In contemporary terms, data privacy has become a prevalent issue, considering the technical legal jargon used in laws enforced by law constituting bodies and a paradigm of directive laws, rather than actual regulation like the FTC (US), and/or the ascent of GDPR (EU). This paper examines the semantic similarity and syntactic dependency vectors between large chunks of law text examining particularly the GDPR and OSN’s Privacy laws, using Natural Language Processing. Although some conventional methods have significant drawbacks like the Degree of Strictness Method, methods like syntactic dependency and semantic similarity vector analysis have been employed. Results show a correlated sequence between the semantic similarity of the most important Articles of the GDPR and OSN’s privacy laws.

Keywords
Identity-Related Data, Privacy Policy, Natural Language Processing, Semantic Similarity Vectors, Syntactic Dependency, Personally Identifiable Information (PII).

1. Introduction:
According to the IBM and Ponemon study on data breaches across the world, the average cost of data breaches this year increased from 3.42 to 3.86 million i.e., 6.4% than the previous year highlighting the importance of monitoring PII extracted by websites and the way it’s processed, used and shared [2]. Governments have taken this into consideration when crafting special laws to protect data privacy. For example, EU has come up with GDPR demanding all OSNs and entities operating in the EU to adhere to them [6]. However, a tool is needed to evaluate the compliance level of these OSN policies to such laws. Attempts have been made to automate the process of evaluation. Recently, many open source packages have been released to utilize Natural Language Processing in such tasks. But none could provide a standard solution that can be applicable to a broader set of laws with different ontologies.

In this paper, we propose a general approach to evaluate the semantic similarity of privacy policies of OSNs with the government laws, by utilizing NLP. We tested our approach on the EU General Data Protection Regulation (GDPR): Chapter 3 and the privacy policies of three websites: Facebook, Twitter and Reddit. We present three models to estimate the compliance of these websites policies to GDPR rules using NLP. First model calculates the degree of strictness of some text. Second model calculates the subject-object-verb syntactic dependency semantic vectors between the bodies of the text. Third model calculates the semantic similarity vectors of articles discussing the same law.

The rest of the paper is organized as follows: Section 2 covers previous work. Section 3 define the pre-requisites, Section 4 explains the implemented models. Section 5 presents the results collected from the models. Section 6 discusses challenges, limitations and future work. Section 7 highlights the results and findings.

2. Related work:
The process of analyzing legal documents using of NLP has earned a lot of interest in recent days. [7] uses NLP techniques for the extraction of rules from the legal documents. Using deontic lightweight ontology and text-structure lightweight ontology, the context dependent modal verbs are identified as parameters to such categories as obligations, prohibitions or permissions. Stanford parser was used for grammatical representation, and a Combinatory Categorial Grammar (CCG) approach was used for extraction of logical dependencies [7]. Although the research proved effective in identifying the context of the modal verbs and whether the context of text structures could be classified according to their modal verbs, it did not play a major role in identifying the semantic similarity between the entities involved in the similarity vector matrix. On 25th May 2018, the European Union’s GDPR went into effect, and since then, websites have been updating their policies according to its requirements. Many efforts have been made to understand the changes in privacy laws and GDPR’s impact on social platforms [6] studied the effect of GDPR on online privacy policy in multiple languages comparing them from before and after GDPR was applied. They used Jaccard similarity index to identify similar policy text to confirm if changes have been made before and after GDPR was enforced. Then they used lemmatization/stemming to find if certain words from the GDPR were mentioned in the policy. Tools were developed using NLP, which helped cloud entities be self-
compliant to their own privacy laws. The user was given an option to create his choice of privacy rules (termed as Consumer rules) using his data. By doing this, the user was providing his/her intentions to the platform on how to use his personally identifiable information. An explanation for an automatic way of detecting typologies of regulations, and extracting the related arguments using Norme In Rete (NIR) project (Legislation On Net) which has access point of normative documents on the web was presented in [1]. SFL Scientific has attempted to algorithmically extract key terms from legal documents and classify the document in a predefined set of categories. Their method involved two steps

i) Feature engineering: using regular expression tokenizer, matching to a fuzzy dictionary and ii) Modeling: XGBoost was implemented as a meta-classifier [8].

