
13	0.67	0.63	0.61	0.79	0.63	0.56	0.52	0.39	0.34	0.63	0.48	0.42
14	0.68	0.69	0.68	0.61	0.48	0.43	0.48	0.34	0.31	0.54	0.4	0.36
15	0.77	0.73	0.68	0.83	0.66	0.52	0.58	0.45	0.36	0.68	0.54	0.43

Table 4.2 – Measures threshold values comparison for norm classification.

We decided to evaluate each component extracted from the contracts individually. For this evaluation we considered only the sentences classified as norms. True positives are the components classified equally for both the annotation in the dataset and our approach; false positives are the components found by our approach and not by the annotation; true negatives are the components that neither the annotation or our results found; and false negatives are the components classified by the annotation and not by our approach. Table 4.3 shows the results from the first zone of the contract (the title zone), which contains the title and the description of the parties. We achieved a high accuracy in the title classification (86%) and in the party (80%) and party alias (86%).

Threshold	0.5	.07	0.8	0.5	0.7	0.8	0.5	0.7	0.8	0.5	0.7	0.8
Component	Accuracy			Recall			Precision			F-Score		
Title	0.86	0.86	0.86	1	1	1	0.86	0.86	0.86	0.86	0.86	0.86
Party	0.8	0.76	0.76	1	1	1	0.8	0.76	0.76	0.88	0.86	0.86
Party alias	0.86	0.83	0.83	1	1	1	0.86	0.83	0.83	0.92	0.9	0.9

Table 4.3 – Threshold values comparison for first contract zone.

Tables 4.4, 4.5 and 4.6 show the results for the deontic modality, action, condition, subject and object components, respectively. We highlighted the best results for each threshold in all tables. The deontic modality had an accuracy of 94% with 97% of precision. We had a high accuracy for the action (94%) too. The condition, in spite of the high accuracy (94%), had a low precision rate (lowest at 36%) due to its rare presence in the norms. Subject and object were the components with the lowest accuracy (52% and 74%) but with a high recall (98%) indicating that in the norms that they existed, they were correctly classified.

Threshold	0.5	.07	0.8	0.5	0.7	0.8	0.5	0.7	0.8	0.5	0.7	0.8
Component	Accuracy			Recall			Precision			F-Score		
Deontic modality	0.83	0.90	0.94	0.97	0.97	0.96	0.85	0.92	0.97	0.90	0.94	0.96

Table 4.4 – Threshold values comparison for deontic modality classification.

Threshold	0.5	.07	0.8	0.5	0.7	0.8	0.5	0.7	0.8	0.5	0.7	0.8
Component	Accuracy			Recall			Precision			F-Score		
Action	0.87	0.90	0.94	0.94	0.94	0.95	0.91	0.96	0.98	0.92	0.94	0.97
Condition	0.91	0.92	0.94	0.03	0.03	0.10	0.36	0.4	0.42	0.08	0.08	0.15

Table 4.5 – Threshold values comparison for action and condition classification.

Threshold	0.5	.07	0.8	0.5	0.7	0.8	0.5	0.7	0.8	0.5	0.7	0.8
Component	Accuracy			Recall			Precision			F-Score		
Subject	0.49	0.50	0.52	0.98	0.98	0.98	0.50	0.51	0.52	0.65	0.65	0.67
Object	0.68	0.70	0.74	0.98	0.98	0.98	0.69	0.71	0.75	0.80	0.81	0.84

Table 4.6 – Threshold values comparison for object and subject classification.

The analysis of these results shows that the major problem were the longer sentences, specifically, norms with itemized components. Norms composed by itemized points create two issues: first, the context is declared in the beginning of norm and repeated using alias in the rest of the norm; and second, the longer sentences. These two situations are confusing for the classifier and have affected our results. To validate that, we manually removed some of these cases from Contract 1 and run our algorithm again. The results are showed in Table 4.7.

With the removal of the longer sentences, we had an improvement in the norm classification of Contract 1 and overall, in the other components as well.

Threshold	0.5	.07	0.8	0.5	0.7	0.8	0.5	0.7	0.8	0.5	0.7	0.8
Component	Accuracy			Recall			Precision			F-Score		
Norms	0.87	0.73	0.67	1	0.91	0.82	0.86	0.77	0.75	0.92	0.83	0.78
Deontic modality	0.67	0.83	1	1	1	1	0.67	0.83	1	0.80	0.91	1
Action	0.38	0.83	0.52	0.33	1	0.98	0.67	0.80	0.52	0.44	0.89	0.67
Condition	0.91	0.89	0.86	0.1	1	0	0.5	0.5	0.5	0.16	0.67	0
Subject	0.58	0.89	0.78	1	1	1	0.58	0.83	0.67	0.91	0.61	0.8
Object	0.50	0.78	0.57	1	1	1	0.50	0.71	0.57	0.67	0.83	0.73

Table 4.7 – Threshold values comparison for smaller sentences classification.

