A machine learning-based approach to identify unlawful practices in online terms of service: analysis, implementation and evaluation

Alfonso Guarino1 · Nicola Lettieri2 · Delfina Malandrino1 · Rocco Zaccagnino1

Received: 22 December 2020 / Accepted: 18 July 2021 / Published online: 28 July 2021 © The Author(s) 2021

Abstract
Terms of Service (ToS) are fundamental factors in the creation of physical as well as online legally relevant relationships. They not only define mutual rights and obligations but also inform users about contract key issues that, in online settings, span from liability limitations to data management and processing conditions. Despite their crucial role, however, ToS are often neglected by users that frequently accept without even reading what they agree upon, representing a critical issue when there exist potentially unfair clauses. To enhance users’ awareness and uphold legal safeguards, we first propose a definition of ToS unfairness based on a novel unfairness measure computed counting the unfair clauses contained in a ToS, and therefore, weighted according to their direct impact on the customers concrete interests. Secondly, we introduce a novel machine learning-based approach to classify ToS clauses, represented by using sentence embedding, in different categories classes and fairness levels. Results of a test involving well-known machine learning models show that Support Vector Machine is able to classify clauses into categories with a F1-score of 86% outperforming state-of-the-art methods, while Random Forest is able to classify clauses into fairness levels with a F1-score of 81%. With the final goal of making terms of service more readable and understandable, we embedded this approach into ToSware, a prototype of a Google Chrome extension. An evaluation study was performed to measure ToSware effectiveness, efficiency, and the overall users’ satisfaction when interacting with it.

Keywords Terms of service · Fairness awareness · Machine learning · Software

1 Introduction

Nowadays, people use computers and mobile devices to do almost everything: to gather and share information, connect on social media, have fun, check online banking, browsing, shopping, and so on. Every app or software installed or website browsed has its own Terms of Service (ToS), i.e., legal agreements governing the relationship between providers and users, establishing mutual rights and obligations. Such contracts bind users by the time they switch on the phone or browse a website on the computer. Despite their relevance, ToS are often neglected. A recent survey exploring the behaviour shown by online users while reading ToS reveals that consumers rarely read the contracts they accept [41].

The problem is that, whatever their content is, ToS are often too long and difficult to read [42]. It has been estimated that reading such policies alone would carry costs in time of over 200 hours/year per Internet user [33]. The cognitive effort needed and the concrete inability of laymen, i.e., users lacking technical and legal skills, of evaluating the fairness level in ToS clauses result in both a general sense of frustration for people and in making a mockery of the “notice and choice” legal regime of online ToS [44]. It is not new that some platforms make use in
their ToS of unfair contractual clauses [36], i.e., "contrary to the requirement of good faith", causing a "significant imbalance in the parties rights and obligations arising under the contract, to the detriment of the consumer". Very often, they disregard not only consumer protection law but also what can be considered the EU’s "acquis", i.e., the set of norms and principles emerging from the body of regulations binding on all EU countries. Informed consent, understood as "freely given, specific, informed and unambiguous agreement expressed through clear statements" represents, in this regard, a guiding principle.

In this perspective, a relevant issue is that public agencies in charge of control concretely lack the resources needed to effectively fight against such unlawful practices. Likewise, users, researchers, and regulators still lack usable and scalable tools to cope with ToS hidden threats. Therefore, a novel solution to increase users awareness about some unfair behaviors and uphold legal safeguards becomes needful [27].

The main contributions of this paper can be summarized as follows.

- We propose a novel definition of ToS unfairness. In support of such a definition, we also define a novel unfairness measure, computed counting the unfair and potentially unfair clauses contained in a ToS, weighted via an ad hoc weighting function which assigns more significance to the clauses that have a direct impact on the customers concrete interests. The suitability of this metric has also been empirical evaluated with domain experts.

- We propose a novel machine learning-based approach to classify clauses in ToS, represented by using sentence embedding, into both categories and fairness classes (a legally determined concept that is much more complex but also growingly relevant in data mining research [23]). Support Vector Machine (for clauses’ categories) and Random Forest (for clauses’ fairness levels) resulted to be the most suitable methods for our specific problem after a comparing phase with other widely adopted classifiers [4, 43]. As a result, we obtained a F1-score of 86% in classifying the clauses into (a predefined set of) categories and up to 81% in classifying them according their level of fairness, i.e., potentially unfair and fair clauses. We remark that this represents an evaluation of the capabilities of widely used machine learning techniques to be used for classifying clauses in ToS. The nature of the problem and available dataset led us to hypothesize that this problem could be faced with basic machine learning techniques, without the use of complex and expensive techniques. Indeed, the obtained results confirmed this initial hypothesis.

- We compared the performance of our approach with state-of-the-art methods; results showed that approach was able to outperform all competitors with regard to for all the analyzed scores, i.e., F1-score, Precision and Recall scores.

- We embedded the proposed approach into ToSware, a prototype of a Google Chrome extension aiming at offer end users with increased knowledge of the categorization of ToS clauses and an increasingly awareness about their unfairness level. The tool has undergone a preliminary evaluation study to assess effectiveness, efficiency and, finally, overall users satisfaction when interacting with it.

The rest of the paper is organized as follows: In Sect. 2, we discuss the main contributions available in literature by highlighting the key differences with our work. In Sect. 3, we present the rationale of our work and basic concepts useful to understand our solution. Section 4 is devoted to explain the problem formulation. Section 5 details the approach adopted to classify ToS clauses according to clauses’ categories and fairness levels. Section 6 introduces ToSware, a new Google Chrome extension to increase users awareness about unfair behaviors hidden inside ToS content. We also discuss here results about an evaluation study aiming at assess effectiveness, efficiency, and overall user satisfaction. Finally, in Sect. 7, we conclude with some final remarks and future directions.

2 Related work

A number of proposals available in the literature have attempted to analyze online legal documents (i.e., Privacy Policies and ToS) to offer protection (including providing awareness [27, 30]) of citizens’ rights. If we exclude those regarding Privacy Policies (agreements required by law to inform users how companies collect and use their personal information), e.g., [5, 18, 20], much work have been done with regards to ToS. Therefore, in this section, we present the only two works available in the literature analyzing unfairness behaviors and ambiguous content of ToS files.

In [36] authors developed a theoretical model to partly automate the process of control of clauses’ fairness in online contracts. This type of automation, deployed into a software (standalone application) called uTerms, would help human lawyers make their work more effective and efficient. The proposed model focused on unfair clauses, only. Moreover, uTerms mainly relies on the use of a
dictionary of human-made rules (manually created rules) constructed starting from 20 contracts (109,000 words).

There are several differences with our work. First, in identifying unfair clauses, uTerms relies on a structure-based identification mechanism. Unfair clauses will be highlighted against a perfect match with rules inside the dictionary. Newly encountered unfair clauses (with no matching words present in the dictionary) will be not triggered and highlighted. Conversely, although we too started from a dataset of manually labeled clauses, we next trained a machine learning method to classify clauses syntactically different (with no common words) from those contained in the training set. In addition, authors in [36] only consider unfair clauses from 5 categories: unilateral change, unilateral termination, liability, choice of law and jurisdiction. In our work, we consider a larger set of categories (see Table 3), by also taking into account potentially unfair clauses and fair clauses. Concerning the implementation, our approach has been deployed into a browser extension, and thus no software installation will be required. Finally, preliminary experiments assessed its efficacy, efficiency and ease of use from the final user perspective.