3. Pre-Requisites:

Web crawler is a tool which navigates through Web pages in an automated manner to extract specific kind of information. We developed a crawler to extract the text of the websites privacy policies and save them in different text files. The web crawler works as follows: i) Website URL is provided as an input. ii) Crawler collects the links from the website main page and/or the API. iii) Extracts the contents of the privacy policy page. iv) uses Boilerpipe [4] to clean by removing the surplus clutter. The final output files from the above steps are used as the input for the implemented models.

NLP is helpful for machines to understand the combination of ways in which language can be modelled. i.e. NLP is used in variety of cyber-security platforms to identify the threat level index surmounted by social bots or other malicious entities. NLP uses grammar and vocabulary rules to perform functionalities like POS tagging, Parsing, Dependency tagging, Similarity vector and other tools to approach language from an objective point of view. These functionalities are present in libraries as algorithms. Of the NLP libraries present, we used Spacy and LexNLP.

SpaCy [9] is a new Python NLP library known for a diverse interface. It presents the user with one algorithm for each purpose. Spacy helps the system understand the sentences, Part-of-Speech tagging, Syntax driven sentence segmentation, identifying the Name Entity Recognition (NER), easy deep learning integration and it supports 33+ languages. In this paper we are using spaCy v2.x to understand the rules and compare them.

Gen-Sim is most commonly used for topic modeling and similarity detection. It’s robust, efficient, and scalable, and the sub-field semantics analysis is one of the most exciting areas of natural language processing. Gen-Sim was not used in any methods but was tested.

LexNLP is the only Python NLP package which converts unstructured legal documents to structured objects. It includes functionalities such as document segmentation, titles and section headings identification, extraction of over eighteen types of structured information like distances and dates, named entities extraction, transforming text into features for model training, and build unsupervised and supervised models such as word embedding or tagging models.

4. Implementation:

To achieve our goal, we experimented with three different implementations using NLP.

4.1 Degree of Strictness Method D(s):

To establish a metric of the legal jargon used in law documents, it is imperative to understand the usage of modal verbs, as stated by the deontic reasoning used in [7]. The Stanford Parser model does evaluate the lightweight ontology context specific to text structure and to deontic reasoning. To attain an idea regarding how a specific law document’s use of jargon is would be a tiresome task, considering the synonyms of prohibition, permission and obligation that might not be considered modal verbs in the first place but modal verbs do not constitute the acts of let’s say prohibition, permission and obligation. There is a whole glossary of legal terms with thousands of terms that can symbolize prohibition or obligation and cannot be extracted from a legal document for the purposes of detecting the context accurately by using current NLP libraries without training the models extensively. However, to acquire an idea of the language rather than an accurate depiction of degree of strictness, a certain list was created with the most common verbs for Permission and Obligation.

\[ \text{Permission} = P = \{\text{“can”, “may”, “could”, etc.}\} \]
\[ \text{Obligation} = O = \{\text{“have to”, “must”, “should”, etc.}\} \]

For prohibition, a different method had to be applied rather than to detect the words from a pre-created dictionary. The context is dependent on adverbs, and Spacy’s detection of token dependency “neg”.

\[ \text{token.pos} = \text{“adv” and token.dep} = \text{“neg”} \]

This is not the only method used for detecting words that symbolize prohibition. So, a sample dictionary of the most common prohibition nouns had to be created.

\[ \text{Prohibition} = \text{Pr} = \{\text{“prohibited”, “restricted”, “consequences”, etc.}\} \]

Once a count of \( P \), \( O \) and \( Pr \) has been extracted from a document, the following formula is used to establish a degree of strictness.

\[ D_s = \frac{(P+O)-P_i}{P+O+P} \]

\( Pr \) and \( O \) are representative of what is restrictive in nature. However, they are not actually a good indicator of whether something is being restricted. That would be a context specific problem. But since both are representative of it, knowing the modal verbs associated with them gives a general idea. \( P \) is subtracted from their sum due to the mix-in of token dependencies “neg” and the permission modal verbs like “can”. For example:
“The hospital can’t ban patients from registering for insurance.”