4.1.3 Conflict identification by action comparison

We explain in this section the experiments we did using our approach to extract the parties, modal verb and action of a norm and compared them to the approach proposed by Aires *et al.* (Aires et al., 2017a). The evaluation is made using a dataset as the gold standard containing 121 norm conflicts out of the 11,928 norm pairs.

Table 4.8 shows the results of the evaluation for each threshold value applied in the similarity algorithm (explained in Section 3.2.1). True positives indicate norm pairs identified by Aires *et al.* approach that are part of the 121 norm conflicts. True negatives indicate norm pairs that have no conflict and were not identified as conflicting. False positives are norm pairs identified as potential conflicts but have no conflict, and false negatives are the conflicting norm pairs that were not identified as such. The evaluation used four more measures: accuracy, precision, recall and f-score.

Threshold	40%	50%	60%	70%	80%
True Positives	83	81	80	47	31
True Negatives	11618	11775	11786	11797	11799
False Positives	189	32	21	10	8
False Negatives	38	40	41	74	90
Accuracy	0.98	0.99	0.99	0.99	0.99
Precision	0.31	0.72	0.79	0.82	0.79
Recall	0.69	0.67	0.66	0.39	0.26
F-Score	0.42	0.69	0.72	0.53	0.39
Specificity	0.98	1.00	1.00	1.00	1.00
Total number of conflicts	121				

Table 4.8 – Threshold values comparison from Aires et al. (2017a).

Our work produces a different input for the conflict identification algorithm. We use SyntaxNet to parse each norm of a contract into its syntactic model and extract the information we want from it. This model is a dependency tree, which is a representation of a syntactic structure of a string. As required by the algorithm, we extract the party, the modal verb and the object from a given sentence (which is a norm provided by the Aires *et al.* work).

For the subject we extract the “nsubj”, the nominal subject, which is the syntactic subject of a clause. In order to find conflicts, we extract the “nsubj” and all its complements. When SyntaxNet does not find any “nsubj”, we assume the same approach as Aires *et al.*, considering subject everything in the sentence that comes after the modal verb.

As an example, the norm “Pilots may have a right to join, notjoin⁴, maintain, or drop their membership in the Union as they see fit”. The dependency tree for this sentence is shown in Figure 4.3. We highlighted in red and in green the two main subjects SyntaxNet found. In red, we show the main modal verb (may), the plural noun (the party Pilots) and the root node (have). In green we show the entire structure that represents the norm action.

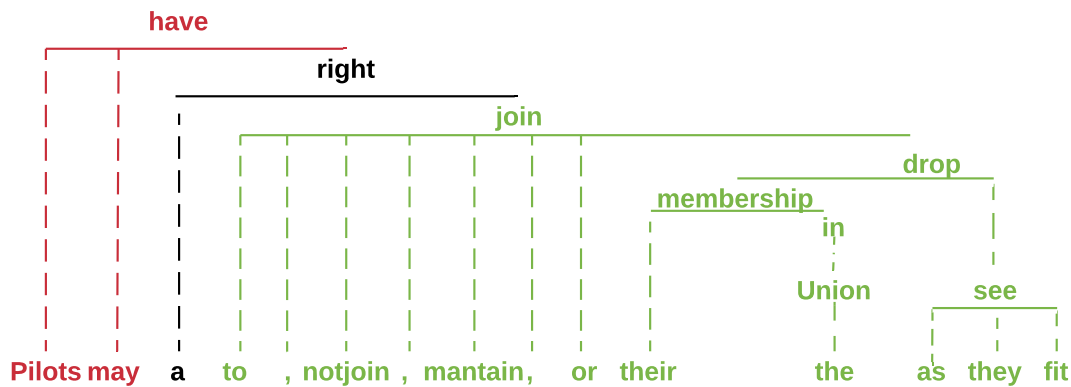


Figure 4.3 – Example of a dependency tree.

Our algorithm extracts the following:

- Parties: Union
- Modal verb: may
- Object: to join notjoin⁵ maintain or drop their membership in the Union as they see fit .

In this example, it is possible to see one of the problems with parsing text. Our algorithm takes *Union* as the party, because it is classified as NNP (proper noun) instead of *Pilots*, which is classified as plural noun. We use this approach to find the party because SyntaxNet does not provide a way to find Named Entities in the parser. We acknowledge that using a specific algorithm for that would improve the results.