The most relevant work in this field has been presented in [31]. Here, authors propose a machine learning-based method and a tool, for partially automating the detection of potentially unfair clauses. Specifically, they offer a sentence classification system able to detect full sentences or paragraphs containing potentially unlawful clauses. Various methods to analyze terms and extract features were envisioned, including TF-IDF, Bag Of Words and Set Tree Kernel. Several machine learning models were then compared, such as SVM, ensemble methods, Convolutional Neural Networks. As a result, they found out that an ensemble method considering all the models they compared was able to achieve the higher accuracy F1-score (around 81%), outperforming all competitors. One of the proposed approaches (the most feasible in terms of computation requirements) was implemented and developed as a web app, named Claudette. The user has to paste the text to be analyzed and the system will produce an output file that highlights the sentences predicted to contain a potentially unfair clause with also information about the predicted category, among eight pre-defined different categories, this potentially unfair clause belongs to.

With regard to this work, we have both common points and differences over various aspects. First of all, the first common point is about the dataset used to train the selected machine learning-based methods. Indeed, we exploited (and extended) their annotated corpus of 50 contracts as they made it available to the community for further research on this topic. This corpus contains contracts selected among some major players in terms of users and global relevance. The second common point is about the categorization of clauses into the eight categories they identified in their work, obtained in turn by extending the categorization presented in [32]. In our work, we further extended this categorization with an extra neutral category, to also take into account clauses which do not represent an issue for the consumer’s rights.

Concerning the differences, these are mainly about the addressed problem and the provide solution. While authors in [31] faced the problem of identifying potentially unfair clauses, by categorizing them into eight categories, we were interested in identifying belonging categories as well as fairness levels. To the best of our knowledge, this is the first work that attempts to classify ToS clauses under both points of views. To the aim of identifying categories and all types of clauses, i.e., fair, potentially unfair, and unfair clauses, we compared several machine learning methods, deriving that the one based on sentence embedding and SVM is able to achieve the higher F1-score (87%), outperforming all competitors in [31]. A detailed description is presented in Sect. 5. Finally, we developed and implement our approach into ToSware, a Google Chrome browser extension, letting the user to stay in the same browsing context while analyzing the ToS and to use, to get information about a specific clause, a familiar system which to interact with. Our solution offers a “Low cost deployment”, allowing users to adopt the system without facing neither installations problems nor difficult configuration of complex systems. To make this last point clearer, we suppose a user has just accessed to website A. The user wants to analyze the ToS displayed. With the systems at [31, 36] he/she has to open a software/new Web page B (and in case type the URL), then copy and paste the ToS into the text area changing his/her context, a task that may result cumbersome. Instead, with ToSware, the user has only to copy and paste the ToS he/she is interested in into the text area of the popup page or, even easier, just use the context menu on right-click, visualizing the clauses classified/explained with the corresponding highlights. A final difference is that ToSware has been tested to assess efficiency (in terms of system performance), efficacy (in terms of identified unfair behaviors) and usability (in terms of easiness of use and general user satisfaction). A detailed description will be provided in Sect. 6.

Finally, in Table 1, we summarize the comparison of our work against the two relevant works just presented, with regard to the following key factors: (i) objective, (ii) methodology, (iii) number of clauses categories, (iv) dataset size, (v) user evaluation. The last column highlights the main results of each work. In addition, a tree diagram shown in Fig. 1 highlights a further comparison of the works in terms of techniques used to define the approaches for computational protection of rights online.

Neural Computing and Applications (2021) 33:17569–17587 17571
3 Background

In this section, we first present the rationale behind our work and then we briefly introduce word and sentence embedding techniques which we used to represent clauses in ToS.

3.1 The big lie of online ToS

To date, the definition of practical strategies to uphold legal safeguards in digital settings is a though issue. In a

---

Table 1 Summary comparison of the existing works for analyzing online ToS

| References | Objective | Methodology | # categ. | # ToS | User eval. | Main remarks |
|------------|-----------|-------------|----------|-------|------------|--------------|
| [36]       | Development of a theoretical model to automate the process of control ToS clauses fairness (English language) | Clauses have been represented with word-based features and then are feeded into a rule-based software where the rules are handcrafted | 5 | 20 US+UK | × | (i) First work in the area; (ii) introducing a set of handcrafted rules for mining the fairness of ToS clauses; (iii) development of uTerms, a standalone software for annotating unfair clauses; (iv) the work only focused on unfair clauses. Best performance: 82% precision, 100% recall |
| [31]       | Development of system to detect potentially unfair clauses in (English language) ToS and categorize them into predefined classes | Clauses have been represented as vectors relying on the keywords within by using TF-IDF and BoW methods. The system then works with a ML model based on SVM feeded with such vectors | 8 | 50 US+UK | × | (i) Comparison of different approaches (ML and text mining) to the issue; (ii) development of Claudette, a Web app implementing the most feasible approach among those studied; (iii) their best approach is not feasible for real-time applications since it requires heavy computation and long times to process a single clause; (iv) no comparison with other works available. Best performance: 80% F1 score |
| ToSware    | Development of a tool providing awareness to users with regard to online ToS (written in several languages) | Clauses have been represented with Google multilingual Universal Sentence Encoder and then are feeded into a SVM and RF classifiers for recognizing their category and the fairness level, respectively | 9 | 50 US+UK (+10 ITA) | ✔️ | (i) proposal of the ToS unfairness metric and its evaluation; (ii) our system classifies both the fairness of the ToS clause (on a three-level scale) and its belonging category; (iii) development of ToSware, a Chrome extension making use of the designed classifiers; (iv) comparison of the proposal with prior works. Best performance: 86% F1 score (category classification) and 81% F1 score (level of fairness classification) |

# categ. = number of categories to classify clauses; # ToS = number of ToS files in the dataset; User eval. = user evaluation of the proposed system

---

Fig. 1 Comparison of the available works in literature analyzing online ToS in terms of proposed approaches
"hybrid" reality in which technologies and social activities melt, new challenges are raised to legal systems as traditional regulations often look unsuitable to safeguard rights.

The 'notice and choice' paradigm, i.e., the regime based on a presentation of terms followed by an action signifying acceptance of the terms themselves (typically a click on an "I agree" button, or simply the use of the website) often fails in providing users with adequate safeguards. Despite its flaws, notice and choice still represent the cornerstone in regulating users’ interaction with online services (e.g., [22]). This circumstance keeps feeding what has been defined the "biggest lie on the Internet" [41], the lie told by users stating "I agree to these terms and conditions" while, as shown in a number of surveys and reports, ToS and privacy policies are overwhelming, often ambiguous and hard to follow and understand to the few consumers who venture to read them [42] and then, basically unknown.

