Limitations of current techniques to detect Obligations and Rights in SLA *

Elena Molino-Peña[1][0000−0001−7024−5300], José María García[1,2][0000−0002−0303−2740], and Antonio Ruiz-Cortés[1,2][0000−0001−9827−1834]

[1] Smart Computer Systems Research and Engineering Laboratory (SCORE), Universidad de Sevilla, Seville, Spain
[2] Research Institute of Informatics Engineering (I3US), Universidad de Sevilla, Seville, Spain
mmolino@us.es, josemgarcia@us.es, aruiz@us.es

Abstract. The need to know and understand the obligations and rights contained in service agreements (also known as Customer Agreement) has generated interest in the industry, and there are numerous projects and tools available for the automatic detection and interpretation of obligations and rights. This is especially beneficial for the customer, as it allows them to automatically know the commitments and risks associated with the use of cloud services, as well as for the provider, who can detect potential liability gaps in the agreement. However, existing tools are only able to extract and partially identify this information. In this study, three limitations have been identified in the patterns proposed by some of the recent techniques to extract information from service level agreements (SLAs). In particular, situations in which some obligation or right is not detected, causes for which they can be misclassified and scenarios in which the actor that performs the action is not detected. In addition, a possible solution to these obstacles is proposed.

Keywords: Obligations · Rights · Service Level Agreements.

1 Introduction

The increasing interest in having tools that provide answers to questions that arise before and after signing a Customer Agreement (CA) has driven the development of this work [19]. Patrick Hsu's tweet, as seen in Figure 1, has had a great impact on the community, generating numerous comments sharing solutions for creating, analyzing clauses or summarizing agreements. One of the tools proposed in this thread is "Terms of Services; Didn't Read" [10], which aims to solve the problem of users not reading the conditions imposed in the

* This work has been partially supported by the following grants: PID2021-126227NB-C21, PID2021-126227NB-C22, TED2021-131023B-C21, and PDC2022-133521-I00 which are funded by MCIN/AEI/10.13039/501100011033 and "ERDF a way of making Europe"; and grant PYC20 RE 084 US, which is funded by Junta de Andalucia/ERDF,UE

This work is licensed under a Creative Commons "Attribution-NonCommercial-NoDerivatives 4.0 International" license.

agreements. To do this, they study the conditions of the most commonly used services agreements and evaluate each of their clauses at different levels, see Figure 2. For this preliminary work, we have focused on the automatic extraction of information in the SLA, which is one of the parts of the CA that has received the most attention to date.

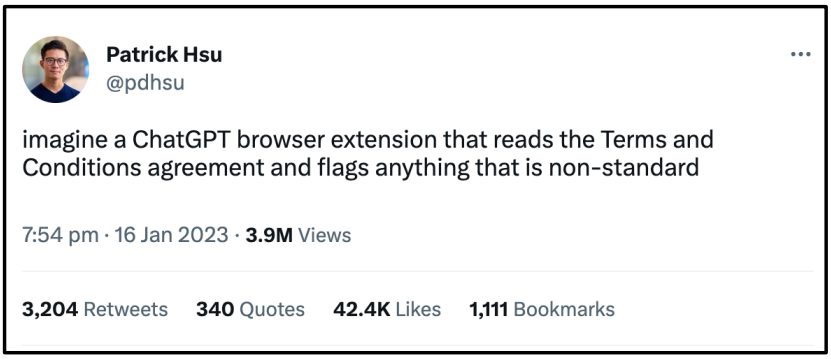

Fig. 1. Tweet from Patric Hsu on the possibility of using ChatGPT in the context of CA [19]