\( P = “can” – \text{auxiliary verb} \quad Pr = “n’t” \text{ or } “not” \).

The \( D_f \) or the difference between two \( D(s) \) or documents can be calculated by:

\[
D_f = D_f^1 - D_f^2
\]  

(2)

4.2 Subject Object Verb Syntactic Dependency via Threshold Method (SOV):

A relatively reasonable approach than the \( D(s) \) method would be to adopt the SOV method. The SOV method finds all the appropriate combinations between tokens and/or words that fit a certain criterion, between two documents. 1. Only two sentences can be compared at a time. 2. All sentences have to be compared between both documents. 3. Tokens and/or words from the foundational document in a given sentence can only be compared to the token/words of another sentence in the second document. The SOV method uses Spacy’s dependency tagging and similarity vector method.

First, a combination of two subjects \( S_1 \) and \( S_2 \) is found between two sentences based on their syntactic dependency. Then, a similarity vector method is used between two subjects to check whether they are semantically similar. If the similarity score is greater than the threshold (user defined), then the search continues within the very same sentence for \( O_1 \) and \( O_2 \), and similarly for \( V_1 \) and \( V_2 \). Once the combination of syntactic dependency similarities all end up above the threshold, it can be assumed that the two sentences are similar based on the defined \( \tau \) or threshold.

\[
\tau = 0.65 \text{ (threshold)}
\]

\[
\begin{align*}
&iff (S_1, \text{dep} = S_2, \text{dep}) \& (S_1, \text{sim}(S_2) \geq \tau) \\
&then \quad iff (O_1, \text{dep} = S_2, \text{dep}) \& (O_1, \text{sim}(O_2) \geq \tau) \\
&then \quad iff (V_1, \text{dep} = V_2, \text{dep}) \& (V_1, \text{sim}(V_2) \geq \tau) \\
&s = 1 \text{ (SOV)}
\end{align*}
\]  

(3)

Although \( s \) is a good indicator of the context similarities between two sentences regarding subjects, objects, and the act performed between the subjects and objects, the similarity vector shows aberrant behavior due to the change in \( \tau \). However, to observe the similarity ratio \( s_0 \) is more important. \( s_0 \) is the SOV syntactic dependency without the constraint of \( \tau \) to give a general idea of a semantically similar sentence to non-semantically similar sentence with the same subject, object and verb dependencies. This can be demonstrated as:

\[
\begin{align*}
&A \text{ is the event where subject dependencies are similar above a threshold.} \\
&A_1 \text{ is the event where subject dependencies are similar without a threshold.} \\
&B \& B_1 \text{ are the events that can only happens when } A \text{ is true} \\
&\quad A = (S_1, \text{sim}(S_2) \geq \tau) \iff (S_1, \text{dep} = S_2, \text{dep}) \\
&\quad A_1 = (S_1, \text{sim}(S_2) \geq \tau) \iff (S_1, \text{dep} = S_2, \text{dep}) \\
&\quad B = (O_1, \text{sim}(O_2) \geq \tau) \iff (O_1, \text{dep} = S_2, \text{dep}) \\
&\quad B_1 = (O_1, \text{sim}(O_2) \geq \tau) \iff (O_1, \text{dep} = S_2, \text{dep}) \\
&\quad C = (V_1, \text{sim}(V_2) \geq \tau) \iff (V_1, \text{dep} = S_2, \text{dep}) \\
&\quad C_1 = (V_1, \text{sim}(V_2) \geq \tau) \iff (V_1, \text{dep} = S_2, \text{dep}) \\
&s_0 = \frac{A_1 \& B_1 \& C_1}{A \& B \& C} 
\end{align*}
\]  

(4)