On top of that, there is a high chance that what a person agreed upon, without reading, includes unfair clauses concerning privacy and beyond [31, 32, 36]. According to art. 3 of the Directive 93/13 on Unfair Terms in Consumer Contracts, a contractual term is unfair if: (a) it has not been individually negotiated; and (b) contrary to the requirement of good faith, it causes a significant imbalance in the parties’ rights and obligations, to the detriment of the consumer. In [32], authors identified five categories of potentially unfair clauses appearing in ToS: (i) establishing jurisdiction for disputes in a country different than consumer’s residence; (ii) choice of a foreign law governing the contract; (iii) limitation of liability; (iv) the provider’s right to unilaterally terminate the contract/access to the service; and (v) the provider’s right to unilaterally modify the contract/the service. In [31], such a taxonomy has been extended introducing: (vi) requiring a consumer to undertake arbitration before the court proceedings can commence; (vii) the provider retaining the right to unilaterally remove consumer content from the service, including in-app purchases; (viii) having a consumer accept the agreement simply by using the service, not only without reading it, but even without having to click on "I agree/I accept". In our work, we further extended such a taxonomy to include in the analysis fair clauses, that is clauses that do not represent an issue for the consumer’s rights.

It is clear that facing the problem is not trivial; the legal mechanism alone for enforcing the prohibition of unfair clauses have often failed to effectively counter this practice so far. However, several studies available in the literature envisioned the possibility for users to benefit from awareness-enhancing mechanisms that help them deal with ToS and privacy policies. In this paper, we dwell on the latter to raise user awareness about the understanding of ToS clauses.

### 3.2 Word and sentence embedding

In the Natural Language Processing (NLP) field, there have been introduced several techniques to understand the meaning of words or sentences for purposes ranging from question answering [49] to sentiment analysis [24], from health [35] to bioinformatics studies [3]. In the last years, word embedding has established itself as one of the most popular representation methods of document vocabulary [10, 45]. Among its capabilities we can cite that of capturing the context of a word in a document, semantic and syntactic similarity, relation with other words, and so on. Word2vec [37, 38] is the most popular technique in this field. It uses the conditional probability $P(w|c)$ to predict the target word $w$ based on its context $c$. They have been used for a variety of tasks, e.g., finance-relating text mining [47], question answering [51], biological sequences mining [2], and so on.

The success of neural networks-based methods for computing word embeddings has motivated the proposal of several methods for generating semantic embeddings of longer pieces of text, such as sentences, phrases or short paragraphs [1]. They are methods to embed a full sentence into a n-dimensional vector space. These sentence embeddings retain some properties, as they inherit features from their underlying word embeddings (e.g., semantic similarity).

Google has developed its own sentence embedding method, named Universal Sentence Encoder, which is capable of dealing with a large number of tasks in NLP [7]. Firstly developed for the English language, it has been subsequently developed for other languages including Italian, German, and Spanish [50]; it is available for developers and researchers through Tensorflow Hub3. This multilingual version specifically embeds text from 16 languages into a single semantic space using a multi-task dual-encoder model that learns tied representations using, in turn, bridging translation tasks, to encourage sentences with identical meanings, but written in different languages, to have similar embeddings [8]. Experiments showed that the Universal Sentence Encoder show good performance with minimal amounts of supervised training data [7]; it takes in input a variable-length text, and the output is a 512-dimensional vector. In this paper, we used the multilingual Universal Sentence Encoder (hereinafter, mUSE) to represent clauses in ToS.

3 https://bit.ly/36BSSS2.
4 ToS unfairness: definition and measurement

In this section, we propose a novel definition of ToS unfairness. In support of such a definition, we also define a novel unfairness measure, computed by taking into account all clauses contained in a ToS.

Informally, a ToS consists of a set clauses, where each clause can belong to one of the categories of clauses introduced in Sect. 3. This categorization has been introduced in [32], extended in [31] and further extended in our work to consider clauses that do not imply unlawful behaviors, and that therefore, represent fair clauses for the consumer (see also Table 3). It is also possible to assign to each category a significance level (weight) that expresses how and/or where the potential suffered harm should be disputed or which laws govern the contract. The re-organization of categories according to the defined weights is shown in Table 2.

To measure the unfairness of a ToS, we first classify each sentence in three possible fairness levels, i.e., fair, potentially unfair or unfair, and then we compute a quasi-weighted sum of the unfair and potentially unfair clauses within the ToS.

Formally, let ToS = \{s_1, s_2, \ldots, s_n\} be a ToS, where s_i is a clause in the contract, for i = 1, \ldots, n. Let C = \{c_1, c_2, \ldots, c_m\} be the set of clauses categories, F = \{f_1, f_2, \ldots, f_k\} be the set of fairness levels, and W = \{w_1, w_2, \ldots, w_n\} be the set of category weights. We define a weight function w : C \rightarrow W which associate to each c_i \in C a weight in W. Thus, we indicate with w(c_i) the weight of c_i. Now, given s_i \in ToS, we indicate with c(s_i) the category of s_i, f(s_i) the fairness level of s_i, and w(s_i) the weight of s_i where w(s_i) = w(c(s_i)). We remark that c(s_i) \in C and f(s_i) \in F.

Table 2 Weights (W) for each of the clauses categories (C) assigned by domain experts in the field of Law, Privacy and Consumer Rights

| W (weights) | C (categories) |
|-------------|---------------|
| 0           | neu           |
| 1           | a, i, law     |
| 2           | ch, cr, ltd, ter, use |

A detailed description of the nine categories is illustrated in Table 3.

In this work, we set C = \{a, ch, cr, j, law, ltd, neu, ter, use\}, where a, ch, cr, j, law, ltd, ter, use, described in Table 3, have been defined in [31], and neu has been introduced here (more details in Sect. 5.1.1). We set F = \{ff, pu, uf\}, where ff = fair, pu = potentially unfair, and uf = unfair, as proposed by [31].

Thus, we set W = \{0, 1, 2\} and the function w assigns weights to clauses according to the Table 2.

Then, given a Term of Service ToS = \{s_1, \ldots, s_n\}, the overall unfairness of ToS is computed as follows:

$$ uf(ToS) = \frac{1}{n} \sum_{s_i \in ToS} w(s_i) $$

A comprehensive representation of the problem is available in Fig. 2.

To summarize, given a ToS with a certain number of clauses to analyze, the basic idea is to: (a) classify these clauses according to their belonging category, (b) classify these clauses according to their fairness level, (c) assign a weight to each clause (see Table 2) using the result of the first classification, and finally, (d) compute the overall ToS unfairness (i.e., uf(ToS)). In Sect. 5, we will describe the methodology followed to define machine learning methods able to perform category classification and fairness level classification.