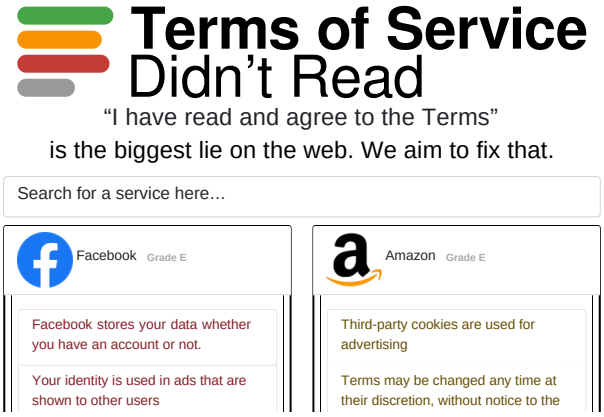

Fig. 2. Excerpt from the project website "Terms of Services Didn't Read"

Taking as a reference the cloud service level agreement model funded by the European Commission [3], an SLA can be understood as a contractual document between the customer and the supplier, defining measurable operational characteristics and their performance in relation to the quality of the service

This work is licensed under a Creative Commons "Attribution-NonCommercial-NoDerivatives 4.0 International" license.

to be achieved by the provider. Typically, a penalty mechanism is specified to promote compliance with the SLA so that the agreement has a clear meaning and impact in practice. Furthermore, the obligations and rights of each of the parties involved are established.

Currently, the customer must analyze and compare the agreements created by different providers, which makes this mainly manual process, tedious, time-consuming, and error-prone. Moreover, although in the literature, different techniques have been applied to automatically extract information from SLAs [4,17], the lack of a clear syntactic structure in sentences limits the ability to detect it accurately, in particular, the obligations, rights, and the actor involved.

Other works that fall within the research line of automated analysis of CA aim at developing catalogs of analysis operations that allow extracting useful information from contracts, SLAs [8,14,15,16], pricing [7] and terms of use [18,20]. Analogous analysis operations have been defined in the context of the automated analysis of feature models [1], microservices architecture [6] and OAS [11].

The rest of the article is organized as follows. Section 2 describes obligations and rights in SLAs and the extraction of these in the Google Compute Engine SLA, Section 3 specifies limitations identified in current approaches, Section 4 discusses our initial solution for detecting obligations, rights, and the actor in SLAs and Section 5 defines the conclusions and future work.

2 Obligations and Rights in SLAs

This section defines the obligations and rights along with a real example of extracting obligations and entitlements in the Google Compute Engine SLA.

2.1 Definition

According to Merriam Webster in law, an obligation is "a promise, acknowledgment, or agreement (as a contract) that binds one to a specific performance (as payment)", in this context it refers to responsibilities [12]. In turn, a right is "something to which one has a just claim", such as stated in the agreement [13].

Typical customer obligations may include paying for cloud services, complying with applicable laws and regulations, using the services in accordance with the acceptable use policy, maintaining the security of its own data and systems, or complaining about the provider's non-compliance. However, the provider's obligations may lie in other aspects such as providing the services at an expected level of performance, complying with applicable laws and regulations, providing technical support for notification of any incidents, providing financial compensation to the customer in the form of service credits, and notifying changes to the agreement or services.

In terms of rights, the most representative from the customer's point of view may be to receive credits compensation if the supplier does not meet the service level objectives (SLOs) or other terms agreed upon in the contract. In addition, the client may be entitled to occasional assistance from the provider or help with

This work is licensed under a Creative Commons "Attribution-NonCommercial-NoDerivatives 4.0 International" license.

data management and privacy. On the other hand, the provider usually has the right to modify the agreement or the services it offers and to suspend them if the customer breaches the terms of service.

2.2 Google Compute Engine SLA

This Google SLA outlines the performance expectations for Compute Engine's hosting and computing service. It is divided into five sections. Firstly, the provider's commitments to the service offered and the possible compensations available to the customer in case of non-compliance are defined, see Figure 3. Next, the necessary terms and definitions are provided. Third, the procedure is specified for customers to request credits. After that, the maximum amount of credits that a customer can receive is detailed. Finally, the exceptions for which the SLA is not applicable are established [9].

