

Towards Disambiguating Contracts for their Successful Execution - A Case from Finance Domain

Preethu Rose Anish*, Abhishek Sainani, Nitin Ramrakhiyani, Sachin Pawar,
Girish K Palshikar, Smita Ghaisas

TCS Research, India

{preethu.rose, a.sainani, nitin.ramrakhiyani, sachin7.p, gk.palshikar, smita.ghaisas}@tcs.com

Abstract

Contract management is key to financial services. Contracts lay down rules for doing business or present guidelines and recommendations for maximizing financial advantage of stakeholders in a given scenario. Contracts related to award of projects by companies to vendors, employment contracts, lease agreements, franchise agreement and even prenuptial agreements have significant financial implications. Making sense of contracts is an important step in achieving organizational goals such as building compliant systems, meeting delivery deadlines, avoiding heavy penalties and steering clear of expensive litigation. The complexity of “contracts language” however, makes it difficult to leverage the guidance they intend to offer. Contracts are written *ex ante*, based on forecasts rather than actual results, and may therefore contain ambiguous and incomplete guidance that can result in unintended violations. We address these problems by aiming to automate the disambiguation of contracts using a generalized architecture – R^3 to (1) Recognize important essential information present in contracts, (2) Reason over the information elements to identify their interrelations and uncover ambiguities and inconsistencies, and (3) Render the information in a visual format (for example, Message Sequence Charts) depicting different elements in contractual obligations.

1 Introduction

The digital transformation is well under way in the financial industry. Emerging business models, FinTechs, cryptocurrency, cyber threats, cyber security, data monetization, payments, greater collaboration and the shifting landscape of regulation and technology are all transforming this sector [Scardovi, 2017]. These transformations have accelerated the pace of business in the banking and financial sector leading to huge opportunities but also unpredictable risks.

A contract is a written agreement between two parties that details the terms of a transaction. Even though contracts are

fundamental in every business sector, their criticality is even more pronounced in the financial services for the reasons stated above. Contracts are vital to regulatory compliance and risk management, as they provide a deep insight into every aspect of an organization’s operation. This is especially significant in the financial sector, where, along with the accelerated pace of business, organizations face extremely stringent regulatory compliance requirements [Veerkar, 2018]. Failing to track compliance carries heavy financial risk. To cite just one example, during the calendar year ending 2018, close to fifteen banking firms and/or individuals were fined for breaching the principles of U.K.’s Financial Conduct Authority - FCA (FCA regulates financial firms providing services to consumers and maintains the integrity of the financial markets in the United Kingdom). The total amount of fine collected was close to £60 million. Clearly, a lack of attention to risk and compliance can cost a fortune to organizations and/or individuals.

An important first step towards achieving organizational goals of all participants involved in execution of contracts is to understand the contract documents comprehensively. However, the complexity of the documents makes it difficult to leverage the guidance they intend to offer. While the unintentional complexity in the documents may be due to the inherent peculiarities of natural language itself, the intentional complexity arises because the text aim to address and mitigate divergent and foreseeable, yet, unrealized scenarios in case they occur. Since contracts are written *ex ante*, based on forecasts rather than actual results, they may also contain ambiguous and incomplete guidance which can result in unintended violations and consequently, unfair penalties.

Contracts (similar taxonomically to regulations [Massey *et al.*, 2014]) contain obligations that must be fulfilled by the parties entering into a business agreement, the rights of the parties, permissions granted, exclusions to be made, and exceptions to the business rules. Additionally, contracts are bound by regulations prevalent in the countries where businesses are to operate, thereby inheriting the ambiguities in regulations. Of the different constructs (obligations, permissions, rights, exceptions, and exclusions) in a contract, it is easy to appreciate that obligations are the most demanding. Failure to fulfill contractual obligations can often entail punitive measures such as penalties and lengthy expensive litigations. The ongoing expensive contract breach case be-

*Contact Author

tween Apple and Qualcomm is a case in point [Reuters, 2017] wherein some communications between participants seem to have breached the non-compete clauses in the contract. While it is not yet clear if this breach was intentional, as noted earlier, due to the ambiguous and complex nature of the texts in contracts, or lack of complete information, unintentional breaches are also likely. How can we contribute to avoiding reducing such scenarios?