Empirical evaluation. To assess the feasibility of the proposed metric, given a dataset of ToS, we verified whether a correlation existed between results of the unfairness measure when applied on this dataset, and the results obtained when the same process was applied by human annotators, i.e., our domain experts (that thus provided their personal opinion about the unfairness of the ToS). Specifically, we firstly asked them to provide a score s \in [1..10] for each ToS in our dataset to express its likelihood to be unfair, so that ToS_i with s = 10 is likely to be more unfair than ToS_j with s = 5. We have also checked the inter-annotator agreement through the Fleiss’ Kappa metric [34] (the adaptation of Choen’s Kappa for more than 2 annotators). The kappa value was 0.746, resulting thus in a “substantial agreement”. Next, we computed the unfairness measure of every ToS in our dataset. The last step involved the application of a Pearson correlation [6] between results of the unfairness measure and the average scores provided by human annotators. We found out that results were strongly correlated $r(48) = 0.9247, p < .00001$, suggesting that even though our metric in its definition is quite simple, its effectiveness is on par with what one would get with domain experts. The scores and unfairness measure gathered/computed are available at https://bit.ly/2IWygZ4.
5 A machine learning based method to classify ToS clauses

In this section, we describe a novel machine learning-based method to classify ToS clauses according to pre-defined categories and a novel machine learning-based method to classify ToS clauses according to three fairness levels. We remark that to the best of our knowledge, this is the first work in which a machine learning-based method to classify ToS clauses according fairness level was proposed. To pursue this goal we defined a methodology encompassing four steps (see Fig. 4):

- **Dataset Building**: In this step, we downloaded a set of XML formatted resources, representing the ToS files containing the clauses, labeled by authors in [31]; we updated the labeling relying on the experience of 5 domain experts and we represented ToS clauses with a sentence embedding method.
- **Validation**: The dataset has been split into training and testing sets; k-fold cross-validation was performed to validate different classifiers;
- **Testing**: The most used classifiers in the literature have been tested on the testing set with the best parameters found during the previous step; we further tested our method against a different kind of dataset that contains non-ToS contracts, made available recently for the research community;
- **Comparison with state-of-the-art methods**: The most effective method, as result of the experiments did in the testing phase, has been compared with some
competitors previously tested for the problem under investigation.

All experiments have been conducted with a 2,8 GHz Intel Core i7 quad-core machine equipped with 16 GB 2133 MHz LPDDR3 RAM.

5.1 Dataset building

The starting point of the defined methodology is the selection of ToS of online services and platforms. For this step, we used ToS of popular online services made publicly available in [31]. The corpus consists of 50 online contracts, selected among those offered by some of the major players (e.g., Google, Snapchat, Spotify, Facebook, Uber, Deliveroo, Dropbox, Rovio, WhatsApp, TripAdvisor, Booking) in terms of different characteristics, such as, number of users, global relevance, and time of establishment of the service. Such files have been released in XML format and have been labeled by domain experts according to a categorization of the contained clauses, in the eight pre-defined categories as we described in Sect. 3.

5.1.1 Dataset labeling

The annotated dataset downloaded in the first phase does not take into consideration clauses extraneous to the taxonomy defined by their authors, that is, not risky clauses for consumers; as a result, these clauses were forced to fall in one of the envisioned categories despite no relevance with the target category existed (probably considering risky something that was not). For this reason, we involved five domain experts to manually annotate sentences which are "neutral", thus adding a \texttt{neu} \texttt{tag} to our taxonomy. As anticipated before, the involved domain experts had a consolidated experience legal, privacy and consumer rights fields.

To better explain our procedure, shown in Fig. 3, we provide a clarifying example analyzing in detail the Spotify ToS. Specifically, starting from the version tagged according to the downloaded taxonomy (Fig. 3a), we used the \texttt{neu} \texttt{tag} for all clauses that do not represent a risk (Fig. 3b). We tagged as, as an example, \texttt{neu} the sentence declaring that: "The Spotify Service includes social and interactive features. Use of the Spotify Service relies on several technical requirements. Your agreement with us includes these Terms and Conditions of Use ("Terms") and our Privacy Policy. (The Terms, Privacy Policy, and any additional terms that you agree to, as discussed in the Entire Agreement section, are referred to together as the "Agreements")."

Fig. 3 Example of labeling performed by domain experts on the Spotify ToS. a The downloaded ToS file, b the labeling with the tag <\texttt{neu}>, c) the labeling with the fairness level
sentences, obviously all clauses fair and not risky for consumers. Finally, we tagged each clause with a degree of fairness across three possible tags, and that is, < 3> for unfair clauses, < 2> for potentially unfair clauses and < 1> for fair clauses (Fig. 3c).

In summary, we built two datasets, the first one named Tags, with clauses split by categories, and the second one named Fairness, with clauses split by fairness levels, whose labeling information are summarized in Table 3. The final goal was, from one hand, to classify ToS clauses in $8^2 + 1$ classes according to the categories ($< a >$, $< ch >$, $< cr >$, $< j >$, $< law >$, $< ltd >$, $< ter >$, $< use >$, and finally the aforementioned $< neu >$ type), and on the other hand, to classify ToS clauses in 3 classes according to adequate fairness level (i.e., fair, potentially unfair, unfair).

### 5.1.2 Clauses representation

It is important to highlight that during the last years we have witnessed a flourishing of online tools enabling digital services’ owners to generate, in a handful of clicks, their ToS. Therefore, we can argue that is plausible a significant part of clauses can be very similar in ToS across a wide range of online services. As a consequence, the embedding of such clauses can lead to a high similarity between the n-dimensional vectors, respectively. For this reason, in order to extract clauses from the raw data borrowed from [31], we employed a Python XML parser based on the ElementTree XML library. Once extracted the sentences/paragraphs, as anticipated in Sect. 3.2, we exploited a sentence embedding method, and in particular the Google multilingual Universal Sentence Encoder (mUSE) [50], to obtain one 512-dimensional vector for each extracted clause. Such embedded vectors represent the features that we used for the chosen classifiers.

The assumption that the embedding of clauses in the same category can lead to very similar n-dimensional vectors can be visually verified in Fig. 5. Specifically, for each analyzed category we have randomly chosen 8 clauses, for each clause we have calculated the 512-dimensional vector, and finally, for each pair of such vectors we calculated the similarity as their inner product. Similarities value $s$ ranges in [0, 1], where 0 means very different clauses and 0 identical clauses. As we can see, the overall similarity is above 0.4, with several areas in which the value is above 0.6.

### 5.2 Validation

The dataset built was split, with a stratified approach, into: (1) training set, obtained by including the 80% of the elements (randomly chosen), and (2) testing set, obtained by including the remaining 20% of the elements.

We compared the most popular machine learning methods available in the literature by implementing them using the scikit-learn Python library, and specifically, Random Forest (RF), Support Vector Machines (SVM), MultiLayer Perceptron (MLP), K-Nearest Neighbors (KNN), AdaBoost (Ada).

Finally, to validate the machine learning methods, we performed a 10-fold cross-validation by using the GridSearchCV method, as also did, as an example, in [11, 39]. In this phase, we tried to optimize the F1-score due to the

---

4 e.g, https://www.termsandconditionsgenerator.com.
slightly unbalancing in our dataset, by evaluating for each method the following hyper-parameters.

- **RF**: Its performance mainly relies on the number of estimators, and therefore we proceeded with trials in the range between 100 and 1000. Best results were found between 300 and 500 estimators.
- **SVM**: We performed trials on different kernels (polynomial, sigmoid, radial) and we made optimizations with regard to the penalty parameter C (from 0.001 to 1000). Best results were found with the radial kernel and C values from 1 to 100.
- **MLP**: It mainly relies on the hidden layers size. The number of hidden layers size was tested with values in the range between 50 and 500 (ten by ten). Best results were found between $\frac{2}{3}$ and $\frac{4}{5}$ of the input layer size. We also adopted the lbfgs optimizer, that has been proved to converge faster and perform better on small datasets [52].
- **KNN**: We tested the radius r with values in the range between 1 and 5 and the weights parameter with uniform and distance values. Best results have been found with $r = 4$ and uniform weights.
- **Ada**: It has been tested on all the available loss function in the Scikit-learn library, the number of estimators with values in the range from 100 to 1000, and the learning rate with values from 0.001 to 1. Best results were found with exponential and square loss functions, learning rate between 0.1 and 1, estimators between 400 and 600.