Table 4.9 shows the results with our input. We highlight the best results for each of the measure. We use the exactly same dataset as Aires *et al.*, with the same measures to validate the results. Unfortunately, our inputs seem to have decreased the values in all thresholds. We believe this happened due to two main factors: first, the absence of an entity extractor instead of only extracting the NNPs; and second, the size of the text found as norm subject might have confused the algorithm that measures the semantic similarity. As

⁴We used the word as it was written in the contract.

⁵We used the word as it was written in the contract.

mentioned before, in their work they use everything that comes after the modal verb. With our approach, we extract a minor piece of text, which could explain the lower result.

Threshold	40%	50%	60%	70%	80%
True Positives	27	23	23	20	19
True Negatives	3923	4184	4189	4211	4240
False Positives	458	197	192	170	141
False Negatives	3923	98	98	101	4240
Accuracy	0.88	0.93	0.94	0.94	0.95
Precision	0.06	0.10	0.11	0.11	0.12
Recall	0.22	0.19	0.19	0.39	0.16
F-Score	0.09	0.13	0.14	0.13	0.14
Specificity	0.90	0.96	0.96	0.96	0.97
Total number of conflicts	121				

Table 4.9 – Threshold values comparison with SyntaxNet output.

5. RELATED WORK

This section describes related work on extraction and representation of contract components. Each approach deals with extracting contract content before processing any kind of data. We show here these different approaches of processing a contract as the base of our own work, which relates it with the main idea of this research: formalize a contract components in a way that it is useful for other algorithms and approaches. We will give labels to the works in order to facilitate a further analysis.

Governatori (Governatori, 2005) (Work 1) introduces an approach to transform business contract rules from natural language into machine readable and executable form. His interest is in to allow explicit monitoring of rules in order to find any case of violation in it. For that, he tries to represent a contract originally from natural language as a set of RuleML rules. RuleML is a description language, based on XML, used for the representation of rules, with hierarchical structure. It provides logical representation in a language that is easier to be suitable to a computer. His process is manual, and he transforms a contract into a logical form using Deontic and Defeasible Logic in the clauses of the contract, by transforming it into facts, definitions and normative rules. He implements the contract rules within RuleML, which is constructed by tags in a similar way as it happens with an HTML file, containing a body with a content and head with a title. In this way, it is possible to obtain the logical representation of the contract through RuleML and from that the contract components can be extracted and implemented into a machine executable syntax.

Gao *et al.* (Gao et al., 2012) (Work 2) proposes an approach to extract service exceptions from contracts. Service exceptions are conditions in the clauses that list what should happen when something out of the general rule of the agreement occurs. For example, in the sentence “In the case of any defect in the service, the company shall issue a credit”, the exception is on “any defect in the service”. Their technique is based on an unsupervised learning algorithm that uses linguistic patterns such as “In (the) case of NP”, where NP is a noun phrase; or “in the case that”, which selects sentential clauses. Their approach results in a tool: *Enlil*, a tool that highlights exception phrases. In summary, they preprocess the contract text, which is to strip HTML tags and all kind of noise from the text and segment it into sentences. From this sentences they extract the exceptions based on patterns. They identify noun phrases corresponding to exceptions by using patterns. To evaluate the system they create a gold standard, manually annotating exceptions from five contracts. Using the conjunction rule, *Enlil* extracts 29 phrases with three false positives and three false negatives. They reach 0,92% of precision. When analyzing their results, they realized that their parser could not process long sentences, due to the complexity of parsing correctly grammatical dependencies.

Gao and Singh (Gao and Singh, 2014) (Work 3) introduce an approach of textual patterns and Machine Learning with the objective of extracting norms from business contracts and assign a type to each norm. In their work they define six types of norms, which are: commitment (dialectical or practical), authorization, power, prohibition, and sanction. The extraction of norms begins by identifying norm candidates. For that, they separate the sentences that contain *signal words*, which are modal verbs such as “can” and “must”, and verbs such as “warrant” and “agree”. Every sentence with a *signal word* is considered a norm candidate. Next, they train a classifier with *features* they consider relevant for the classification of the norm type (e.g., the main verb). The classifier uses Naïve Bayes, Support Vector Machine and logistic regression algorithms. They also used four heuristics to extract elements of a norm (subject, object, antecedent, and consequent). To evaluate the work, they created a golden standard using 1000 sentences from Onecle¹ contract repository.

