mathematics

MDPI

Article

Visual Analytics Using Machine Learning for Transparency Requirements

Samiha Fadloun [1,*], Khadidja Bennamane [1], Souham Meshoul [2,*], Mahmood Hosseini [3] and Kheireddine Choutri [4]

[1] Ecole Nationale Supérieure d'Informatique (ESI), BP 68M, Oued Smar, Algiers 16309, Algeria
[2] Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[3] JPMorgan Chase, 1 Chaseside, Bournemouth BH7 7DA, UK; mahmood.hosseini@jpmorgan.com
[4] Aeronautical Sciences Laboratory, Aeronautical and Spatial Studies Institute, Blida 1 University, Blida 0900, Algeria; choutri.kheireddine@univ-blida.dz
* Correspondence: s_fadloun@esi.dz (S.F.); sbmeshoul@pnu.edu.sa (S.M.)

Abstract: Problem solving applications require users to exercise caution in their data usage practices. Prior to installing these applications, users are encouraged to read and comprehend the terms of service, which address important aspects such as data privacy, processes, and policies (referred to as information elements). However, these terms are often lengthy and complex, making it challenging for users to fully grasp their content. Additionally, existing transparency analytics tools typically rely on the manual extraction of information elements, resulting in a time-consuming process. To address these challenges, this paper proposes a novel approach that combines information visualization and machine learning analyses to automate the retrieval of information elements. The methodology involves the creation and labeling of a dataset derived from multiple software terms of use. Machine learning models, including naïve Bayes, BART, and LSTM, are utilized for the classification of information elements and text summarization. Furthermore, the proposed approach is integrated into our existing visualization tool TRANSPVIS to enable the automatic detection and display of software information elements. The system is thoroughly evaluated using a database-connected tool, incorporating various metrics and expert opinions. The results of our study demonstrate the promising potential of our approach, serving as an initial step in this field. Our solution not only addresses the challenge of extracting information elements from complex terms of service but also provides a foundation for future research in this area.

Keywords: information visualization; visual analytics; transparency analytics; machine learning; natural language processing

MSC: 68T01; 68T30; 68T35; 68T50; 68T99

Citation: Fadloun, S.; Bennamane, K.; Meshoul, S.; Hosseini, M.; Choutri, K. Visual Analytics Using Machine Learning for Transparency Requirements. *Mathematics* **2023**, *11*, 3091. https://doi.org/10.3390/math11143091

Academic Editors: Shuo Yu and Feng Xia

Received: 30 May 2023
Revised: 5 July 2023
Accepted: 11 July 2023
Published: 13 July 2023

1. Introduction

Digitalization has automated daily tasks through machine learning, with each task having software designed for it. For example, YouTube's video recommendation system suggests new content based on user preferences. These systems access a huge amount of data from other apps, such as search history or social media. However, users often give consent to use their personal data when accepting app terms of use without fully understanding the extent of data revealed or who it is accessible to.

Software has accompanying transparency notices, including terms of use and privacy policies. Before using an app, users must accept and read these terms. They contain blocks of text focused on privacy, security, regulation, etc., and represent the software's transparency requirements. Understanding the flow of information shared between software

stakeholders has become crucial to protect users' personal data. Researchers have developed a model and language to represent transparency requirements, called TranspLan. TranspLan [1] is based on three components: stakeholders, information elements, and the binary relationships between them. Stakeholders are involved in information exchange and can be people, departments, or organizations. Information elements are pieces of information exchanged between stakeholders such as data, policies, and processes. While the original representation of transparency documents based on long texts can be read by anyone, generally only transparency experts can use existing representations. However, all people interacting with software also deserve to understand how their personal data are handled and used. Unlike experts, they need a much simpler and automated representation than what already exists.

For example, a new tool has been developed to display TranspLan information, called **TranspVis** [2]. **TranspVis** helps users to effectively express their knowledge by manually creating information elements, stakeholders, and their relationships using interactive interfaces and functionalities. However, there is no automation for analyzing data from the original text policy. **TranspVis** falls under the domain of information visualization. However, it can also be integrated with other domains such as machine learning, given its advancements in various fields including medical research [3–5], agriculture [6,7], data science [5,8], and more. From machine learning, adding a text analysis and natural language processing [8,9] would be interesting, especially in relation to software policies, to extract transparency requirements, keywords, and relationships.

In this context, visual analytics [10,11] emerges as a solution that addresses this integration and combines various domains with information visualization. Visual analytics for ML can assist transparency experts or users in automatically extracting and visualizing information elements and stakeholders to quickly and comprehensively understand software policies.

Through this paper, our aim is to propose a visual analytics tool for the automatic detection and visualization of information elements in transparency. The goal is to make the comprehension of software transparency requirements more accessible to all users, not just experts. The main contributions of this work can be summarized as follows: (1) analyze software policies with machine learning and natural language processing, (2) create an annotated text data set for transparency analytics, (3) propose an approach to automatically detect information elements, and (4) integrate the proposed machine learning model with **TranspVis** [2] in order to create a visual analytics tool.

The remaining sections of the paper are organized as follows: Section 2 provides an overview of transparency engineering and the current state of the art, encompassing machine learning, natural language processing (NLP), and visual analytics. In Section 3, we conduct a thorough analysis of the requirements based on valuable feedback from experts in the domains of transparency and computer science. Section 4 presents our proposed approach to transparency analytics utilizing NLP techniques, along with the enhancements made to the existing **TranspVis** (Section 5) tool to meet the specified requirements. Section 6 encompasses an evaluation of the proposed tool and subsequent discussions. Finally, the paper concludes by summarizing the research work undertaken and outlining future research directions.