In the \( s_0 \) calculation, it is important to note that object and verb dependency similarities are not cumulatively added if the threshold for subject dependency condition isn’t true. This is because, in order for object dependency similarities to be calculated between two sentences without the threshold, the subject dependencies should be similar. This means that in theory, summing the similarity vectors without the threshold constraint won’t make sense unless the foundational parts of speech element have higher similarity. For example, if \( S_1 \) = “Hospital” and \( S_2 \) = “Clinic”, then a similarity vector would be cumulatively added before \( \tau \) is applied. \( (O_1, \text{sim}(O_2)) \) can not be calculated if \( \tau \) is not applied to the subjects or \( A \) isn’t true. These restrictions on cumulative non-threshold applied calculations to take place because if two subjects are not similar, it doesn’t make sense to cumulatively add the objects or the verbs and their respective similarity vectors. This method was tested on the 16 articles of GDPR’s Chapter 3 “The Rights of The Data Subject”. \( \tau \) was set at 0.65 after testing of multiple self-typed paragraphs with known dependencies where the similarity vectors pin pointed a range between 0.50 and 0.75 as the most appropriate ones.

4.3 Semantic Similarity Vector:

In this algorithm we based our results on the similarity vector between GDPR articles and website privacy policy sections that are dealing with the same subject.

Step 1: We loaded 12 GDPR articles into our program and the privacy policy from 3 websites. The website privacy policy was divided into sections. Initially, we experimented with lexNLP’s “Get Sections” method. However, the results were not logical. Instead we developed our own section division method that relies on text separation since titles are clearly stated in privacy policy documents.

Step 2: To map GDPR articles to policy sections based on topic; we extracted Noun Chunks from the privacy policy sections as well as from GDPR article titles. Using similarity vector,
we identified privacy policy sections that has noun chunks similar to GDPR article title noun chunks. However, this method missed a few sections, so we went through another phase of mapping by looking for the noun chunk in the entire section. For word search we tried using NLP by similarity method, however, it was very slow and not able to identify enough variances of the same word. Instead we used get close matches method from “Difflib” library. This function was more dilute, but we adjusted the matching by increasing the matching coefficient.

Step 3: The organization of GDPR articles is very clear as each section talks about one topic. However, website privacy policies are divided in varies ways for example the same GDPR law may be mentioned in pieces across different sections. Thus, in this step we mapped exact sentences between policy and GDPR. Calculating the similarity vector between cleaned text. The results were found twice once between the two sentences with similar noun chunks and once between the policy sentence and the full GDPR article discussing the same topic. Based on our observation and manual analysis, when the similarity vector is below a certain threshold the two statements are referring to different topics even if they use similar phrases. Thus, we manually identified a threshold which gives us the sentences having similar meaning.

Figure 1: Semantic Similarity vector method

5. Results and Findings:

5.1 Degree of Strictness & SOV Semantic Similarity Vector

The degree of strictness method is insufficient to accurately gauge the accurate lightweight ontology and/or the context of the texts. The first issue is the mix-in issue between the token dependencies “neg” and the permissive modal verbs. Establishing an accurate context from the word groupings in the different parts of the sentence is difficult to establish. Therefore P(i) was for the most part ignored and treated as P in the actual implementation of the program. However, the consequence was a deflated D(s) due to the large number of P being the reason for a diminished D(s), for a what-if mix-in situation between the “neg” dependency and the permissive modal verb.

Another huge limitation is the dis-regard for the context. The D(s) method simply establishes a count of the modal verbs and/or related synonyms. This limits the approach to either increasing the pre-created dictionary to contain all possible synonyms of the three types of modal verbs, and/or the glossary of legal terms, or absolutely disregard modal verbs and or their synonyms in the first place and extract an idea of the contextual deontic reasoning behind the text structure.

Facebook, Twitter and Reddit’s privacy laws were compared to GDPR’s Chapter 3 “Rights of The Data Subject” and it’s articles. The results from the D_f, s and s_0 methods are shown in Appendix A.

Although the D_f is a relatively unreliable method for gauging the degree of strictness in absolute contextual terms, there does seem to be a correlation between the three companies in terms of their legal language, with respect to modal verb usage and/or their synonyms. Appendix A shows that Facebook’s average D_f is greater than the other two counterparts in respect to all articles. With a mean of 0.69 D_f, Facebook’s legal language and/or usage of momentous modal verbs or their synonyms is different from the GDPR’s usage. Article 16, “The Right to Rectification” seems to have the lowest D_f, perhaps due to the short amount of text in the actual article, surfacing only to a paragraph. There does seem to be many anomalies like in Article 19 “Notification obligation regarding rectification or erasure of personal data or restriction of processing” which seems to have almost no usage of modal verbs except for obligatory ones due to which there seems to be a high D_f for all companies. So, an inference can be drawn that the usage of the D_f is the least favorite indicator of the contextual strictness of a legal document, no matter how sparse the pre-created dictionary is.