Chalkidis and Androutsopoulos have done research on how to automatically extract elements from contracts. In their first approach (Chalkidis et al., 2017) (Work 4) they define 11 types of contract elements to work with, they are: Contract Title, Contracting Parties, Start Date, Effective Date, Termination Date, Contract Period, Contract Value, Governing Law, Jurisdiction, Legislation Refs and Clause Headings. To extract the elements, they separate a contract into “zones of extractions”, and use machine learning algorithms, such as linear classifiers (Logistic Regression, linear SVM) with hand-crafted features, word and POS tag embedding to execute the extraction of elements. From this work they constructed a labeled dataset of approximately 3,500 English contracts with gold contract element annotations and a larger unlabeled dataset of approximately 750,000 English contracts. As a follow-up of their work, they experimented with the data from this dataset using deep learning techniques (?) (Work 5). They used the same approach for extracting elements in zones, and they search for the same 11 types of elements, this time training a bidirectional LSTM with a logistic regression layer (BiLSTM-LR), operating on pre-trained word, POS tag, and token-shape embedding. This last experiment outperforms in most cases the best methods of their previous work.

Aires *et al.* (Aires et al., 2017a) (Work 6) has the objective to find conflicts in norms written in a contract. Their approach has two main tasks: first, pre-process a contract to extract its norms. For that, they train a machine learning algorithm capable to distinguish norms from common sentences. They train the algorithm with a manually labeled dataset created specifically for this work, which contains a set of sentences classified as norms and non-norms. Second, they identify the modality in each norm and search for possible conflicts among the ones in the same contract. To find a possible conflict they identify three components inside a norm: party, deontic meaning, and norm action. We show in the next section our experiment with this work and the results we had when changing the input for algorithm of conflict extractor created by Aires *et al.*

¹<http://contracts.onecle.com>

We decide to divide the process of development of the work to extract information from contracts in three parts, in order to facilitate the categorization of each one of the parts. They are:

- Input: the starting point of the work.
- Processing: the main processing of the work.
- Output: the result of the work.

In Table 5.1 we summarize the features of the work described here with the one we aim to develop in ours. To facilitate the analysis of this research we introduce two concepts: the first is related to the process of development of the works and the second is related to the information extracted from norms.

RELATED WORK				
Work	Extracted nents	compo-	Automated work	Manual work
1	None		None	Processing: describe the contract
2	Sentences		<i>Input:</i> remove unwanted tags from contract <i>cor-</i> <i>pus.</i> <i>Processing:</i> ex- tract sentences (pattern based)	None
3	Norms		<i>Input:</i> remove unwanted tags from contract <i>cor-</i> <i>pus.</i> <i>Processing:</i> pat- tern based	None
4	Contract title, parties, contract period, jurisdic- tion, governing law, leg- islation references, con- tract value clause		<i>Processing:</i> machine learning algorithms to extract the elements	<i>Input:</i> manually anno- tated English contracts
5	Contract title, parties, contract period, jurisdic- tion, governing law, leg- islation references, con- tract value clause		<i>Processing:</i> different neural networks used to extract elements	<i>Input:</i> manually anno- tated English contracts
6	Parties, norms, norm type		<i>Processing:</i> classify the norm type	<i>Input:</i> manually an- notated contracts with conflicts and norm/not norms
Our work	Title, party, modal verb, object, subject, action, condition		Processing: compo- nents extraction	None

Table 5.1 – Comparisons among related work.

We introduce here a level of granularity for the norm information extraction. This granularity is related specifically to the extraction and representation of norms, and shows which elements are extracted.

1. identify norms and not norms
2. identity parties inside the norms
3. identify subject and object

4. identify condition
5. identify exception
6. identify CTD
7. identify modality
8. identify norm references

Table 5.2 shows the granularity found in each of the related works, following the description we showed above.

Work	1	2	3	4	5	6	7	8
1	x						x	
2					x			
3	x		x					
4		x						
5		x						
6	x	x					x	
Our work	x	x	x	x			x	

Table 5.2 – Granularity table.

We described in this section works that extract and classify components of contracts with different approaches. We could not find one that automatically extracts and defines these components using only a contract as an input or even that classifies all the components that our approach does.

6. CONCLUSION

The large amount of data produced every day increases the amount of research on Natural Language Processing. This data often needs preprocessing before it can be transformed into information amenable to computer processing, which leads to the need to automate such transformation as much as possible. This is the case of processing contracts, where we have a large amount of text that needs to be structured in a suitable way for further analyses.

In this dissertation we define a document structure and information extraction mechanism for preprocessing contracts and automatically extract and classify the components of a contract. To do, so we analyze a contract structure, and processed its sentence using dependencies trees.