2. State of the Art

The role of technology in enhancing transparency is becoming increasingly significant. Scholars have noted that the next generation of transparency policies will be characterized by technological advancements and collaborative approaches [12,13]. Various models have been proposed to analyze transparency requirements. One such model is the Transparency Actors Wheel [14,15], which focuses on mapping the flow of information among relevant stakeholders. By identifying the stakeholders involved, this model aids in understanding the origins of information, its providers, its recipients, and the channels through which it is transmitted.

The primary focus of this study is the Transparency Depth Pyramid, which represents the second model of interest. The pyramid examines various levels of transparency and categorizes transparency requirements into the following dimensions, as defined by Bannister et al. [16]:

- Data Transparency: This dimension encompasses questions related to "what, when, where, or who?" and involves information, content, or data. For instance, in a hosting service platform, data transparency would provide clients with information on the server performance and the pricing for each plan.
- Process Transparency: This dimension pertains to the question of "how?" and encompasses procedures, processes, behaviors, and interactions. It delves into the specifics of how certain tasks are carried out. For example, it examines the encryption methods employed for data stored on servers and the measures in place to safeguard servers against cyber threats in a hosting service platform.
- Policy Transparency: This dimension pertains to questions related to "why?" and encompasses goals, intentions, and policies. For example, policy transparency in a hosting service platform would clarify why encryption is necessary for servers or why users have a limited storage capacity.

In summary, the Transparency Depth Pyramid model explores these different dimensions to enhance understanding and provide transparency in data, processes, and policies. These dimensions are referred to as information elements, which are our main focus.

On the other hand, a prototype called Transplan [1] is proposed to handle the Depth Pyramid dimension. Additionally, the **TRANSPVIS** [2] visualization is enhanced by introducing a new interface. However, no automation or machine learning approaches are applied in these prototypes. Our main objective is to automatically extract information elements from texts using machine learning approaches. We synchronize these extracted elements with a visualization that amplifies their impact in the visual analytics domain.

Visual analytics [10,11,17] are widely used in both theory and practice. We may also find other fields that have used information visualization and data analysis in visual analytics, like mathematics [18], machine learning (https://trustmlvis.lnu.se/ (accessed on 15 May 2023)) [19,20], and natural language processing (NLP) [9,21] (as a subfield of machine learning). Beginning in 2010, the number of papers proposing visual analytics with NLP [22] has grown significantly.

In the field of transparency analytics, researchers in visual analytics often concentrate their efforts on one of the following transparency requirements: data, process, policy, or relationships. A significant portion of research in this area has been dedicated to visualizing data privacy and cybersecurity. Cybersecurity researchers, for instance, prioritize network security by identifying potential vulnerabilities and attacks [23,24]. Zhang et al. [24] developed a visual analytics system for a large-scale network topology to evaluate the effectiveness of key technologies. Their focus was on computers and the processing of data, as well as potential attacks by other users.

For example, researchers often concentrate their efforts on addressing specific transparency requirements within social networks on the internet, such as Twitter or Facebook. As an example, Wang et al. [25] utilized GraphProtector, a visual analytics tool, to protect the privacy of social network users. Their study involved the creation of databases containing the personal information of individuals. GraphProtector is an innovative system designed to preserve privacy in network datasets through the interactive synthesis and application of various anonymization models.

Moreover, a study conducted by DeHart et al. [26] focused on privacy issues in social media networks such as Facebook and Twitter. The researchers collected data from users of these platforms and employed machine learning algorithms to analyze the collected information. The study involved a survey administered to social media users. A total of 250 participants took part in the study, with many of them having multiple social media accounts across platforms like Facebook, Twitter, and Reddit. The text-based responses

provided by the participants were analyzed using text analytical methods, and the content was categorized into three main groups: identity, age, and gender.

Chou et al. [27] have made significant contributions to the field of privacy-protecting visualization. The primary objective of their work is to mitigate the potential privacy concerns associated with the widespread practice of data collection. By utilizing a combination of data mining and visualization methods, they aim to enhance privacy protection measures and ensure the responsible handling of sensitive data.

A few privacy policy visualization systems have been proposed. However, many of them suffer from limited interactivity and static interfaces, which pose challenges for both experts and novices in terms of comprehension. For example, Ghazinour et al. [28,29] developed a privacy policy system that adopts a human-centric model of privacy policy notation. Other researchers have utilized tools such as UML or draw.io to represent policies and privacy. Nonetheless, these systems still face limitations in terms of their static interfaces and lack of interactive elements. These factors contribute to difficulties in understanding the systems and can prolong the time required for system analysis and utilization [30].

As mentioned earlier and described in Table 1, transparency researchers tend to focus more on analyzing private data rather than developing visualizations that assist in understanding and decision making based on such data. Additionally, existing visualizations often address only specific transparency requirements, such as privacy, security, or stakeholder visualization, without effectively synchronizing users' knowledge with the underlying database (DB). Moreover, transparency analyses rely on a limited set of straightforward tasks and restricted visualization techniques to extract and visualize information elements from software policies.

Table 1. Comparison criteria between research on transparency and visualization ("X" applied, "/" not applied).