Appendix A also refers to the s method. Facebook had the highest SOV dependencies relative to Reddit and Twitter. Article 13, 14, 15 and 17 had the most SOV syntactic dependency commonalities with all 3 companies. Article 13 of the GDPR pertains to the nature of information provided to the data subject by the controller. It talks about the specific elements a controller should disclose to the data subject. Now, a common assumption would be that all 3 companies in legal terminology adhere to Article 13, at least on paper. That assumption is far-fetched due to the reality that one knows the subjects, objects and verbs were used in the same syntactic dependency in the company’s privacy law and the GDPR’s privacy law. However, it is also not incorrect to assume otherwise. The s method identifies the theme of the discussion with its’ primary players more accurately than the context, although the verb dependency root causation is enough to assume that the context just might be the same. Another keen observation might be that Article 13 and 14 have more text than Article 19 and 20, and therefore are more likely to end up with a higher s. Although that is true to some extent, it still doesn’t account for the fact that Article 23 which talks about judicial restrictions and has the same amount of text as Article 13 and 14, still shows a lower s. That is because Facebook, Twitter and Reddit are more likely to discuss their data policy and disclosure policy rather than judicial proceedings and regulatory functions.
Article 17 discusses the right to be forgotten as its main theme, and discusses personal data in abundance, therefore being correctly recognized by the $s$ method for showing the most dependencies.

Appendix A also shows how SOV Semantic Similarity ratio is an optional way of looking at how $\tau$ affected the relationship between the SOV dependencies and the un-recognized or potential dependencies which were rooted out of the equation because of the threshold. Article 13, 14 and 15 had the most potential in terms of reaching an absolute SOV syntactic dependency but couldn’t because of the $\tau$ set at 0.65. A sensitivity test might reveal how this figure changes subject to lowering $\tau$. This also reveals that even though Facebook, Twitter and Reddit discuss the same subject and/or the object, actions recommended might have been different due to the syntactic dependency of the verbs.

5.2 Semantic Similarity Vector:

Our method can identify the existence and absence of GDPR rules in privacy policy laws. Based on figure 2, we can observe that both Reddit and Twitter are applying all GDPR articles from 12 to 23. However, Facebook is not applying articles 20 and 21. Both figure 2 and figure 3 represent the similarity vector results. Figure 6.a shows the sentence to sentence similarity, whereas figure 6.b shows sentence to article similarity. As we see the article to sentence similarities are slightly higher. When dividing the text into sentences we may ignore sentence dependency and paragraph flow. So, we repeated the similarity vector calculation considering complete GDPR article against policy sentence trying to some extent to bring back the complete context into the picture.

6. Discussion and limitations

The SOV syntactic dependency approach faces the similarity threshold optimization problem. That is, what value of $\tau$ would be the most appropriate since tokens that might be somewhat similar or a lot similar might be left out of the computation due to a higher or lower $\tau$. This creates the need for $\tau$ to either be calculations based on past computations, but the context of legal language and jargon makes it a trivial exercise.

Another important issue is setting a metric for an entire document’s similarity vector to another document. The approach used above either by using $s_0$ to get the impact of the $\tau$ or using $s$ to develop a numeric count do not take into consideration the degree of importance of auxiliary verbs or multiple verbs acting differently upon an object.

To enhance section mapping we thought about using dependency tagging to identify the main subject of a sentence, which should help us identify the main topic of that sentence. However, due to the complex structure of law lexicon it was difficult to automatically identify the topic of a sentence. Furthermore, different sentences depend on each other, but NLP is unable to find these semantic dependencies. For example, lexNLP promises to extract conditions from legal documents but from testing we observed that it only extracts based on the sentence so if the condition is divided between two sentences it will only identify the first half containing the condition.