During our research, we were able to process the contracts and generate an output of its components with a good accuracy when dealing with well formed sentences. The problems we faced remained in the longer sentences, with context being kept for several lines of text and confusing the classifier. Although we acknowledge this issue, we reiterate that using our approach, we can deal with more complex sentence structures, which is not possible when using fixed rules. Our key contribution is the definition of a document structure and a mechanism to parse documents that lends itself to multiple contract processing tasks, which we exemplified with two tasks: norm conflict identification and contract fairness estimation.

As an additional contribution of our work, we created a dataset of 15 contracts annotated with: title, party, party alias, norm, norm action, norm condition, subject and object. This dataset is available for download¹ and, if possible, to be increased and tested for other researches. Our dataset is useful for training machine learn algorithms to contract validation since we have more than 1200 annotated norms with its components.

For future work, we should refine our extraction of content from the parser, in order to improve the extraction of norm actions and conditions. To use an algorithm specific for Named Entity extraction is also something we plan to implement, so we don't rely only on proper nouns to find the parties. Also, we need to use more algorithms of finding semantic similarity to see if there is something else to be improved in the comparison of strings.

¹Dataset available at https://github.com/DaniBauer/contract_dataset

REFERENCES

- (2015). Identifying potential conflicts between norms in contracts. Master's thesis. Faculdade de Informática.
- Aires, J. P., Pinheiro, D., de Lima, V. S., and Meneguzzi, F. (2017a). Norm conflict identification in contracts. *Artificial Intelligence and Law*, 25(4):397–428.
- Aires, J. P., Pinheiro, D., and Meneguzzi, F. (2017b). Norm Dataset: Dataset with Norms and Norm Conflicts.
- Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., and Collins, M. (2016). Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*.
- Axelrod, R. (1986). An evolutionary approach to norms. *American political science review*, 80(4):1095–1111.
- Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.
- Bessone, D. (1987). *Do contrato: teoria geral*. Forense.
- Bird, S. (2006). Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics.
- Cambria, E., Poria, S., Gelbukh, A., and Thelwall, M. (2017). Sentiment analysis is a big suitcase. *IEEE Intell Syst*, 32(6):2–8.
- Cannon, J. P., Achrol, R. S., and Gundlach, G. T. (2000). Contracts, norms, and plural form governance. *Journal of the Academy of Marketing Science*, 28(2):180–194.
- Chalkidis, I., Androutsopoulos, I., and Michos, A. (2017). Extracting contract elements. In *Proc. of the 16th Int. Conf. on Artificial Intelligence and Law*, pages 19–28.
- Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, 37(1):51–89.
- Covington, M. A. (2001). A fundamental algorithm for dependency parsing. In *Proceedings of the 39th annual ACM southeast conference*, pages 95–102. Citeseer.
- Curtotti, M. and McCreath, E. (2010). Corpus based classification of text in australian contracts. In *Proceedings of the Australasian Language Technology Association Workshop, Melbourne, Australia*, pages 18–26.

- Curtotti, M., McCreath, E., and Sridharan, S. (2013). Software tools for the visualization of definition networks in legal contracts. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law, ICAIL '13*, pages 192–196, New York, NY, USA. ACM.
- De Marneffe, M.-C., MacCartney, B., Manning, C. D., et al. (2006). Generating typed dependency parses from phrase structure parses. In *Lrec*, volume 6, pages 449–454.
- Dignum, F. (1999). Autonomous agents with norms. *Artificial Intelligence and Law*, 7(1):69–79.
- Dignum, F. (2002). Abstract norms and electronic institutions. In *Proceedings of International Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA'02)*, pages 93–104.
- Dixson, R. J. (2002). Analyzing english grammar-an introduction to feature theory.
- Fehr, E. and Fischbacher, U. (2004). Social norms and human cooperation. *Trends in cognitive sciences*, 8(4):185–190.
- Fehr, E., Klein, A., and Schmidt, K. M. (2007). Fairness and contract design. *Econometrica*, 75(1):121–154.
- Fellbaum, C. (1998). *WordNet*. Wiley Online Library.
- Fine, T. L. (2006). *Feedforward neural network methodology*. Springer Science & Business Media.
- Føllesdal, D. and Hilpinen, R. (1970). Deontic logic: An introduction. In *Deontic logic: Introductory and systematic readings*, pages 1–35. Springer.
- Frakes, W. B. and Baeza-Yates, R. (1992). Information retrieval: data structures and algorithms.
- Gao, X. and Singh, M. P. (2014). Extracting normative relationships from business contracts. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 101–108. International Foundation for Autonomous Agents and Multiagent Systems.
- Gao, X., Singh, M. P., and Mehra, P. (2012). Mining business contracts for service exceptions. *IEEE Transactions on Services Computing*, 5(3):333–344.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.
- Governatori, G. (2005). Representing business contracts in ruleml. *International Journal of Cooperative Information Systems*, 14(02n03):181–216.