Paper	Application	Data	Process	Policy	NLP	Visualization	DB Synchronization
Zhang et al. [24]	Networks	X	/	/	/	X	/
Wang et al. [25]	Social networks	X	/	/	/	X	/
Dehart et al. [26]	Social networks	Collected from users	/	/	X	/	/
Fadloun et al. [2]	Software policies	X	X	X	/	X	/
Chou et al. [27]	Privacy protecting	X		/	X	X	/
Ghazinour et al. [29]	Software policies	X	/	X	/	draw.io	/
Hossieni et al. [1]	Software policies	X	X	X	/	/	/
Julta et al. [30]	Software policies	/	/	X	/	UML/draw.io	/

3. Requirements Analysis

Based on interviews conducted with domain experts and computer scientists, a set of requirements was developed. These specifications were shared with the experts and further refined based on their valuable feedback, resulting in the following finalized list:

- **[R1] Automation of data generation:** Data generation is crucial for efficiently processing large amounts of data in a timely manner. By automating the data generation process, we can save time, reduce errors, and improve the accuracy and consistency of the data in transparency anlytics.
- **[R2] Transparency visual analytics:** Transparency visualization is becoming increasingly important as transparency becomes a key requirement for many information systems. Effective visualization techniques are necessary to communicate complex

transparency information to users and facilitate their understanding of the different aspects of transparency.

- **[R3] Interactions:** The proposed tool should be interactive. These interactions can help users automatically add information elements to their databases and improve them.
- **[R4] Synchronization between model, visualization and data:** Synchronization between the model, visualization, and data is crucial to ensure that the displayed information is accurate and up-to-date. By synchronizing these elements, users can be confident that the visualized data are representative of the underlying model and the latest available data.

In the following, we extend a visual analytics tool called **TRANSPVIS** [2] to address these requirements.

4. Proposed Approach: Transparency Requirements and NLP

Our solution's unique feature is the integration of machine learning in the process of detecting information elements related to software transparency in transparency documents [**R1**] (Figure 1). This breakthrough in the field of software engineering guided our thinking towards the proposed solution. Our basic hypothesis is to take a text (paragraph) and generate its information element along with its classification. To do so, we thought of a solution that enables the integration of the classification task with the text generation task. We sought to simulate the human thinking process. By reading software terms and conditions texts, we found that not all paragraphs contain information exchanges between software stakeholders. Some paragraphs are more related to general information and therefore do not affect the process of generating information elements. Thus, we thought of starting with a classification model that filters paragraphs and keeps only those discussing information exchanges. Then, these paragraphs were longer than the typical length of an information element, which is why we thought of shortening them using summary generators. Finally, each generated information element will be classified based on its characteristics, either as data, a process, or a policy.

We briefly present the different steps shown in the above diagram before detailing and justifying our choices in the following sections:

1. **Data Collection**: In this stage, we collect software terms of use and privacy policies to create the project dataset. These terms are collected by visiting the pages of these conditions online and copying the content of these text blocks. The collected data are validated by an expert in transparency.

In order to collect the maximum number of paragraphs, we visited the terms of use and privacy policy pages of different applications and software (for example, this page 2 represents the terms of use of the PREZI software), and we retrieved the raw content of the pages to build the dataset. In Table 2, we provide the sizes of the datasets that we were able to construct. For the first and second classifiers, we retrieved 500 paragraphs with labels. For classifier 1, the labels indicate whether the paragraphs contain information elements or not (true or false). For the second classifier, the labels indicate the class of the paragraphs, such as data, process, or policies. For the summary generation, we collected 2111 paragraphs and filtered them based on the type of text, such as cookies, personal information, etc. Then, we obtained 1000 valid paragraphs out of the total 2111.

Table 2. Machine learning dataset size.

Model	Size
Classifier 1	500 pairs
Summary generation	1000 pairs
Classifier 2	500 pairs

2. **Manual Annotation**: Data annotation is the process of labeling paragraphs derived from terms of use documents based on the presence or absence of an information element. An information element is a brief summary of a paragraph's primary topic. We manually built the necessary information elements for paragraphs that are tagged as holding an information element (true). Furthermore, we divide them into three categories depending on their content: data, policy, and procedure. The kind and source of information acquired by the service provider are referred to as data. The rules and regulations that govern the usage of the service are referred to as policy. The activities and procedures that the service provider or the user may execute with the data are referred to as processes. After completing the data annotation, experts review and verify the accuracy of each label and class assignment.