Finally, the selected classifiers were trained with the Tags and the Fairness datasets.

### 5.3 Testing

At the end of the Validation step, we obtained the best parameters to train and test our classifiers, on the testing set. All results about both the Tags and the Fairness datasets are shown in Table 4. Specifically, with regard to the Tags dataset, the better F1-score performance, i.e., 86%, is achieved by SVM, whereas for the Fairness dataset, the classifier with higher performance is RF, with a F1-score of 81%. SVM and RF achieved the higher values also for Precision and Recall scores. Therefore, the outcome of this analysis is the choice to implement the classifier to distinguish the tags by using SVM and the classifier to distinguish the fairness levels by using RF. In the Scikit-learn library, for the RF classifier, there is also the possibility of implementing the classes probabilities to have as output the probabilities of labels/classes instead of the labels/classes themselves. In a real usage scenario, therefore, we can rely on the clause’s probability in being fair, potentially unfair or unfair (e.g., instead of having fair/unfair as labels of clauses, we will have something like “0.70 probability a given clause is unfair”).

### 5.4 Testing “into the wild”

Very recently, researchers of "The Atticus Project"\(^5\) have made available the Contract Understanding Atticus Dataset (CUAD) v1, a corpus of more than 13,000 labels in 510 commercial legal contracts with rich expert annotations created to reduce the societal costs of contract review. The dataset has been manually labeled (with a year-long effort pushed forward by dozen of law student annotators,  

---

\(^5\) https://www.atticusprojectai.org/.
of lawyers, and machine learning researchers) to identify 41 types of legal clauses in commercial contracts that are considered important in contract review in connection with a corporate transaction, including mergers & acquisitions, corporate finance, investments and IPOs [21]. Thus, we used this dataset as further testbed to evaluate the effectiveness of the Tags classifier (SVM and mUSE), in the classification of ToS clauses’ categories.

Since the CUADv1 dataset contains a larger number of legal clauses (41 against 9) for a different type of contracts (commercial contracts against ToS), we asked to one of our domain experts to assess whether there existed a mapping between the different types of clauses. Our expert identified four types of common clauses, although they were indicated with a different nomenclature. Specifically, the mapping was the following: “Change of Control” → <ch>, “Governing Law” → <law>, “Cap on Liability” → <ltd>, “Termination for Convenience” → <ter>. The expert, thus, concluded his analysis with the following clauses: <ch>, <law>, <ltd>, and <ter> that could be further analyzed.

We report the results obtained in this phase in Table 5. We observe that, although on different kind of legal contracts, the proposed method exhibits encouraging performance with F1 scores ranging from 0.76 to 0.82. The motivations behind some errors are the following: There are some clauses belonging to <law> have been wrongly classified as <j>, since in many cases they refer to specific countries like <j> clauses do in our original dataset. In other cases, <ter> clauses have been wrongly classified as <ltd>, as an example for the following clause: “Company at its sole discretion may at any time alter or cease providing the Customer Service which it has agreed to provide to Client relating to Client Website pursuant to this Agreement without any liability to Company.”

### 5.5 Comparison with state-of-the-art methods

The encouraging results obtained in the Testing step led us to proceed with the comparison of our approach with relevant works proposed in the literature. It is worth to note that, with regard to the category classification task, as discussed in Sect. 2, we compared our method with those presented in [31] and [36], i.e., the only relevant works presenting a similar approach for classifying unfair contractual terms. Instead, the Fairness-level classification method proposed here is the first attempt to classify clauses in three possible classes, i.e., fair, unfair and potentially unfair. Indeed, in [31] the authors faced a different problem, i.e., firstly how predicting whether a given sentence contains a (potentially) unfair clause and then how predicting the category to which this specific clause belongs to. Notwithstanding the lack of related works to which refer to for the efficacy of our fairness level classification method, we provide an in-depth analysis of our method’s capability when addressing this task.

Results about the comparison for category classification are shown in Table 6, and as we can see, our method, shown in the table as “SVM+mUSE”, shows a better F1-score against all others analyzed methods. The competitors represent: (a) The classifiers that authors in [31] used in their analysis, where the best performing model resulted to be the C8 classifier, an Ensemble method of C1, C2, C3, C6 and C7; (b) the rule-based method used in [36] on the five categories identified in [32], that is, (<ch>, <j>, <law>, <ltd>, <ter>). To perform this last comparison, we had to tweak our SVM+mUSE approach and reduce the number of categories to only five, and thus re-fitting it. We refer the reader to [31, 36] for a detailed description of the
methods and their combinations. In summary, this experiment showed that our approach can classify categories with F1-scores of 0.87 when analyzing 9 categories (comparison with the method proposed in [31]), and 0.91 when analyzing 5 categories (comparison with the method proposed in [36]).

Moreover, notice that the motivations behind the outperforming performance of our method could lie in the use of sentence embedding features from a powerful pre-trained neural network, while state-of-the-art works used word-level features such as bag of words (see Fig. 1).

We can now proceed with the analysis of the ability of our method to detect the clauses categories for a given fairness level. We will describe the results for each type of fairness level in turn.

Unfair clauses. In Fig. 6, we show the results about the capability of our method in detecting "unfair" clauses. Specifically, we can see results about only the Content removal, Termination, Jurisdiction and Limitations of liabilities categories, as for the specific analyzed dataset, the unfair clauses occur in such categories, only. Our method shows a very high F1-score, especially for Jurisdiction and Limitations of liabilities categories. The lower value for Content removal clauses can be due to the heterogeneity or less similarity between the clauses in this category. As an example if we consider: "Rovio may manage, regulate, control, modify or eliminate virtual goods at any time, with or without notice" (Rovio ToS) and "You also agree that Spotify may also reclaim your username for any reason" (Spotify ToS), their similarity is low, notwithstanding they are both Content removal type clauses.

Potentially unfair clauses. Concerning potentially unfair clauses, our method has comparable performance with the best method available in the literature (i.e., C8 in Table 6, an ensemble of different SVM classifiers). Indeed, except for Content removal, in which we obtained a higher precision but poor recall, all differences are negligible (see Fig. 7). Similarly to the previous case, this result is due to the heterogeneity of clauses within this category (see also Fig. 5 where indeed Content removal clauses show slightly lower similarities).

Fair clauses. In Fig. 8, we show results about the capability of our method in detecting fair clauses. Similarly to the analysis for discovering unfair clauses, we show here three categories as fair clauses are available only here. In addition, the best performance is obtained when dealing with Jurisdiction and Limitations of liabilities categories.