Compute Engine Service Level Agreement (SLA)

During the Term of the agreement under which Google has agreed to provide Google Cloud Platform to Customer (as applicable, the "Agreement"), the Covered Service will provide a Monthly Uptime Percentage to Customer as follows (the "Service Level Objective" or "SLO"): S_1

Covered Service	Monthly Uptime Percentage
Instances in Multiple Zones	>= 99.99%
A Single Instance	>= 99.5%
Load balancing	>= 99.99%

If Google does not meet the SLO, and if Customer meets its obligations under this SLA, Customer will be eligible to receive the Financial Credits described below. Monthly Uptime Percentage and Financial Credit are determined on a calendar month basis per Project or, for a Single Instance, per instance. This SLA states Customer's sole and exclusive remedy for any failure by Google to meet the SLO. Capitalized terms used in this SLA, but not defined in S_2

Fig. 3. Excerpt from the Compute Engine SLA which contains S_1 and S_2

The obligations and rights detected in this agreement will be detailed following a pattern. First, they are enumerated using the following notation: S_i where i is the order in which the statements appear in the agreement. Then, the sentence, its type, and a brief explanation of the selected type along with its agent are specified.

This work is licensed under a Creative Commons "Attribution-NonCommercial-NoDerivatives 4.0 International" license.

- S_1: "Google has agreed to provide Google Cloud Platform to Customer (as applicable, the "Agreement"), the Covered Service will provide a Monthly Uptime Percentage to Customer as follows", mentioned in the Figure 3.
  - Identified type: obligation.
  - Agent and justification: the provider must offer the contracted services and meet the Monthly Uptime Percentage metric.
- S_2: "Customer will be eligible to receive the Financial Credits described below.", mentioned in the Figure 3.
  - Identified type: right.
  - Agent and justification: the customer shall be entitled to receive Financial Credits if the supplier fails to comply with their commitments.
- S_3: "Customer must notify Google technical support within 60 days from the time Customer becomes eligible to receive a Financial Credit."
  - Identified type: obligation.
  - Agent and justification: the customer must apply for the credits in order to be eligible to receive them.
- S_4: "Customer must also provide Google with server log files showing loss of external connectivity errors and the date and time those errors occurred."
  - Identified type: obligation.
  - Agent and justification: the customer must send information to the supplier to be eligible for the credits.
- S_5: "The aggregate maximum number of Financial Credits to be issued by Google to the customer for any and all Downtime Periods that occur in a single billing month will not exceed the amount due by the customer for the respective Covered Services in the Regions that did not meet SLO for the applicable month."
  - Identified type: obligation.
  - Agent and justification: the provider must send the customer the corresponding credits according to his monthly fee.
- S_6: "Financial Credits will be made in the form of a monetary credit applied to future use of the service and will be applied within 60 days after the Financial Credit was requested."
  - Identified type: obligation.
  - Agent and justification: the provider must send the credits in a specific period.

As we can see, each of these sentences is different and does not conform to a specific syntactic pattern, which poses a challenge for automated detection.

3 Drawbacks of current proposals

The current solutions proposed to extract information from SLAs consist of applying natural language processing techniques that do not involve Transformers [4]. These techniques rely on predefined patterns to detect desired parts of the text or to process the text sequentially. This means that they cannot process the

This work is licensed under a Creative Commons "Attribution-NonCommercial-NoDerivatives 4.0 International" license.

complete text and do not have the attention mechanism used by Transformers, which would allow them to leverage the full meaning of the sentence.