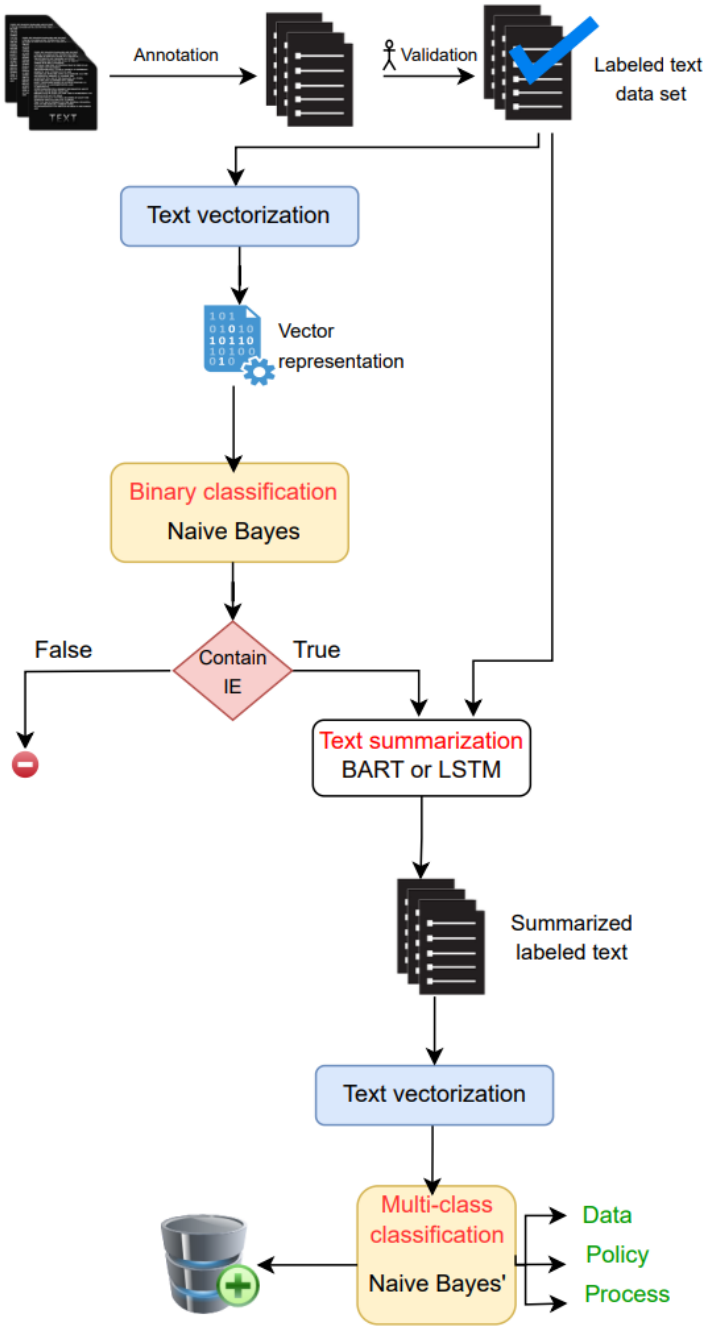

Figure 1. Proposed workflow of various natural language processing methods applied to extract information elements from software conditions and policies.

3. **Data Preprocessing**: After annotating the paragraphs from the terms of use, we move on to the data pre-processing step, which is an automatic step that takes the paragraphs as input and generates the words of the vocabulary. Data cleaning consists of the following steps: automatic filtering and cleaning and tokenization. Automatic filtering includes removing unwanted characters and symbols from the paragraphs, such as punctuation marks, numbers, and special characters. Additionally, we correct mistakes during the cleaning process. For example, we transformed English language abbreviations into their original form. For instance, "haven't" will be transformed into "have not". Tokenization is the splitting of the paragraphs into individual words or tokens, which are then added to the vocabulary.
4. **Text vectorization**: Machine learning models require numerical data as input. Hence, any textual data must undergo a transformation into a numerical representation. This process enables mathematical operations and model building on the text data. In this study, we adopt a vector representation for each paragraph, where the vector dimension is equal to the size of the vocabulary and each token has a numerical weight associated with it. We use a bag-of-words representation that incorporates n-grams [31] and computes the token weights based on the TF-IDF (Term Frequency-Inverse Document Frequency) method [32]. This method accounts for both the occurrence and the frequency of the tokens, and assigns higher weights to tokens that are less frequent, as they are more likely to affect the meaning of the sentence. The BART [33] model does not rely on vector representation. Instead, it handles long paragraphs and employs its own API to generate summaries.
5. **Text summarization**: We used the BART model [33,34] or LSTM [35] to shorten the lengthy texts we worked with. This model can create abstractive summaries of long documents and is very effective for text summarization.
    - LSTM: We chose a summary generator using LSTM. Thus, we took a paragraph or a long sentence and applied several processing layers to encode it. We then passed this encoded version of the text to another module that will decode it into a shorter version, which we will refer to as an information element. We apply the process as follows:
        - Sequence-to-Sequence Modeling (Seq2Seq): Seq2Seq models in NLP are used to convert sequences of type A into sequences of type B. Our problem is considered a Seq2Seq problem because it takes a long paragraph as input to be processed (the role of the encoder) and produces a shorter sentence containing the main idea of the input paragraph as output (the role of the decoder).
        - We use an approach proposed by Petal et al. [36] to set up the encoder–decoder (training and inference) and to achieve the best possible results, we also used:
            * Attention Mechanism: Despite the encoder–decoder architecture having several advantages, it also has limitations. To overcome these limitations, we utilized the concept of the attention mechanism. The idea behind the attention mechanism was to allow the decoder to flexibly utilize the most relevant parts of the input sequence, especially in long sequences. This mechanism is achieved by assigning weights to each word or token, and the most relevant tokens are those with the highest weights.
            * Word2Vec: Word embedding is a technique that transforms individual words into a numerical representation of the word (a vector). This vector is learned in a way that resembles a neural network, aiming to capture various features of the word in relation to the entire text. These features can include semantic relationships, definitions, context, etc. With these numerical representations, we can perform tasks such as identifying word similarity or dissimilarity. For example, the degree

of similarity between the words "purpose" and "reason" should be high. The effectiveness of Word2Vec as a word embedding technique lies in its ability to group similar word vectors and make strong estimations about the meaning of a word based on its occurrence in the text. These estimations lead to the creation of associations between words in the corpus.

  - Beam Search: To improve the output of the decoder, we have the choice between using a greedy search algorithm or a beam search algorithm. Greedy search algorithms choose the best word at each step, but experiments have shown that the highest-scoring word may not always be the best fit for constructing the sentence. On the other hand, beam search algorithms consider multiple candidate sentences at each step and choose the one with the highest score as the summary.