6 ToSware: a prototype tool for terms of service aWAreness

Existing systems trying to make ToS easier to understand have been implemented so far as a standalone application or Web services. To foster the usage from any user (also without technical skills), and to guarantee low cost deployment (to avoid the burden of installing specific software and configure complex systems), and to reduce the cumbersome process of select, copy, go to a new Web page, paste, we embedded our approach in a Chrome browser extension so that the user can continue to use something familiar system (that is his/her browser), without changing context while browsing.

ToSware is a prototype extension aiming at support individuals in evaluating ToS and better understand the (un)fairness of their clauses, just having a look at some provided information. Specifically, it provides visual aids in the form of highlighted parts, icons and probability percentages, allowing the user to assess the category and the "fairness-critical" level of information contained in the ToS. Awareness about the category of clauses contained in a ToS is guaranteed through the use of suitable and intuitive icons (see Fig. 10b). Awareness about the "fairness-critical" level of information, to inform about the presence

| Work          | Method                           | P   | R   | F1  |
|---------------|----------------------------------|-----|-----|-----|
| [31] (C1)     | SVM-single model (sm)            | 0.73| 0.83| 0.77|
| [31] (C2)     | SVM-combined model (cm)          | 0.80| 0.78| 0.78|
| [31] (C3)     | Tree Kernels                     | 0.78| 0.72| 0.74|
| [31] (C4)     | Convolutional Neural Network     | 0.73| 0.74| 0.72|
| [31] (C5)     | Long Short-Term Memory network   | 0.70| 0.72| 0.70|
| [31] (C6)     | SVM-Hidden Markov Models sm      | 0.76| 0.78| 0.76|
| [31] (C7)     | SVM-Hidden Markov Models cm      | 0.86| 0.69| 0.76|
| [31] (C8)     | Ensemble of (C1, C2, C3, C6, C7) | 0.83| 0.80| 0.81|
| **This work** | SVM + mUSE                       | 0.86| 0.86|** 0.87**|
| [36]          | Rule-based (five categories [32])| 0.82| 1.00| 0.88|
| **This work** | SVM + mUSE (five categories [32])| 0.90| 0.92|** 0.91**|

Bold values indicate the classifiers (and the corresponding f1-score achieved) which exhibit the best results.
of unfair, potentially unfair and fair clauses, contained in a TOS file, is guaranteed through a chromatic categorization, i.e., by using red, yellow and green colors to suggest unfair, potentially unfair and fair behaviors, respectively. Finally, the probabilities inform users about the extent to which a clause can be said to have a certain level of fairness. The overall objective is to guarantee a friendly (ease-to-use interface) and effective (response-time efficiency) user experience (see Figs. 11 and 12).

The browser extension prototype we designed and developed is composed, as shown in Fig. 9, by a client-side component and a server-side component that interact and exchange data through a request/response mechanism; we will describe these components in detail in the following, starting from a typical use case scenario. Specifically, we will first describe the ToSware's architecture with the functionalities provided by its components and then the evaluation study we performed to assess efficiency, in terms of system performance, effectiveness, in terms of correct classification on a newly introduced dataset, and finally, usability, in terms of overall user satisfaction.

### 6.1 ToSware implementation

To better explain the functionalities provided by ToSware we describe here the typical scenario in which a user can be involved in, by giving subsequently details about the architecture and its main components.

#### 6.1.1 Use case scenario

While browsing the Spotify’s Terms of Service, the end user experiences troubles in understanding its content, and therefore he/she wishes an explanation. The steps to follow are described in the following.

---

6 Available at https://bit.ly/2IWxgZ4.
7 https://www.spotify.com/it/legal/end-user-agreement/.
1. The user inputs the ToS file or part thereof into ToSware UI (a simple inputting content action in a form, shown in Fig. 10a).

2. The text is received and elaborated by the server-side component; the ToS content is analyzed and results, that is, belonging category and fairness levels, will be returned to the client.

3. The client-side component will show the ToS clauses according to visual metaphors [48]. In particular, we make use of customized awareness icons for each category (see Fig. 10b) and a simple badge for their fairness level (see Figs. 11 and 12).

6.1.2 ToSware components

As shown in the rightmost part of Fig. 9, the server-side component in ToSware has been developed exploiting Django, a high-level Python Web framework\(^8\). When a ToS content is sent to the server for the analysis, ToSware handles it through several steps: (a) detecting of the language with TextBlob\(^9\), (b) splitting the text with a well-known technique developed in [25], (c) embedding the texts into vectors via mUSE, (d) analyzing texts embedding via the Tags Classifier and the Fairness Classifier. Texts and related labels are returned to the ToSware client-side component in JSON format. We want to emphasize that clauses classified as neutral are not returned to the client to reduce the number of response packets.

As shown in the leftmost part of Fig. 9, the client-side component allows users to input ToS content to an easy-to-use user interface implemented by using technologies such as JavaScript and Bootstrap, a popular front-end open source toolkit to quickly design and customize responsive mobile-first sites. The user has to input the ToS or part thereof into the provided form and then click on the “Analyze” button (see Fig. 11). As default option, we do not show clauses classified as fair to reduce the visual clutter; an extra action (“Show all clauses”) will instead reveal them (see Fig. 12).

After the analysis phase performed server-side, the clauses are returned to the client-side component and are shown to the user marked with the customized awareness icons (Fig. 10b). To ease the understanding of this visualization, we set up a complementary Web page (reachable from ToSware) summarizing the taxonomy used, and the visualization meaning in the form of a table. For each identified clause, the client also shows the fairness level through a chromatic categorization and a probability percentage that informs users about the extent to which that clause can be said to have the identified level of fairness.

6.2 ToSware evaluation

In this section, we first describe the results of the browser system performance when ToSware loads ToS content (a

---

\(^8\) [https://www.djangoproject.com/].

\(^9\) [https://textblob.readthedocs.io/en/dev/].
new dataset of Italian ToS) and analyzes it to classify clauses and then we describe the results of the evaluation study we performed to assess the user satisfaction about the tool and its functionalities.

Browser performance. For this experiment, we instrumented Selenium WebDriver to perform two tasks: (1) copy-pasting into ToSware of 111 terms coming from 10 new Italian ToS10, (2) click on the “Analyze” button. These terms were previously annotated by 5 domain experts in the legal field. Meanwhile, via the psutil Python library11 we monitored system resources usage (CPU and memory). Firstly, results showed that the time required to analyze each clause and classify it was lower than 2 seconds (the processing of 80% of requests lasted 1.3 seconds). For more 70% of the time under experimentation the RAM usage, the client-side, was under 4Mb while server-side the average usage was about 10Mb. Since the whole computation is performed server-side, the CPU usage on the client is negligible, while server-side, we had, however, positive results with 60% of requests that used less than 10% of the CPU.

Effectiveness. With regard to the effectiveness of our approach on this new “previously unseen” dataset of contracts, we obtained performance F1-scores of 83% and 79% when classifying ToS clauses into categories and fairness levels, respectively. In this experiment, scores were slightly lower than those achieved in the Testing step; by analyzing in detail the sentences we found out that the majority of the errors have been made due to the different writing style between English and Italian ToS clauses (Italian clauses tend to be written in a more articulated way).