- Griffo, C., Almeida, J. P. A., Guizzardi, G., and Nardi, J. C. (2017). From an ontology of service contracts to contract modeling in enterprise architecture. In *21st IEEE Enterprise Computing Conference EDOC*.
- Gupta, N. (2013). Artificial neural network. *Network and Complex Systems*, 3(1):24–28.
- Han, B., Cook, P., and Baldwin, T. (2013). Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):5.
- Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.
- Hyvärinen, A., Gutmann, M., and Entner, D. (2010). Unsupervised machine learning.
- Ikonomakis, M., Kotsiantis, S., and Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS transactions on computers*, 4(8):966–974.
- Ingason, A. K., Helgadóttir, S., Loftsson, H., and Rögnvaldsson, E. (2008). A mixed method lemmatization algorithm using a hierarchy of linguistic identities (holi). In *Advances in Natural Language Processing*, pages 205–216. Springer.
- Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3):31–44.
- Jurafsky, D. and Martin, J. H. (2014). *Speech and language processing*, volume 3. Pearson London.
- Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.
- Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.
- Karlapalem, K., Dani, A. R., and Krishna, P. R. (2001). A frame work for modeling electronic contracts. In *International Conference on Conceptual Modeling*, pages 193–207. Springer.
- Kim, M.-Y., Kang, S.-J., and Lee, J.-H. (2001). Resolving ambiguity in inter-chunk dependency parsing. In *NLPRS*, pages 263–270.
- Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.
- Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques.

- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*.
- Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.
- Matthews, P. H. (2007). *Syntactic relations: A critical survey*, volume 114. Cambridge University Press.
- Meneguzzi, F. and Luck, M. (2009). Norm-based behaviour modification in BDI agents. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*, pages 177–184.
- Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Minelli, M., Chambers, M., and Dhiraj, A. (2012). *Big data, big analytics: emerging business intelligence and analytic trends for today's businesses*. John Wiley & Sons.
- Modgil, S., Faci, N., Meneguzzi, F., Oren, N., Miles, S., and Luck, M. (2009). A framework for monitoring agent-based normative systems. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*, pages 153–160.
- Nadh, K. and Huyck, C. R. (2009). Prepositional phrase attachment ambiguity resolution using semantic hierarchies.
- Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2):103–134.
- Nivre, J. (2005). Dependency grammar and dependency parsing. *MSI report*, 5133(1959):1–32.
- Oakley, R. L. (2005). Fairness in electronic contracting: minimum standards for non-negotiated contracts. *Hous. L. Rev.*, 42:1041.
- Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

- Pantel, P. and Lin, D. (2000). An unsupervised approach to prepositional phrase attachment using contextually similar words. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 101–108. Association for Computational Linguistics.
- Petrov, S. (2016). Announcing syntaxnet: The world's most accurate parser goes open source. *Google Research Blog*, 12.
- Poppo, L. and Zhou, K. Z. (2014). Managing contracts for fairness in buyer–supplier exchanges. *Strategic Management Journal*, 35(10):1508–1527.
- Powers, D. M. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation.
- Prisacariu, C. and Schneider, G. (2007). A formal language for electronic contracts. In *FMOODS*, volume 7, pages 174–189. Springer.
- Rabiner, L. R. and Juang, B.-H. (1993). Fundamentals of speech recognition.
- Reinhart, T. (2016). *Anaphora and semantic interpretation*. Routledge.
- Ringgaard, M., Gupta, R., and Pereira, F. C. (2017). Sling: A framework for frame semantic parsing. *arXiv preprint arXiv:1710.07032*.
- Russell, S., Norvig, P., and Intelligence, A. (1995). A modern approach. *Artificial Intelligence*. Prentice-Hall, Englewood Cliffs, 25:27.
- Sak, H., Senior, A., and Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.
- Socher, R., Bauer, J., Manning, C. D., et al. (2013). Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 455–465.
- Somers, H. (1999). Example-based machine translation. *Machine Translation*, 14(2):113–157.
- Sunstein, C. R. (1996). Social norms and social roles. *Columbia law review*, 96(4):903–968.
- Tan, A.-H. et al. (1999). Text mining: The state of the art and the challenges. In *Proceedings of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases*, volume 8, pages 65–70.
- Von Wright, G. H. (1951). Deontic logic. *Mind*, 60(237):1–15.