- BART is a sequence-to-sequence model that is pre-trained as a denoising autoencoder. It corrupts the text with a noising function and learns to restore it. It uses a standard transformer-based architecture with a bidirectional encoder like BERT and a left-to-right decoder like GPT. Ee chose to use BART [33] due to its large number of tokens and free access, as well as its similarity to GPT. According to Lewis et al. [33], BART is a pretrained model that combines bidirectional transformers and autoregressive transformers. Before BART, there were BERT (Devlin et al. [37]) and GPT (Radford et al. [38]). We explain the difference in training between the three models, in the example shown in Figure 2. The original document is $ABCDE$, and the interval $[C, D]$ is masked before encoding (BART 2020), resulting in the corrupted document '$A - B - E$' as input to the encoder. The decoder needs to reconstruct the original document using the output from the encoder and the preceding uncorrupted tokens.

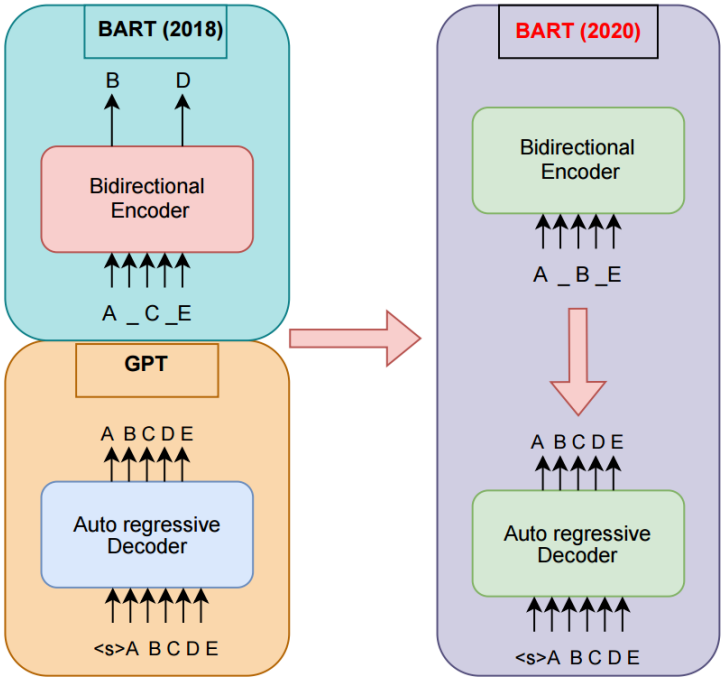

Figure 2. Schematic comparison of BART (2018 on the **left**) with BERT (2020 on the **right**) and GPT [33].

In our summarization task, the input sequence is the paragraph we want to summarize, and the output sequence is a summary (information element).

6. **Classification**: We performed a two-stage classification. In the first stage, the goal of the classification is to identify whether the input text contains an information element or not. Therefore, in the first stage, a binary classification is performed. A naïve Bayes classifier [39] is used for this purpose. To train the model, the vector representation of the annotated raw data is fed as input with the corresponding label (yes/no). In the second stage, the model is trained on the vector representation of the summaries obtained using BART or LSTM [35] for only texts containing IEs. The aim is to identify the type of the IE that is data, a process, or a policy. Therefore, in the second stage, a multi-class classification task is performed.

Naïve Bayes is a generative, probabilistic model that predicts the class of a paragraph presented as a word vector by calculating the probability of belonging to each class based on the following mathematical formula.

$$p(c|d) = \frac{p(c|d) * p(c)}{p(d)} \quad (1)$$

where c represents the class or category to which a data instance belongs, and d represents the data instance. The equation calculates the probability of a instance (paragraph) belonging to a certain class (binary or multi class).

There are three types of naïve Bayes models [39]: Gaussian, multinomial, and Bernoulli. In our case, we used a multinomial naïve Bayes classifier because it is primarily used for document classification problems. It observes the frequency of words to make predictions. In the first model, we used a multinomial naïve Bayes classifier to eliminate paragraphs that do not contain information elements. For the last model, we used it to obtain the classes of the information elements. We chose the naïve Bayes classifier due to its effectiveness in classification tasks, especially when working with small datasets. In our research, we have 500 paragraphs, and the naïve Bayes classifier is expected to yield better results considering the size of our dataset.

7. **Data splitting**: This process is performed to create different subsets of data for training and evaluating the models. We partitioned our data into two subsets: the training data and the validation data. The training data are used to fit the models and adjust their parameters. It comprises 80% of the total data. The validation data are used to assess the performance of the models and measure their accuracy. It comprises 20% of the total data. In our solution, we created distinct datasets for each model, meaning that the 80% used for the first model is not the same as the 80% used for the second and third models.

5. TRANSPVIS Visual Analytics Tool

TRANSPVIS [2] (https://youtu.be/8O8BWa0w2so (accessed on 31 December 2022)) is a visualization tool for transparency components (stakeholders, information elements, and relationships between stakeholders and information elements). One of the drawbacks of this visualization is that it is manual, as each transparency element has to be added manually to the visualization using interactions.

Our work involves extending **TRANSPVIS** by integrating machine learning techniques to transform it into a visual analytics tool [**R2**]. Therefore, we enhance the **TRANSPVIS** tool by integrating automatic functionalities that enable the detection of information elements in transparency documents through user interactions [**R3**]. Synchronization [**R4**] is offered between **TRANSPVIS** and ML results and the database. Figure 3 illustrates the workflow process of our proposition. Users can directly interact with the tool, express their knowledge, or upload datasets from the database. These datasets can be generated by ML models or saved by users (see Figure 4). They can analyze them or improve/update them if the user is an expert in transparency analytics.