A number of people from customers' and vendors' organizations are involved in fulfilling contractual obligations. Their roles are mentioned in the documents while specifying the terms and conditions of an obligation. Contracts often also specify governance processes for fulfilling obligations. However, due to the large size of the documents, the convoluted Legalese-like language and the ambiguities, it is difficult to read contracts comprehensively and decipher who needs to do exactly what, while fulfilling an obligation. Disambiguating this complex text and presenting relevant information in a succinct, understandable form would therefore be useful to the participants. Our work on automating the disambiguation is motivated by the difficulties faced by people responsible for fulfilling contractual obligations.

We employ a generalized Recognize-Reason-Render (R3) architecture designed for disambiguating texts present in complex documents such as contracts and regulations. When we process contracts using the R3 architecture, the Recognize layer employs mechanisms to identify important elements such as obligations and corresponding actions and deadlines mentioned if any in the contract. The Reason layer establishes interrelations between the identified elements, for example, actor responsible for a given obligation or a trigger corresponding to an action to be taken for complying with the obligation. The Render layer constructs a user-friendly visual depiction in the form of Message Sequence Charts (MSC) to depict important actions, triggers, and actors involved in complying with contractual obligations.

The rest of the paper is organized as follows. In section II, we provide a brief overview of the R³ architecture. In section III, we present an illustration of disambiguation of a contract using the Recognize-Reason-Render components. Section IV is on related work and section V concludes the paper.

2 Overview of the R³ Architecture

The organizational goals such as avoiding penalties for non-compliance or ensuring no escalations in a project necessitate disambiguation of the documents that are meant to guide different stakeholders in fulfilling their responsibilities. For example, to make sure that a project does not run into escalations and penalties for unacceptable deliveries of products or services, we need to understand the obligations, rights, and the terms and conditions of service from a contract. With the aforesaid automation therefore, we must be able to address the following three aspects: (1) Recognize patterns that point to elements of interest aligned with the goal (for example - linguistic patterns characteristic of obligations and rights), (2) Apply reasoning that allows to derive meaning from the identified elements by determining their interrelations (such as trigger for the onset of an obligation), and (3) render the

output of reasoning in a way that can be understood by people responsible for executing contracts.

3 Applying R³ Architecture for Disambiguating Contracts

In this section, we present an illustration of disambiguation of a contract using the Recognize-Reason-Render components. Fulfilling contractual obligations so as to avoid heavy penalties and loss of credibility is a goal that motivates this disambiguation.

3.1 Recognize Layer

As stated earlier, contracts often outline governance processes for fulfillment of stated obligations. In our observation, a governance process consists of (i) multiple *actors* (which could be human users, organizations, designations or roles); and (ii) various *interactions* among actors, which are typically either *physical actions* (e.g., replace faulty parts) or *communication actions* involving exchange of information, instructions or control (e.g., approve, request, confirm, comply). In addition, a governance process must deal with external or internal (within the organization) *events* (e.g. termination) and *conditions* (e.g., increased costs, business interruption). The interactions are often ordered in specific ways, and may be associated with time expressions indicating durations, deadlines, frequency etc. Our goals are (i) to extract governance processes from a given contract; and (ii) express (or render) each governance process in a simple intuitive, visual form, which is easy for non-legal users to understand and implement in their workflows.

We categorize the information in contractual Obligation into TRIGGER, ACTION, ACTOR and TIMEX. We take a sample contract pertinent to financial construction and explain the categorization and the method that we have devised for extracting these pieces of information from the contracts document. Financing constructions is an important line of business for banks and housing finance companies. Financing a construction involves a legal contract or loan agreement, which typically includes governance processes for various eventualities, such as delays in completing the construction, or poor quality of construction. An example text in a legal contract¹ (available in public domain) containing description of a governance process is given below (entities are annotated).