A similar approach to this idea is proposed by Breaux et al. [2], who focus on semantic parameterization of obligations and rights in privacy policy documents. They assume that sentences containing "will" could be considered obligations, and those containing "may" must refer to a right. On the other hand, in the case of Natolana et al. [17], they propose a comprehensive framework for the extraction of knowledge from the SLA by combining and improving previously applied text mining and semantic web techniques. Specifically, one of the tasks they perform is to extract obligations and rights in SLAs, taking the following assumptions:

- Sentences containing the word "must", "should", "will" impose obligations on the actor in the sentence.
- Sentences are detected if they adhere the following rule: <Noun/Pronoun> <modal verb> <verb>

Although most sentences are in accordance with the guidelines proposed by Natolana, there are exceptions where these rules are not followed. Some sentences may not require a modal verb, but indicate an obligation or a right. Therefore, this restriction will be referred to as a missing statement. Also, there are instances where sentences that convey a right can include the verb "will"; this limitation will result in misleading classifications.

As mentioned above, the methods that have been used to date are limited in the automatic detection of SLA obligations and rights. If the framework proposed by Natolana were applied to the Google SLA discussed in Section 2.2, some statements would not be correctly identified.

For example, the framework is not able to detect S_1 ("Google has agreed to provide Google Cloud Platform to Customer..."), because the verb in the sentence does not include a modal verb, but an auxiliary verb plus the main verb, resulting in a missing statement. Another point to consider is the case of S_2 ("Customer will be eligible to receive the Financial Credits described below."), it would not be classified correctly because even though it contains the verb "will", it indicates a right for the client. Thus, it is a misleading classification.

Another task proposed by Natolana et al. is the detection of the actor who is going to perform the action. In this case, they evaluate the category of each word and determine a possible actor by labeling the words of each sentence using the part-of-speech tagging technique. The model they suggest is a loop that checks two possible condictions for each word in the sentence:

- If previous word is the nominal subject and the current word is an auxiliary verb, then the actor is the preceding one. Example sentence: "At our discretion, **we may** issue the Service Credit to the credit card...".
- In the other case, If the preceding word is the main verb and the current word is the noun that receives the action, then the actor is the current word. Example sentence: "Your failure to provide the request will **disqualify you** from receiving a Service Credit".

This work is licensed under a Creative Commons "Attribution-NonCommercial-NoDerivatives 4.0 International" license.

The main problem with this pattern is to detect sentences in passive voice such as S_5 ("maximum number of Financial Credits to be issued by Google to the customer...") and S_6 ("Financial Credits will be made in the form of a monetary credit...") discussed in Section 2.2. In fact, passive voice sentences can sometimes omit the actor, making them difficult to detect by patterns, as in S_6.

These limitations have been identified when we tried to replicate the work done by Natolana et al. [17]. In our work, we used their detection criteria to generate a corpus of obligations and rights, but some of the sentences were not correctly classified and others were not even detected.

Although these drawbacks have been detected in the previous ad-hoc proposals, their application can benefit the client by providing an initial idea of the obligations and rights that both parties would have, facilitating the comparison between the SLAs from different providers.

4 Initial Solution Proposal

Our initial proposal aims to improve the ability to detect obligations, rights, and the actor involved, which it will allow reasoning about the information extracted and provide value to both the customer and the provider. The important advances in natural language processing and deep learning techniques lead us to consider that an excellent alternative to extract information is to apply a Large Language Model (LLM).

Most of these models are prepared to address a variety of tasks, such as text classification, translation, question-answering, or text summarization. These models have been trained with a large corpus, which allows them to be flexible solutions for different domains. However, their main limitation is that they are not specifically designed to solve a particular problem. Therefore, it is usual to perform a fine-tuning process of a pre-trained model with a specific dataset to address the required task [5].

Given the diversity of options, we decided to fine-tune a model specifically for classifying statements into the categories of obligation, right, or neither. In this context, due to the limited amount of labeled data in the agreement domain, it has been necessary to generate a corpus [3] of SLA sentences from various providers, classified into the categories mentioned above. Despite performing a manual tagging of 53 SLAs from 18 leading cloud providers, it is important to note that the available dataset is still quite limited.