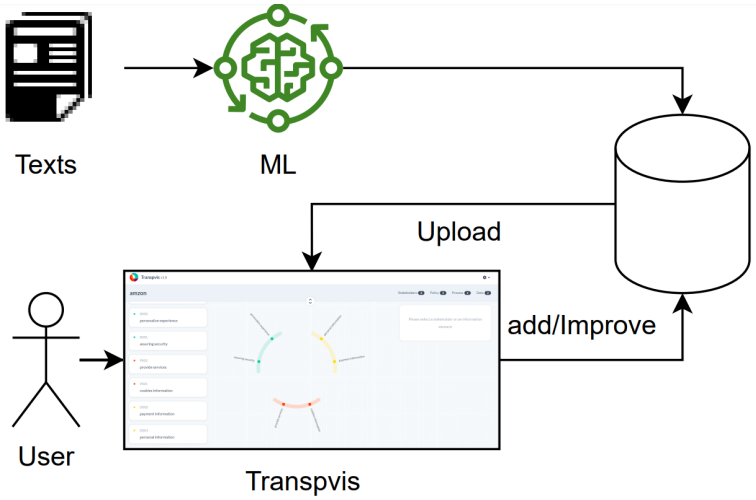

Figure 3. TRANSPVIS's workflow process.

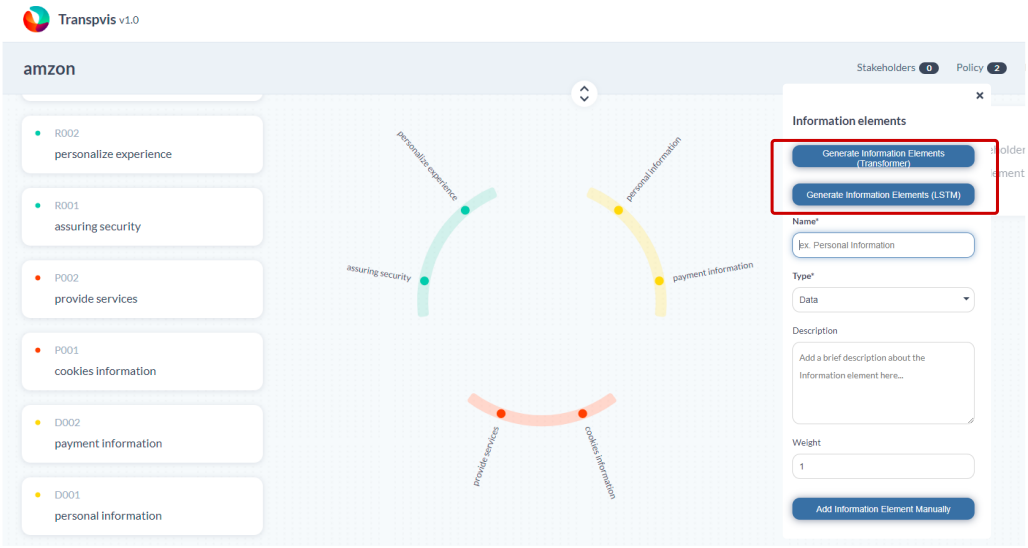

Figure 4. TRANSPVIS has been enhanced with a new feature for information element generation using machine learning. We have added two buttons that allow for automatic generation of information elements using either BART or LSTM.

6. Evaluation

After building the system and explaining in detail the system design and implementation phases, the evaluation phase is necessary to obtain feedback on the clarity, accuracy, and reliability of the system, including the prediction and automatic generation of information elements with the main visualization.

First, we evaluate the machine learning model independently using metrics such as precision, recall, and F-score for classification, and precision and loss for LSTM. Then, we assess the visualization system in conjunction with machine learning by involving an expert in the visual analysis cycle. This approach aims to enhance the knowledge and facilitate constructive examination through two test cases Notion and prezy Policies.

6.1. Model Evaluation

Classifier

To evaluate the performance of classification models, there are several metrics. Precision quantifies the number of positive class predictions that actually belong to the positive class. Recall quantifies the number of positive class predictions made from all positive examples in the dataset. The $F1_{score}$ is defined as the harmonic mean between precision and recall.

To evaluate our model, we use the $F1_{score}$ because this metric is used when there is an imbalance between classes. It is defined in terms of precision and recall. The formulas for each of these measures are as follows:

$$Precision = \frac{true\ positive}{true\ positive + false\ positive} \tag{2}$$

$$Recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{3}$$

$$F1_{score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

True positives and true negatives refer to samples that were successfully classified as true or false in the case of the first classification model, or classified as data, process, or policy in the case of the last classification model. False negatives and false positives are misclassified samples.

We obtained an $F1_{score}$ equal to 0.81, demonstrating the model's ability to predict positive individuals well. As for the information classification model, we obtained acceptable results for each class. The results obtained are represented in Tables 3 and 4. We would like to point out that in binary classification, we obtained good precision results for both classes, with a precision of 0.83 for the false class and 0.69 for the true class. However, the recall score is low for the false class, measuring at only 0.15. This can be attributed to the high similarity between the classes and the small size of the datasets. On the other hand, the accuracy for the binary classification (true/false) is 0.70.

Regarding multi-class classification, we achieved satisfactory metrics for the data class. However, as mentioned earlier, we encountered challenges due to the close similarity between the policy and process classes, resulting in low recall values of 0.14 for the policy class and 0.25 for the process class. Furthermore, the accuracy in multi-class classification is 0.62.

Table 3. Evaluation of binary classification.

Metrics/Class	False	True
Precision	0.83	0.69
Recall	0.15	0.98
$F1_{score}$	0.26	0.81

Table 4. Evaluation of multi-class classification.