[When [Contractor]_{Actor}'s Work is completed]_{Trigger}, [Contractor]_{Actor} shall [give written notice]_{Action} to [Owner]_{Actor} and/or [Bank]_{Actor}. [Owner]_{Actor} and/or [Bank]_{Actor} shall [make final inspection]_{Action} of [Contractor]_{Actor}'s work [within seven days]_{Timex} [after receipt of notice of completion]_{Trigger} and [upon acceptance thereof]_{Trigger} by [Owner]_{Actor} and [Bank]_{Actor}, [payment shall be made of the remaining balance due]_{Action}. Such [acceptance shall not be unreasonably withheld]_{Action} and

¹From sample contract at http://www.basnettdbr.com/pdfs/ConstCont_101117.pdf

[in the event [Owner]_{Actor} and/or [Bank]_{Actor} refuses to accept]_{Trigger}, [Owner]_{Actor} and/or [Bank]_{Actor} as the case may be shall [within said seven day period]_{Timex} [notify [Contractor]_{Actor} in writing of such refusal]_{Action} and shall [specify the reasons therefore]_{Action}. [Contractor]_{Actor} shall [within the next ten days]_{Timex} [take appropriate steps to remedy any deficiencies set forth as reasons for refusal]_{Action} and [upon completion thereof to the satisfaction of [Owner]_{Actor} and/or [Bank]_{Actor}]_{Trigger} shall [be entitled to prompt payment of any remaining balance due]_{Trigger}.

Using the above example text, we first provide a definition of each of the element we extract from contracts.

- **OBLIGATION (OBLG)** – a statement in the legal contract requiring a specific commitment from a user, which she must be made aware of. A governance process is expressed through OBLG statements.
- **TRIGGER** - a part of an OBLG statement containing a condition or event (e.g., an alert, an abnormal situation) that must be monitored by a user. Example: When contractor’s work is completed.
- **ACTION** - a part of an OBLG statement containing a concrete action that must be taken by a user. Example: give written notice. ACTION is almost always paired with the corresponding trigger. Example, [When contractor’s work is completed]_{TRIGGER}, [Contractor shall give written notice to Owner and Bank]_{ACTION}.
- **ACTOR** - typically, people, roles or organizations. Example: Bank, Contractor, Owner.
- **TIMEX** - time expressions (e.g., deadlines) associated with actions and triggers. Example: within seven days

We now outline our method for extracting the elements of a governance process from the text within a legal contract.

OBGL Classification: We use Multinomial Naïve Bayes as the classifier to predict the class label (OBGL or NOT_OBGL) for each sentence in the given contract. For identifying obligations, the classifier obtained a Precision, Recall and F-score of 91.8%, 91% and 91.3% respectively.

Trigger Extraction: Next, we detect and extract the text fragment corresponding to TRIGGER in the given OBLG sentence *S*, using rule-based *information extraction* (IE). First, we create an enriched representation of *S*, by adding information like POS tags, phrase structures, dependency relations, semantic roles and named entities. We then write simple regex patterns on this enriched text to extract triggers, which makes the patterns much more general and does not depend too much on detailed structure of the trigger text fragment. For example, triggers after receipt of notice of completion, or upon completion thereof to the satisfaction of, are reliably indicated by *cue words* such as when or upon and the rest of the trigger bears a specific dependency relation with this cue word.

For extracting triggers from the obligation statements, we obtained a precision of 86.5% and a recall of 80%.