Yimam, S. M., Gurevych, I., Eckart de Castilho, R., and Biemann, C. (2013). Webanno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6, Sofia, Bulgaria. Association for Computational Linguistics.

ATTACHMENT A – Contract example

FORMATION AGREEMENT

Among

DREAMWORKS ANIMATION SKG, INC.,

DREAMWORKS L.L.C.,

[HOLDCO] LLLP

and

THE STOCKHOLDERS AND OTHER PERSONS PARTY HERETO

Dated As Of October [], 2004

TABLE OF CONTENTS

ARTICLE I

Definitions

Section 1.01. Certain Defined Terms	1
Section 1.02. Other Definitional Provisions	6

FORMATION AGREEMENT, dated as of October [], 2004,
among DREAMWORKS ANIMATION SKG, INC., a Delaware corporation
(the "Company"), DREAMWORKS L.L.C., a Delaware limited
liability company ("DW"), [HOLDCO] LLLP, a Delaware limited
liability limited partnership ("Holdco"), and the
stockholders

and other persons party hereto.

WHEREAS, DW, the Company and DreamWorks Animation L.L.C., a Delaware limited liability company ("DWA LLC"), have entered into a Separation Agreement dated as of the date hereof, providing for the separation of the animation business from DW;

WHEREAS, on the Separation Date (as defined below) immediately prior to effectiveness of the Underwriting Agreement (as defined below), DW made a distribution-in-kind to its members (in accordance with Article VIII of the Sixth Amended and Restated Limited Liability Company Agreement of DW) of its interest in DWA LLC;

WHEREAS, the distributed DWA LLC interests will be contributed to the Company in exchange for Common Stock (as defined below);

WHEREAS, each Contributing Member (as defined below) desires to form Holdco and to contribute any shares of such Common Stock received from the Company, other than the IPO Sale Shares (as defined below) and other than as set forth in Section 2.04, to Holdco in exchange for partnership interests in Holdco;

WHEREAS, the Contributing Members desire to provide for the sale, in a follow-on secondary offering, of all or a portion of the shares of Common Stock held directly by the Contributing Members and the shares of Common Stock contributed to Holdco by the Contributing Members; and

WHEREAS, the Company, Holdco and certain other parties hereto have entered into a Registration Rights Agreement, dated as of the date hereof (the "Registration Rights Agreement"), that, among other things, provides for certain procedures with respect to the Follow-on Offering and the Universal Triggered Offering (each as defined below);

NOW, THEREFORE, in consideration of the foregoing and the respective covenants and agreements set forth herein, and intending to be legally bound hereby, the parties hereto agree as follows:

ARTICLE I

Definitions

Section 1.01. Certain Defined Terms. As used in this Agreement:

"Agreement" means this Formation Agreement, as it may be amended, supplemented, restated or modified from time to time.

<PAGE>

2

"Amended LLC Agreement" means the Seventh Amended and Restated Limited Liability Company Agreement of DW, dated as of October [], 2004, as it may be amended, supplemented, restated or modified from time to time.

"Asserted Liability" has the meaning assigned to such term in Section 6.05(d).

"Business Day" means any day that is not a Saturday, a Sunday or other day on which banks are required or authorized by law to be closed in The City of New York.

"Charter" means the Restated Certificate of Incorporation of the Company, as amended or restated from time to time.

"Claims" has the meaning assigned to such term in Section 6.05(a).

"Claims Notice" has the meaning assigned to such term in Section 6.05(d).

ARTICLE II

Distribution and Contribution; Holdco Transactions

Section 2.01. Contributions and Redemptions of Preferred Interests; Distribution of DWA LLC Interests; Execution of Amended LLC Agreement. (a) On the Separation Date, after consummation of the transactions contemplated in Section 2.01 of the Separation Agreement, (x) Thomson shall contribute 33-1/3% of the Class T/T Interests to the Company in exchange for the number of shares of Common Stock set forth on Schedule 2.02 and (y) Universal shall contribute 50% of the Class U Interests to the Company in exchange for the number of shares of Common Stock set forth on Schedule 2.02 (the "Preferred Contributions"). For

the avoidance of doubt, the number of shares of Common Stock received in exchange for the Preferred Contributions shall be equal to (i) in the case of Universal, \$75 million divided by the

<PAGE>

7

IPO Price and (b) in the case of Thomson, \$50 million divided by the IPO Price. Immediately after consummation of the Preferred Contributions, DW shall redeem such Class T/T Interests and such Class U Interests from the Company in exchange for (i) all of DW's 100% interest in the capital stock of DreamWorks Inc. and (ii) the number of DWA LLC Interests set forth in Schedule 2.01(a) (the "Preferred Redemptions"). DW acknowledges that it will treat the Preferred Redemptions as a liquidating distribution with respect to the Class T/T Interests and Class U Interests so redeemed and shall report the Preferred Redemptions as such under Section 732(b) of the Internal Revenue Code.