Metrics/Class	Data	Policy	Process
Precision	0.63	0.67	0.50
Recall	0.96	0.14	0.25
$F1_{score}$	0.76	0.24	0.33

6.2. TRANSPVIS *Visual Analytics Evaluation*

To evaluate our system, we used the visual analysis cycle that begins with data collection. Raw data have no value in itself since they do not reflect meaningful information unless processed to extract the necessary information according to the needs. We used the data we collected to create a machine learning model, and TRANSPLAN used the transparency data to create the complete transparency visualization. Expert feedback will be used to refine the model and integrate it correctly with the visualization to create knowledge. A visual analysis will encourage constructive examination, correction, and rapid improvement of the models, which will improve knowledge and facilitate decision making.

6.2.1. Case Study

In this section, we will conduct a case study comparing the results obtained by the two models (BART and LSTM) with our initial view of the solution.

Since the goal of the model is to simulate the human thinking process, our main measure of evaluation is the closeness of the prediction to a sentence written by a human after reading and understanding a paragraph.

Notion Policy

In this first case, we take as input a part of the terms of use for the Notion software (https://www.notion.so/Privacy-Policy-3468d120cf614d4c9014c09f6adc9091 (accessed on 31 July 2022)).

We have input 18 paragraphs (the number of paragraphs is calculated by the number of line breaks). The following presents the results of generating the information elements:

- Result of the BART method: Figure 5 presents the results of generating information elements using the BART transformer.
- Detailed observation of visualization using BART: The system generated 11 information elements. We noticed that it eliminated the titles and kept only the paragraphs containing information elements. We also noticed that the information elements are a bit long compared to what we expected, which is good because it provides more details about the information element, but on the other hand, it requires users to read long sentences.

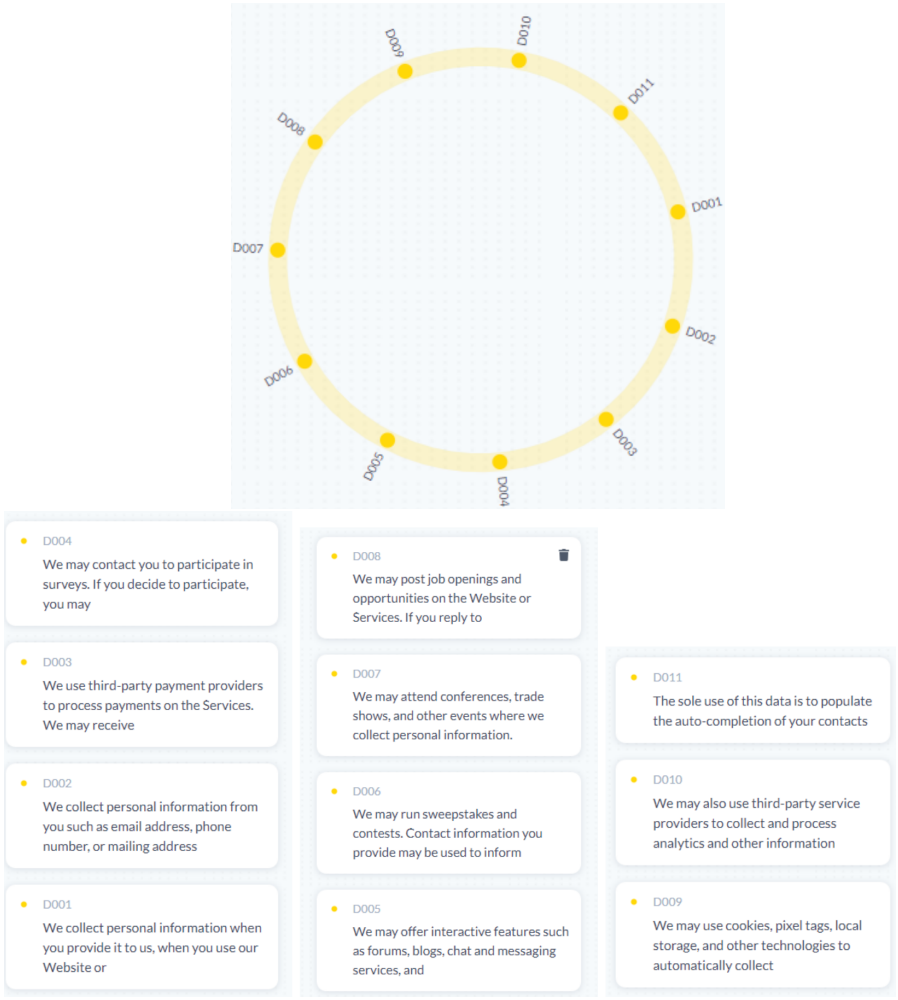

Figure 5. BART results of notion policy. The yellow color represents the information element data type. Ten data types have been extracted and are represented by their respective IDs.

- Results of the LSTM method: Figure 6 represents the results of generating information elements using the LSTM network.
- Detailed observation of visualization using LSTM: The system generated fewer information elements even though the input text is the same for both models. In terms of prediction quality, LSTM gave good results using short informative phrases containing essential keywords. However, in terms of the quantity of prediction, during the testing phase, we noticed that our model suffers from an underfitting problem; it does not generalize well to new data. Another note is that the classification of the information element "cookies" changed between the first and second model. In fact, the classification of the information element is governed by how it is formulated. Thus, if we say "personal information", it can be classified as data, but if we say "why collect information elements", the classification changes to policy. Thus, the classification of the information element is only related to how it is formulated. Therefore, the most important part of this system is the summary generator.