Action Extraction: Next, we detect and extract the text fragment corresponding to ACTION in the given OBLG sentence *S*, using rule-based IE, similar to that used for TRIGGER extraction. For ACTION extraction, we obtained a Precision of 76.1

Actor Extraction: For ACTOR extraction, we follow the actor extraction algorithm in [Patil *et al.*, 2018]. The approach not only identifies canonical mentions of various actors but also their aliases mentioned as pronouns and generic noun phrases (NP). The algorithm utilizes WordNet hypernym structure to identify actor mentions. Then it uses first order logic rules containing linguistic knowledge to infer aliases in Markov Logic Networks framework.

Timex Extraction: We use the HeidelbergTime tool [Strotgen and Gertz, 2010] to extract time expressions from OBGL statements. It defines a temporal expression as a tuple of three elements: time expression as it occurs in the textual document, type of expression, and its value in a normalized form. To extract a temporal expression, this tool uses hand-crafted rules such that each rule consists of rule for identifying the time expression, information about the type of expression, and a function to normalize the identified time value. It further uses post-processing steps to resolve underspecified values and remove invalid temporal expressions.

3.2 Reason Layer

Having recognized and extracted the governance process elements described in sub-section 2.1, we employ the Reason layer to establish associations between them. We extract the governance process and map it to a Message Sequence Chart (MSC) [Mauw, 1997]. Every actor gets a separate timeline in the MSC. Actors are associated with the actions they initiate or receive. In the example contract, Contractor is the initiator of the action deliver a written notice and Owner and Bank are recipients (beneficiaries) of this action.

Since we extract the governance process and model it as an MSC, we can potentially identify ambiguities. In the example process, the phrase prompt payment is ambiguous because although prompt is a time indicator, it cannot be associated with any concrete deadline for payment. Use of the word unreasonably yields a similar ambiguity. Such ambiguities can be brought to the notice of the users, who can subsequently clarify them. Apart from ambiguities, one can also identify *incompleteness* issues with the governance process in the contract. For example, the example text does not include the governance process that corresponds to the *time-out* event; e.g., what action is to be taken if the Owner does not notify the Contractor of the refusal to accept the work, within ten days. This kind of incompleteness can potentially lead to lack of action on part of the participants (Owner and Contractor in this case). This kind of incompleteness can potentially lead to lack of action on part of the participants. The lack of required action may be wrongly interpreted as a violation leading to unfair penalties. An early identification of the absence of information can mitigate this risk by way of motivating discussions among participants about the required action and avoid unpleasant and inequitable situations.