(b) On the Separation Date, immediately after consummation of the DW Distribution, each Member (other than Universal and Thomson) shall execute and deliver a pledge agreement in favor of the lenders under the Revolving Credit Facility, which pledge agreements shall provide for the pledge of Common Stock having an aggregate value of \$300 million (valued at the IPO Price), allocated among such Members in an amount equal to their participation percentages in DW (as of the date hereof) as set forth on Schedule 2.01(b) multiplied by \$300 million (which amount shall be subject to adjustment in the case of Contributing Members based upon the Final Allocation of such pledged shares of Common Stock).

Section 2.02. Contribution of the DWA LLC Interests to the Company; Issuance of Common Stock by the Company. On the Separation Date, after consummation of the DW Distribution and following effectiveness of the Underwriting Agreement, each Member (or DWI II, in the case of DW Investment Inc.) shall contribute all its right, title and interest in and to the DWA LLC Interests to the Company in exchange for the number of shares of Class A Stock, Class B Stock or Class C Stock, as applicable, set forth on Schedule 2.02 (the "Contribution"). The Company hereby acknowledges that it intends to continue the existence of DWA LLC as a partnership for Federal income tax purposes.

Section 2.03. Residual DW Distribution. (a) On the Separation Date, immediately after consummation of the PDI Merger (as defined in the Separation Agreement), DW shall distribute (in accordance with Article VIII of the Sixth

Amended and Restated Limited Liability Company Agreement of DW) all its right, title and interest in and to all shares of Class A Stock then held by DW (after giving effect to the LLC Employee Distribution (as defined in the Separation Agreement)) to the Members listed on Schedule 2.03(a) hereto, in the amounts set forth on Schedule 2.03(a) (the "Residual DW Distribution").

IN WITNESS HEREOF, the parties hereto have caused this Agreement to be duly executed and delivered as of the date first written above.

DREAMWORKS ANIMATION SKG, INC.,

by

Name:

Title:

Address:

DREAMWORKS L.L.C.,

by

Name:

Title:

Address:

[HOLDCO] LLLP,

by

Name:

Title:

Address:

ATTACHMENT B – JSON output example

```
{
  "party2": " Celgene Corporation",
  "party1": " Johnson Matthey Inc.",
  "title": "custom manufacturing agreement",
  "party1_alias": "JMI ",
  "party2_alias": "Celgene",
  "norms": [
    {
      "party2": "",
      "party1": "",
      "condition": "",
      "action": " be referenced in orders and other correspondence related
hereto as Agreement No",
      "modal_verb": "may",
      "object": "",
      "deontic_modality": "Permission",
      "norm": "This Agreement may be referenced in orders and other
correspondence related hereto as Agreement No",
      "subject": ""
    },
    {
      "party2": "Celgene",
      "party1": "JMI",
      "condition": "",
      "action": " for evaluation and regulatory filing purposes only and ",
      "modal_verb": "shall not",
      "object": "Celgene",
      "deontic_modality": "Prohibition",
      "norm": "JMI supplied the Pilot Phase Material to Celgene for evaluation
and regulatory filing purposes only and Celgene covenants that
such Pilot Phase Material shall not be used for human consumption",
      "subject": "JMI"
    },
    {
      "party2": " Celgene",
      "party1": "",
      "condition": "by Celgene and perform the tests specified
```

```

for Material in Exhibit B",
  "action":" qualify analytical methods as provided by Celgene
and perform the tests specified for Material in Exhibit B",
  "modal_verb":"will",
  "object":"",
  "deontic_modality":"Obligation",
  "norm":"4.6 Testing The parties hereto acknowledge and agree
that JMI will qualify analytical methods as provided by
Celgene and perform the tests specified for
Material in Exhibit B",
  "subject":""
},
{
  "party2":" Celgene",
  "party1":"JMI",
  "condition":"",
  "action":" that all Resolving Agent delivered to JMI shall
include instructions on proper handling requirements and
shall be packaged regulations , , ordinances and statutes",
  "modal_verb":"shall",
  "object":"JMI",
  "deontic_modality":"Permission",
  "norm":"5.2 Celgene shall ensure that all Resolving Agent
delivered to JMI shall include instructions on proper
handling requirements and shall be packaged regulations
ordinances and statutes",
  "subject":"Celgene"
},...

```