User evaluation. For this preliminary evaluation study, we followed the standard Human-Computer Interaction (HCI) methodology [26], by envisioning three different phases, as defined and implemented in other contexts [16, 17, 28, 29]. We recruited 25 participants among computer scientist (44%) and individuals from the humanities field (56%). The sample was balanced in gender with a mean age of 33.

Prior research has shown that five users are the minimum number required for usability testing, since they are able to find approximately 80% of usability problems in an interface [46]. However, other research studies stated that five users are not sufficient and specifically, authors in [40] expressed that the appropriate number depends on the size of the project, with 7 users being optimal in small projects and 15 users being optimal in a medium-to-large project.

At a first stage, we asked participants to fill in a preliminary questionnaire asking for demographic information and information about ICT experience and skills. Then, we gave them information about ToSware and of its main functionalities. In the subsequent Testing phase, we asked them to use ToSware for a 10-minute session and then accomplish two tasks: (1) “go to https://alfonsino.delivery and select a single ToS clause to understand” (Task 1) and (2) “go to https://www.calzedonia.com/it/ and select a ToS paragraph to understand” (Task 2). At the end of each task, the users answered to questions to evaluate whether it was successfully completed, rate how easy and quick was to perform the task (standard questions from the after scenario questionnaire12). At the end of this testing phase, we asked users to spend 10 minutes to fill in the standard Questionnaire For User Interaction Satisfaction (QUIS) [9]. Finally, in the third and last phase, the last 5 minutes were required to respond to a summary survey, in which we asked users to rate their perception about: (a) increased

---

10 The selected ToS, both in the original and annotated form, are available here: https://bit.ly/2IWxgZ4.
11 https://psutil.readthedocs.io/en/latest/.

---

Fig. 10 ToSware UI and visual metaphors. a How users can enter ToS content to analyze; b Icons used to make ToS clauses more readable and easy to understand for beginner users

---

12 https://garyperlman.com/quest/quest.cgi?form=ASQ.
understanding of the proposed concepts, (b) increased awareness of the meaning of clauses, (c) the usefulness of the proposed instrument and finally, (d) their behavioral intention to use ToSware in future.

Results of the evaluation study showed, firstly, that all participants rated (on a 5-point Likert scale) as very easy to perform the assigned tasks (M = 4.0 for both tasks). As depicted in Fig. 14, the analysis of the software usability through the QUIS questionnaire showed that, on average, all posed questions were rated very positively, confirming that participants were highly satisfied with the software proposed.

Finally, also questions posed in the Summary survey questionnaire, and shown in Fig. 13, were all positively rated. Specifically, at the questions: Q1: “Do you understand the meaning of the shown highlights?” and Q2: “Do you understand the meaning of the shown visual metaphors (icons)?”, most of participants provided a positive response, with only 8% of participants that not understood the use of icons (M = 4.5, SD = 0.6 for Q1 and M = 3.9, SD = 1.4 for Q2).

Moreover, 72% of participants stated that ToSware was able to facilitate the understanding of critical clauses (M = 3.9, SD = 0.7 for Q3). Finally, 88% rated ToSware a useful tool.

Fig. 11 ToSware front-end: results shown to the user asking information. Default: only potentially unfair and unfair clauses are shown to the user.

Fig. 12 ToSware front-end: results shown to the user asking information. All clauses are shown to the user. Additionally, a tooltip can provide further explanations.
instruments (M = 4.3, SD = 0.7 for Q4), while 80% of participants will continue to use ToSware in future (M = 4.4, SD = 0.8 for Q5).

7 Conclusion

The "notice and choice" legal regime used to rule the agreement on online ToS has shown severe flaws. Due to various reasons (intricacies in the texts, lack of legal skills), users struggle to grasp all the implications of clauses they are agreeing upon and often end up skipping reading them. This allows companies to take advantage of inscrutable and unfair contractual clauses that limit their liabilities or allow them to arbitrarily interrupt services at any time.

To tackle this issue, we proposed a novel machine learning-based approach whose main goal is to make ToS more readable/understandable in order to increase user awareness and "technologically enhance" consumer rights protection. Our approach exploited SVM for the clauses category classification task (F1-score of 86%) and RF for the fairness level categorization task of such clauses (F1-score of 81%).

With regard to the results of the experiments about the comparison with state-of-the-art works, although showing slight performance differences, our approach is able to reach the highest F1-score.

The approach has been embedded in ToSware, a prototype of a Google Chrome extension, which has been evaluated to analyze the impact on the user experience. The prototype's code is available online\textsuperscript{13}.

As future works, we are currently working toward three directions. First, we will define ad hoc models for Categories and Fairness level classifications, and we will also enlarge the dataset of annotated ToS in order to perform further experiments with ToS in several languages. The second direction is about the employment of our strategy in the field of privacy policies; the idea is to verify whether machine learning based methods could be efficiently employed to identify ambiguous behaviors in policies governing the privacy of individuals and of their personal data. Finally, further design and development enhancements are planned about ToSware, with the final goal of making it available soon on the Google Chrome web store. Thereafter, an exhaustive user evaluation, comparing different systems, and performance benchmarking [12, 14, 15, 19] will be performed to assess the overall usability and efficiency.

Funding Open access funding provided by Università degli Studi di Salerno within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Sanjeev A et al. (2017) "A Simple but tough-to-beat baseline for sentence embeddings." ICLR
2. Asgari E, Mofrad MR (2015) Continuous distributed representation of biological sequences for deep proteomics and genomics. PloS One 10(11):e141287

\textsuperscript{13} https://github.com/alfonsoguarino/ToSware.
3. Badal VD, Kundrotas PI, Vakser IA (2018) Natural language processing in text mining for structural modeling of protein complexes. BMC Bioinf 19(1):84:1-84:10

4. Bakour K, Unver HM (2021) VisDroid: Android malware classification based on local and global image features, bag of visual words and machine learning techniques. Neural Comput Appl 33:3133–3153. https://doi.org/10.1007/s00521-020-05195-w

5. Bannihiattu Kumar V, Iyengar R, Nisal N, Feng Y, Habib H, Story P, Cherivirala S, Hagan M, Cranor L, Wilson S et al (2020) Finding a choice in a haystack: automatic extraction of opt-out statements from privacy policy text. Proc Web Conf 2020:1943–1954

6. Benesty J, Chen J, Huang Y, Cohen I (2009) Pearson correlation coefficient. In: Noise reduction in speech processing, pp. 1–4. Springer

7. Cer D, Yang Y, Hua N, Limtiaco N, John RS, Constant word-based features. In: proceedings of the 2nd international conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 169–174

8. Chidambaram M, Yang Y, Cer D, Yuan S, Sung YH, Strope B, Kurzweil R (2018) Learning cross-lingual sentence representations via a multi-task dual-encoder model. arXiv preprint arXiv:1810.12836

9. Chin JP, Diehl VA, Norman KL (1988) Development of an instrument measuring user satisfaction of the human-computer interface. In: proceedings of the SIGCHI conference on human factors in computing systems, CHI, pp. 213–218

10. D’Ambrosio R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning, pp. 160–167

11. Cosimato A, De Prisco R, Guarino A, Malandrino D, Lettieri N, Sorrentino G, Zaccagnino R (2019) The conundrum of success in music: playing it or talking about it? IEEE Access 7:123289–123298