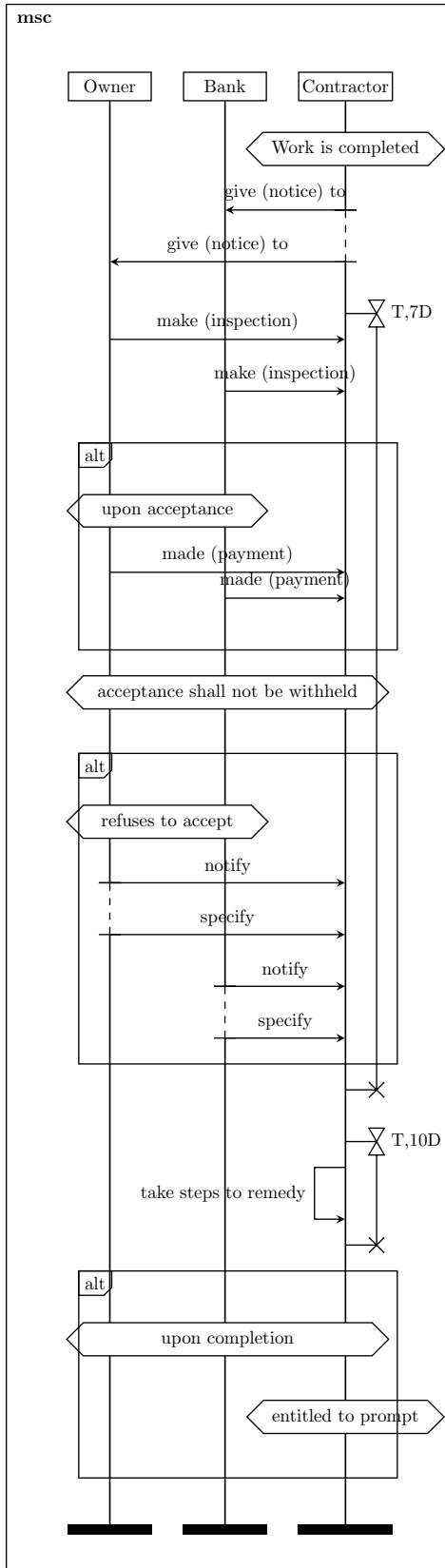


Figure 1: MSC depicting an example governance process

3.3 Render Layer

Once the mentions of the above types of entities are extracted and mapped to the governance process as described above, we use a simple visual (yet mathematically rigorous) notation, - MSC to depict the process. Fig. 1 depicts the example governance process depicted as an MSC.

The MSC notation is an international standard [Mauw, 1997] and is similar to the sequence diagram notation in UML. We chose MSC because, unlike other notations for representing business processes (for e.g., BPMN and BPEL), MSC has a clearer and simpler rigorous semantics, making formal analysis and verification of process models much easier. Interactions between actors are mapped to *messages* (directed arrows) in the MSC, where the initiator becomes the sender of the message and the recipient becomes the receiver of the message. We label each message with the associated action in the governance process. We map multiple messages to a *co-region* in the MSC on the sender's timeline, when it is not clear in which order the actor sends these messages. We map each trigger to a *condition* in the MSC formalism, which is labeled with text that indicates a state of the actor(s) and which is depicted as a hexagon containing the text of the trigger. *Timer* facility in the MSC formalism is used to denote whenever something is required to happen within a stipulated time (*deadlines*). MSC allows the ALT-box notation to specify alternate paths in the interactions among actors.

The MSC notation visually shows a *timeline* for each actor, in which the events (i.e., messages) that the actor participates in are depicted chronologically. Thus, the interactions depicted on a particular actor's timeline need to be temporally ordered. While text order (i.e., the order in which interactions are mentioned in the text), is a good initial approximation, in reality, many constructs in English can be used to alter the position of some interaction on the actor's timeline. We have designed an ILP-based event ordering algorithm, which maps the event ordering problem to an optimization problem, and the optimal solution is used to depict these interactions on the actors' timelines.

The algorithm to generate MSCs deals with a number of issues such as actor *co-reference* (an actor may be mentioned in many different ways, including pronouns), identifying the sender (i.e., initiator) and receiver(s) (i.e., recipients) for each action, and temporal ordering of messages. When the sender (or receiver) cannot be inferred, our algorithm uses *environment* as a generic actor for this purpose. Actions can also be co-referenced (e.g., upon acceptance thereof), which is the well-known NLP problem of *event co-reference*. For details of MSC extraction algorithm, see [Palshikar *et al.*, 2019a; Palshikar *et al.*, 2019b].

3.4 Related Work

Disambiguation of complex documents such as regulations and contracts has been an evolving research area. Researchers have approached this problem from various angles such as extracting obligations and rights [Breux *et al.*, 2006], identifying and classifying ambiguities [Massey *et al.*, 2014], integrating information from multiple sources to make sense of ambiguous texts [Ghaisas *et al.*, 2018] and formalizing contracts [Lomuscio *et al.*, 2012]

Information extraction (IE) from documents has been employed extensively for uncovering important details present in complex documents thereby disambiguating them [Palshikar, 2012]. Most of the text analytics work on contracts [Lippi *et al.*, 2017; Indukuri and Krishna, 2010; Gao and Singh, 2014; Curtotti and McCreath, 2010; Gao *et al.*, 2012; Hachey and Grover, 2004] is focused on extracting either entire sentences or clauses, rather than extracting specific contract elements. We found very few references on extraction of specific contract elements.