Therefore, we decided to use an innovative approach known as SetFit for training the *paraphrase-mpnet-base-v2 model* [4], a modification of the Bert model [21]. SetFit is a framework that has proven to be highly effective for fitting pre-trained models, even when a limited number of examples are available, by applying contractive learning techniques [22]. The result is a model [5] specially trained to detect obligations and rights in the SLA domain.

[3] https://huggingface.co/datasets/marmolpen3/sla-obligations-rights
[4] https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2
[5] https://huggingface.co/marmolpen3/sla-obligations-rights

This work is licensed under a Creative Commons "Attribution-NonCommercial-NoDerivatives 4.0 International" license.

After having a model that can extract a catalog of sentences indicating an obligation or right from any SLA, a query is made to the OpenAI API to extract the actor using the *gpt-3.5-turbo model* [6]. The process consists of generating a prompt with the extracted obligations or rights, and as a result, a list with the actors that must perform the action is obtained.

The proposed solution for information extraction automatically assigns the phrase type based on the semantics of the input data, avoiding the dependency on specific patterns. However, it is important to note that this assignment may be affected by the knowledge and possible biases of the classifier. In addition, one of the most robust models for actor detection is used, reducing the problem found in passive sentences.

We have developed an initial prototype [7] following this approach. Although we need to validate this solution further, we have obtained promising results. The information extracted allows us to reason about the terms of the agreement. Based on a set of metrics, we can, for instance, analyse whether the agreement is beneficial to the customer or the provider, as well as the risks involved in subscribing to the cloud service, among other analysis operations.

5 Conclusions and future work

Automatic extraction of obligations and rights is essential to ensure transparency and responsibilities in the agreement, allowing users to understand the commitments and risks involved in subscribing to a cloud service. In addition, it is useful to answer different analysis operations, for example, to identify gaps in the responsibilities defined in the agreement. To achieve these objectives, a system capable of detecting the obligations and rights in the SLA by itself is required, taking into account the semantic meaning of phrases and the identification of the actor who must perform the corresponding action. An alternative is to apply NLP techniques based on the Transformer architecture or a hybrid of the previous and more current techniques.

Throughout this initial research on automatic detection of rights and obligations in SLAs, we have found three limitations in existing proposals: missing statements, misleading classifications, and scenarios in which the actor that performs the action is not detected. However, we have also observed that current tools, such as fine-tuning pre-trained models with large amounts of data, could overcome these limitations.

To address these challenges, our future work includes further exploring the use of models based on transformer architecture by improving the automatic extraction of obligations, rights, or detecting any other important information contained in the agreements. This will require collecting a larger amount of categorized data, training an existing model, and carrying out a deep evaluation, preparing the model to solve a specific task effectively. Finally, we will need to define the analysis operations and the metrics to evaluate them.

[6] https://platform.openai.com/docs/models/gpt-3-5
[7] https://github.com/isa-group/iContracts

This work is licensed under a Creative Commons “Attribution-NonCommercial-NoDerivatives 4.0 International” license.

Acknowledgements. Authors would like to thank Juan Carlos Alonso Valenzuela for his help in the comprehension of natural lenguage processing techniques. We would also like to thank the reviewers for their valuable comments.