12. Cozza F, Guarino A, Isernia F, Malandrino D, Rapuano A, Schiavone R, Zaccagnino R (2020) Hybrid and lightweight detection of third party tracking: design, implementation, and evaluation. Comput Netw 167:106993

13. Cronbach LJ (1951) Coefficient alpha and the internal structure of test. Psychometrika 16(3):297–334

14. D’Ambrosio R, Pasquale SD, Iannone G, Malandrino D, Negro A, Patino G, Scarano V, Spinelli R, Zaccagnino R (2017) Privacy as a proxy for green web browsing: methodology and experimentation. Comput Netw 126:81–99

15. De Prisco R, Guarino A, Lettieri N, Malandrino D, Zaccagnino R (2021) Providing music service in ambient intelligence: experiments with gym users. Expert Syst Appl 177:114951

16. De Prisco R, Malandrino D, Pirozzi D, Zaccagnino G, Zaccagnino R (2017) Understanding the structure of musical compositions: Is visualization an effective approach? Inf Vis 16(2):139–152

17. Erra U, Malandrino D, Pepe L (2019) Virtual reality interfaces for interacting with three-dimensional graphs. Int J Hum Comput Interact 35(1):75–88

18. Fukushima K, Nakamura T, Iked a D, Kiyomoto S (2018) Challenges in classifying privacy policies by machine learning with word-based features. In: proceedings of the 2nd international conference on cryptography, security and privacy, pp. 62–66

19. Grieco R, Malandrino D, Scarano V (2006) A Scalable Cluster-based Infrastructure for Edge-computing Services. World Wide Web 9(3):317–341

20. Hankous H, Fawaz K, Lebret R, Schaub F, Shin KG, Aberer K (2018) Polisis: automated analysis and presentation of privacy policies using deep learning. In: 27th USENIX Security Symposium, pp. 531–548

21. Hendrycks D, Burns C, Chen A, Ball S (2021) CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review. arXiv.org/abs/2103.06268

22. House W (2014) Big data: seizing opportunities, preserving values (report for the president). Washington DC, USA: Executive Office of the President. https://bit.ly/31VESF

23. Jalali ZS, Wang W, Kim M, Raghavan H, Soundarajan S (2020) On the information unfairness of social networks. In: Proceedings of the 2020 SIAM International Conference on Data Mining, pp. 613–521. SIAM

24. Kaur H, Mangat V et al (2017) A survey of sentiment analysis techniques. In: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), pp. 922–227

25. Koehn P, Schroeder J (2007) Experiments in domain adaptation for statistical machine translation. In: Proceedings of the second workshop on statistical machine translation, pp. 224–227

26. Lazar J, Feng JH, Hochheiser H (2010) Research methods in human-computer interaction. Wiley, New York

27. Leenes R (2011) Framing techno-regulation: an exploration of state and non-state regulation by technology. Legisprudence 5(2):143–169

28. Leon PG, Ur B, Shay R, Wang Y, Balebako R, Cranor LF (2012) Why Johnny can’t opt out: a usability evaluation of tools to limit online behavioral advertising. In: J.A. Konstan, E.H. Chi, K. Hsiok (eds.) CHI Conference on Human Factors in Computing Systems, CHI ’12, Austin, TX, USA - May 05 - 10, 2012, pp. 589–598. ACM

29. Lettieri N, Altamura A, Malandrino D (2017) The legal macro-scope: experimenting with visual legal analytics. Inf Vis 16(4):332–345

30. Lettieri N, Guarino A, Malandrino D, Zaccagnino R (2019) Platform economy and techno-regulation-experimenting with reputation and nudge. Future Internet 11(7):163

31. Lippi M, Palka P, Contissa G, Laiorgia F, Micklitz HW, Sartor G, Torroni P (2019) Claudette: an automated detector of potentially unfair clauses in online terms of service. Artif Intell Law 27(2):117–139

32. Loos M, Luzak J (2016) Wanted: a bigger stick. on unfair terms in consumer contracts with online service providers. J Consumer Policy 39(1):63–90

33. McDonald AM, Cranor LF (2008) The cost of reading privacy policies. Isjlp 4:543

34. McHugh ML (2012) Interrater reliability: the kappa statistic. Biochemia Medica 22(3):276–282

35. McTaggart S, Nangle C, Caldwell J, Alvarez-Madrazo S, Colhoun H, Bennie M (2018) Use of text-mining methods to improve efficiency in the calculation of drug exposure to support pharmacopidemiology studies. Int J Epidemiol 47(2):617–624

36. Micklitz HW, Palka P, Panagis Y (2017) The empire strikes back: digital control of unfair terms of online services. J Consumer Policy 40(3):367–388

37. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781

38. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119

39. Nayak G, Ghosh R, Jia X, Mithafi V, Kumar V (2020) Semi-supervised classification using attention-based regularization on coarse-resolution data. In: Proceedings of the 2020 SIAM International Conference on Data Mining, pp. 253–261

40. Nielsen J, Landauer TK (1993) A Mathematical Model of the Finding of Usability Problems. In: Proceedings of the
41. Obar JA, Oeldorf-Hirsch A (2020) The biggest lie on the internet: ignoring the privacy policies and terms of service policies of social networking services. Inf Commun Soc 23(1):128–147
42. Oltramari A, Piraviperumal D, Schaub F, Wilson S, Cherivirala S, Norton TB, Russell NC, Story P, Reidenberg J, Sadeh N (2018) Privonto: a semantic framework for the analysis of privacy policies. Semantic Web 9(2):185–203
43. Rao RS, Pais AR (2019) Detection of phishing websites using an efficient feature-based machine learning framework. Neural Comput Appl 31(8):3851–3873
44. Reidenberg JR, Russell NC, Callen AJ, Qasir S, Norton TB (2015) Privacy harms and the effectiveness of the notice and choice framework. ISILP 11:485
45. Turian J, Ratinov L, Bengio Y (2010) Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th annual meeting of the association for computational linguistics, pp. 384–394. Association for Computational Linguistics
46. Virzi RA (1992) Refining the test phase of usability evaluation: How many subjects is enough? Human Factors 34(4):457–468
47. Wang C, Miao Z, Lin Y, Gao J (2019) User and topic hybrid context embedding for finance-related text data mining. In: 2019 International Conference on Data Mining Workshops (ICDMW), pp. 751–760. IEEE
48. Ware C (2019) Information visualization: perception for design. Morgan Kaufmann, Burlington
49. Wu Q, Teney D, Wang P, Shen C, Dick A, van den Hengel A (2017) Visual question answering: a survey of methods and datasets. Comput Vision Image Underst 163:21–40
50. Yang Y, Cer D, Ahmad A, Guo M, Law J, Constant N, Abrego GH, Yuan S, Tar C, Sung YH et al. (2019) Multilingual universal sentence encoder for semantic retrieval. arXiv preprint arXiv: 1907.04307
51. Yang Z, Li L (2019) An online retrieval question answering system for featured snippets triggering. In: ICDMW, pp. 49–55. IEEE
52. Zhao R, Haskell WB, Tan VY (2018) Stochastic l-bfgs: improved convergence rates and practical acceleration strategies. IEEE Trans Signal Process 66(5):1155–1169

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.