Gao and Singh [2014] proposed a topic modeling based technique to extract business events and their temporal constraints from contract text. Since our purpose is to disambiguate complex text, we extract elements at a much finer granularity, reason over the extracted elements and render them as a user-friendly visual depiction. Kalia *et al.* [2013] proposed an approach for extracting commitments from email and chat conversations. Their approach deals with ad hoc communications unlike our approach where the focus is on legal contracts and ours is a complex problem given that legal texts are written in natural language in its full complexity. Chalkidis and Androusoopoulos [2017] employed deep learning methods to extract various contract elements. They extract 11 elements namely Title, Party, Start, Effective, Termination, Period, Value, Gov. law, Jurisdiction, Legislative reference and headings. However, they neither reason nor render it the way we do. Biagioli *et al.* [2005] extracted provision types (Definitions, Obligations etc.) and their arguments (for instance, for obligations, the arguments are Addressee, Action, Third party) from law documents. This work of Biagioli *et al.* [2005] is close to ours as they also extract obligations, actions and actors. However, they do not extract conditions (Triggers in our case) and further they do not aim to render them in a user-friendly visual depiction.

4 Conclusion

We have been able to employ the R3 architecture to demonstrate that it is possible to recognize the crucial information elements in a contracts document, reason over them to determine their interrelationships with a high accuracy and render the result of this reasoning in a user-friendly visual form using MSCs. Importantly, the MSC representation also lets us identify ambiguities and inconsistencies in the text thereby allowing for informed discussions around contractual clauses that likely to remain unresolved in meaning and lead to unintentional violations and painful punitive actions.

Given the challenges and the crucial digital disruption happening in the financial domain, we took a contracts document related to financing construction for our experiments. Since our approach does not rely on any domain specific ontologies, it is domain independent and therefore is generalizable across domains.

In disambiguating contracts and presenting them as MSCs, we take a significant step towards making it easier for participants to fulfill their respective obligations. We will next validate the acceptability of the rendering with practitioners to gain their insights on determining the future direction of this work. Further, we would apply our extraction technique

on larger datasets from other domains to strengthen our generalizability claim.

References

- [Scardovi, 2017] Scardovi, C., 2017. Digital transformation in financial services. Springer.
- [Veerkar, 2018] Anand Veerkar, 2018. Why the Banking Industry Needs Enterprise Contract Management Software, <https://www.icertis.com/blog/banking-contract-management/>, Last accessed on 9-04-2019
- [Massey *et al.*, 2014] A. K. Massey, R.L. Rutledge, A.I. Antón, P.P. Swire, Identifying and classifying ambiguity for regulatory requirements, *In Proc. Of 22nd IEEE international Requirements Engineering Conference (RE)*, 2014, pp. 83-92
- [Reuters, 2017] From gadget news NDTV at: <https://gadgets.ndtv.com/mobiles/news/qualcomm-sues-apple-for-breach-of-contract-says-it-shared-information-with-intel-1770592>, Last accessed on 09-04-2019
- [Mauw, 1997] S. Mauw, 1997. ITU-TS Recommendation Z. 120: Message Sequence Chart (MSC).
- [Palshikar *et al.*, 2019a] G.K. Palshikar, N. Ramrakhiyani, S. Patil, S. Pawar, S. Hingmire, V. Varma, P.Bhattacharyya, Extraction of Message Sequence Charts from Software Use-Case Descriptions, *The 2019 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT 2019 Industry Track)*, 2019, Minneapolis, USA.
- [Palshikar *et al.*, 2019b] G.K. Palshikar, S. Pawar, S. Patil, S. Hingmire, N. Ramrakhiyani, H. Bedi, P. Bhattacharyya, V. Varma, Extracting Message Sequence Charts from History Narratives, *Workshop on Narrative Understanding co-located with NAACL-HLT 2019*.
- [Patil *et al.*, 2018] S. Patil, S. Pawar, S. Hingmire, G.K. Palshikar, V. Varma, P. Bhattacharyya, Identification of Alias Links among Participants in Narratives, *Association of Computational Linguistics (ACL) 2018*.
- [Breux *et al.*, 2006] T. D. Breux, M. W. Vail, and A. I. Anton, A.I., Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. *In Requirements Engineering, 14th IEEE International Conference* (pp. 49-58). IEEE.
- [Ghaisas *et al.*, 2018] S. Ghaisas, A. Sainani, A. P.R. Anish, P.R., 2018, Resolving ambiguities in regulations: towards achieving the Kohlbergian stage of principled morality, *In 2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pp. 57-60. IEEE
- [Palshikar, 2012] G. K. Palshikar, Techniques for named entity recognition: a survey, in S. Bruggemann, C. D'Amato (Ed.s), *Collaboration and the Semantic Web: Social Networks, Knowledge Networks and Knowledge Resources*, IGI Global, 2012, pp. 191-217.