References

1. Benavides, D., Segura, S., Ruiz-Cortés, A.: Automated analysis of feature models 20 years later: A literature review. Information systems **35**(6), 615–636 (2010)
2. Breaux, T., Anton, A.: Analyzing goal semantics for rights, permissions, and obligations. In: 13th IEEE International Conference on Requirements Engineering (RE'05). pp. 177–186 (2005). https://doi.org/10.1109/RE.2005.12
3. Committee, L.T.: "Cloud Model Service Level Agreement". Tech. rep., Funded by the European Commission and implemented by the European Bank (2020)
4. De Marco, L., Ferrucci, F., Kechadi, T., Napoli, G., Salza, P.: Towards automatic service level agreements information extraction. pp. 59–66 (04 2016). https://doi.org/10.5220/0005873100590066
5. Face, H.: The hugging face course, 2022. https://huggingface.co/course (2022), Accessed 04/18/2023
6. Fresno-Aranda, R., Fernández, P., Durán, A., Ruiz-Cortés, A.: Semi-automated capacity analysis of limitation-aware microservices architectures. In: Proc. 19th International Conference on the Economics of Grids, Clouds, Systems, and Services. pp. 75–88. Springer (2023)
7. Fresno-Aranda, R., Fernández, P., Ruiz-Cortés, A.: A catalogue of analysis operations for api pricing plans. In: Submited to XVIII Jornadas de Ingeniería de Ciencia e Ingeniería de Servicios. SISTEDES (2023)
8. García, J.M., Fernández, P., Pedrinaci, C., Resinas, M., Cardoso, J., Ruiz-Cortés, A.: Modeling service level agreements with linked USDL agreement. IEEE Transactions on Services Computing **10**(1), 52–65 (2016)
9. Google: Compute engine service level agreement (sla). https://cloud.google.com/compute/sla (2021), accessed 05/06/2023
10. Hugo Roy: Terms of Services. https://tosdr.org/en/frontpage#ratings (2023), Accessed: 07/05/2023
11. Martin-Lopez, A., Segura, S., Müller, C., Ruiz-Cortés, A.: Specification and automated analysis of inter-parameter dependencies in web APIs. IEEE Transactions on Services Computing **15**(4), 2342–2355 (2021)
12. Merriam-Webster: Obligation. https://www.merriam-webster.com/dictionary/obligations#legalDictionary (2023), Accessed: 05/05/2023
13. Merriam-Webster: Right. https://www.merriam-webster.com/dictionary/right#legalDictionary (2023), Accessed: 05/05/2023
14. Müller, C., Gutierrez, A.M., Fernandez, P., Martín-Díaz, O., Resinas, M., Ruiz-Cortés, A.: Automated validation of compensable slas. IEEE Transactions on Services Computing **14**(5), 1306–1319 (2021)
15. Müller, C., Oriol, M., Franch, X., Marco, J., Resinas, M., Ruiz-Cortés, A., Rodriguez, M.: Comprehensive explanation of sla violations at runtime. IEEE Transactions on Services Computing **7**(2), 168–183 (2013)
16. Müller, C., Resinas, M., Ruiz-Cortés, A.: "Automated Analysis of Conflicts in WS-Agreement". IEEE Transactions On Services Computing **7**(4), 530–544 (2014)
17. Natolana-Ganapathy, D., Pande-Joshi, K.: "A Semantically Rich Framework to Automate Cloud Service Level Agreements". IEEE Transactions on Services Computing (2022)

This work is licensed under a Creative Commons "Attribution-NonCommercial-NoDerivatives 4.0 International" license.

18. Otto Hanson, J.D.: TermScout. https://www.termscout.com/, Accessed: 27-04-2022
19. Patrick Hsu: Request an extension that reads the Terms and Conditions agreement. https://twitter.com/pdhsu/status/1615059981044441088?s=12&t=aohMYFx2OGhmajYpbxs9XA (2023), Accessed: 07/05/2023
20. Peña, E.M., Garcia, J.M., Ruiz-Cortés, A.: Operaciones de Análisis sobre los Términos de Uso en Customer Agreements. In: Proc. XVII Jornadas de Ingeniería de Ciencia e Ingeniería de Servicios. SISTEDES (2022), http://hdl.handle.net/11705/JCIS/2022/041
21. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2019), http://arxiv.org/abs/1908.10084
22. Tunstall, L., Reimers, N., Jo, U.E.S., Bates, L., Korat, D., Wasserblat, M., Pereg, O.: Efficient few-shot learning without prompts (2022)

This work is licensed under a Creative Commons “Attribution-NonCommercial-NoDerivatives 4.0 International” license.