Similarity vector calculation produced a very narrow scale to identify the degree of compliance. As we discarded similarity less than a certain threshold indicating different topics. Reducing our ability to conclude the degree of adherence of privacy policies to government laws.

For future work we can look further into the following topics; resolving sentence dependencies across a paragraph by developing a method to resolve pronouns. Furthermore, NLP can be trained to recognize similar sentences, thus by preparing a dataset containing legal statements with similar meanings similarity vector can provide more accurate results. Gen-Sim is also a good tool for validating the distance between word vectors, but it again came with the problem of semantic similarities where language has to be trained.

7. Conclusion:

With our growing reliance on online services and the increasing amount of PII collected by these services, we need a method to analyze the compliance of the privacy policy provided by these online services and the privacy policy laws enforced by different countries. In this project we developed three algorithms using NLP to evaluate the compliance of website privacy policy to government privacy laws. We tested our algorithms on GDPR ‘Rights of the Data User’ vs the privacy policies of three websites Facebook, Twitter and Reddit. The idea was to see if the same context is in discussion.
References


Appendix A:

Tabulated results

<table>
<thead>
<tr>
<th>τ = 0.65</th>
<th>Facebook</th>
<th></th>
<th>Reddit</th>
<th></th>
<th>Twitter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$D_f$</td>
<td>$s$</td>
<td>$s_0$</td>
<td>$D_f$</td>
<td>$s$</td>
</tr>
<tr>
<td><strong>Article 12</strong></td>
<td>0.86</td>
<td>85</td>
<td>0.06</td>
<td>0.727</td>
<td>45</td>
</tr>
<tr>
<td><strong>Article 13</strong></td>
<td>0.636</td>
<td>25723</td>
<td>0.15</td>
<td>0.504</td>
<td>14938</td>
</tr>
<tr>
<td><strong>Article 14</strong></td>
<td>0.818</td>
<td>29570</td>
<td>0.13</td>
<td>0.685</td>
<td>16419</td>
</tr>
<tr>
<td><strong>Article 15</strong></td>
<td>0.629</td>
<td>16144</td>
<td>0.15</td>
<td>0.497</td>
<td>8812</td>
</tr>
<tr>
<td><strong>Article 16</strong></td>
<td>0.163</td>
<td>9</td>
<td>0.03</td>
<td>0.03</td>
<td>15</td>
</tr>
<tr>
<td><strong>Article 17</strong></td>
<td>0.663</td>
<td>29570</td>
<td>0.06</td>
<td>0.53</td>
<td>1327</td>
</tr>
<tr>
<td><strong>Article 18</strong></td>
<td>0.941</td>
<td>245</td>
<td>0.07</td>
<td>0.808</td>
<td>231</td>
</tr>
<tr>
<td><strong>Article 19</strong></td>
<td>1.163</td>
<td>16</td>
<td>0.06</td>
<td>1.03</td>
<td>12</td>
</tr>
<tr>
<td><strong>Article 20</strong></td>
<td>0.463</td>
<td>974</td>
<td>0.12</td>
<td>0.33</td>
<td>312</td>
</tr>
<tr>
<td><strong>Article 21</strong></td>
<td>0.591</td>
<td>76</td>
<td>0.04</td>
<td>0.459</td>
<td>190</td>
</tr>
<tr>
<td><strong>Article 22</strong></td>
<td>0.941</td>
<td>76</td>
<td>0.03</td>
<td>0.808</td>
<td>122</td>
</tr>
<tr>
<td><strong>Article 23</strong></td>
<td>0.437</td>
<td>553</td>
<td>0.05</td>
<td>0.57</td>
<td>1701</td>
</tr>
<tr>
<td><strong>Mean</strong></td>
<td>0.69</td>
<td>8586.75</td>
<td>0.08</td>
<td>0.58</td>
<td>3711.08</td>
</tr>
<tr>
<td><strong>STD</strong></td>
<td>0.27</td>
<td>12747.97</td>
<td>0.05</td>
<td>0.26</td>
<td>6106.45</td>
</tr>
</tbody>
</table>

Appendix B:

Method 1 and 2: [https://github.com/mahadkhanleghari/NLP-Data-Security](https://github.com/mahadkhanleghari/NLP-Data-Security)