- [Lippi *et al.*, 2017] M. Lippi, P. Palka, G. Contissa, F. Lagoia, H. W. Micklitz, Y. Panagis, G. Sartor, P. Torroni, Automated Detection of Unfair Clauses in Online Consumer Contracts, *Legal Knowledge and Information Systems*, 2017, p.145.
- [Indukuri and Krishna, 2010] K. V. Indukuri and P. Radha Krishna, Mining e-contract documents to classify clauses, in *Proc. of the 3rd Annual ACM Bangalore Conference (Compute)*. Bangalore, India: ACM, 2010, pp. 1–5.
- [Gao and Singh, 2014] X. Gao, M. P. Singh, Extracting normative relationships from business contracts, *In Proc. of the international conference on Autonomous agents and multi-agent systems*, 2014, pp. 101-108
- [Curtotti and McCreath, 2010] M. Curtotti, E. McCreath, Corpus based classification of text in Australian contracts, *In Proc. of the Australasian Language Technology Association Workshop*, pp. 18–26, Melbourne, Australia, 2010
- [Gao *et al.*, 2012] X.Gao, M.P.Singh, P.Mehra, “Mining business contracts for service exceptions”, *IEEE Transactions on Services Computing*, 5, pp. 333–344, 2012
- [Hachey and Grover, 2004] B. Hachey, C. Grover, Sentence classification experiments for legal text summarisation, *In Proc. 17th Annual Conference on Legal Knowledge and Information Systems (Jurix-2004)*, pp. 29-38.
- [Gao and Singh, 2014] X. Gao, M.P. Singh, Mining contracts for business events and temporal constraints in service engagements, *IEEE Transactions on Services Computing*, 2014, 7(3), pp.427-439
- [Chalkidis and Androutsopoulos, 2017] I. Chalkidis, I. Androutsopoulos, A Deep Learning Approach to Contract Element Extraction, *In JURIX*, 2017, pp. 155-164
- [Biagioli *et al.*, 2005] C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni, C. Soria, “Automatic Semantics Extraction in Law Documents”, *In Proc. of the 10th International Conference on Artificial Intelligence and Law*. Bologna, Italy, 2005, pp. 133–140
- [Strotgen and Gertz, 2010] J. Strotgen, M. Gertz, HeidelbergTime: high quality rule-based extraction and normalization of temporal expressions, *Proc. Fifth Int. Workshop on Semantic Evaluation, ACL 2010*, pp. 321-324, 2010
- [Kalia *et al.*, 2013] A. Kalia, H. Motahari Nezhad, C. Bartolini, M. Singh. Monitoring commitments in people-driven service engagements. *Proc. SCC*, pp. 160–167 2013
- [Lomuscio *et al.*, 2012] A. Lomuscio, H. Qu, M. Solanki. Towards verifying contract regulated service composition. *JAAMAS*, 24(3):345–373, 2012