Figure 6. LSTM result of notion policy. Six information elements have been extracted: four for data (yellow), one for process (red), and one for policy (green).

Prezi Policy

In this case study, we take as input a part of the Prezi terms of use (https://prezi.com/privacy-policy/ (accessed on 31 July 2022)).

We introduced eight paragraphs to the system (the number of paragraphs is equal to the number of line breaks). In the following, we present the results of generating information elements:

- Results of the BART method: Figure 7 shows the results of generating information elements using the BART transformer. Detailed observations of the visualization using BART: The system generated eight information elements. We noted that it only kept the paragraphs containing information elements. As in the previous case study, we also noticed that the information elements are a bit longer than what we planned. The current visualization cannot contain long information elements, so we displayed the information elements as IDs (labels).
- Results of the LSTM method: Figure 8 shows the results of generating information elements using the LSTM network.
- Detailed observations of the visualization using LSTM: The system generated four information elements despite the fact that the same text was introduced for both

models. This is due to the underfitting problem we encountered with LSTM. The problem is in the amount of data used to train the model. Therefore, the algorithm can make accurate predictions, but the initial assumption about the data is incorrect. Training this architecture on more data will improve the results.

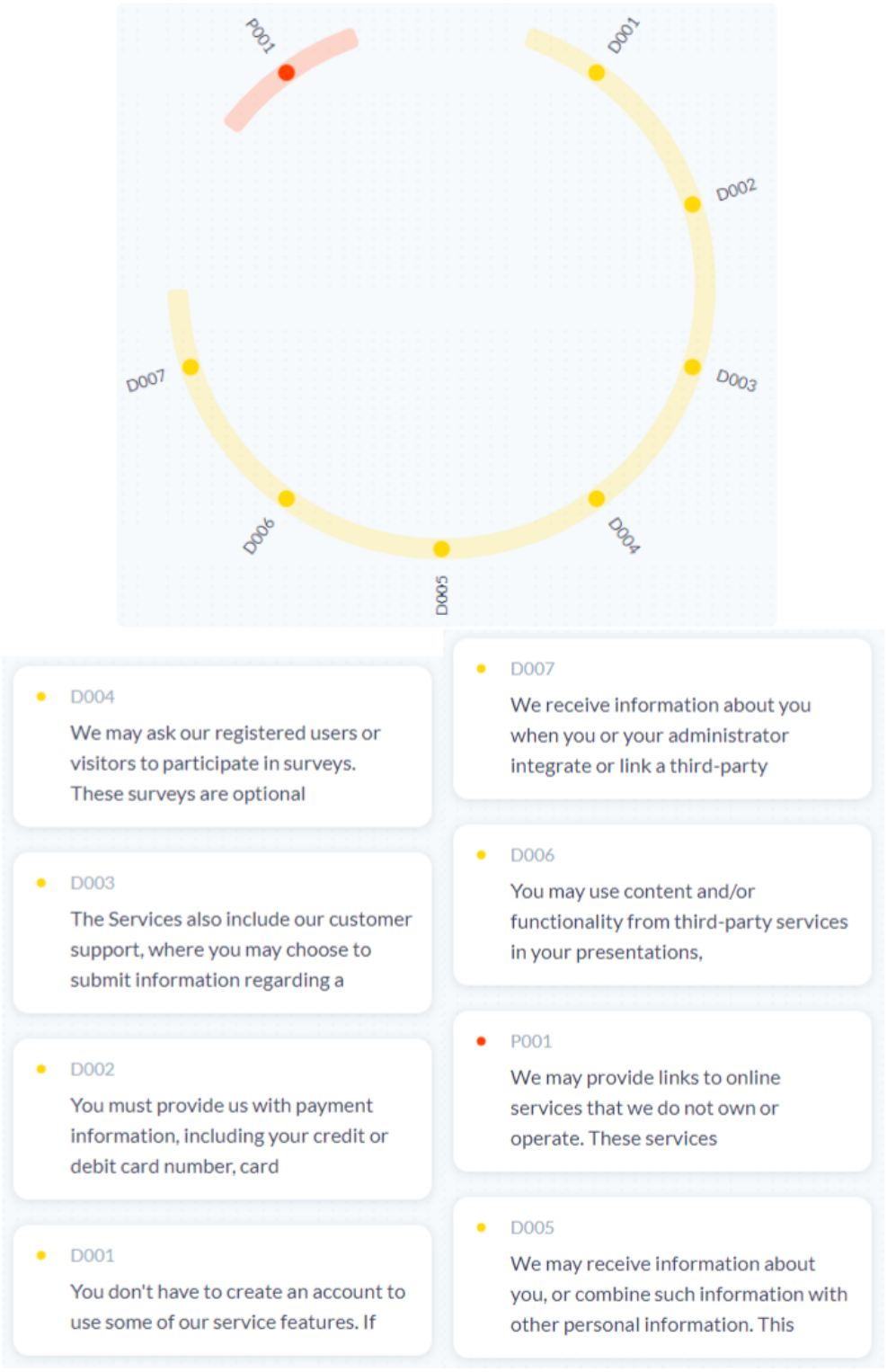

Figure 7. BART results of Prezi policy. Eight information elements have been extracted: seven for data (yellow) and one for process (red).

Figure 8. LSTM result of Prezi policy. Four information elements have been extracted: three for data (yellow) and one for process (red).

BART and LSTM Models Comparison

In the following, we compare the two prediction methods based on the observations obtained from the tests performed in the previous step. The main difference between the LSTM architecture we built layer by layer and the transformer-based model built by Facebook is in the amount of data used for training. We used a dataset approved by an expert with a size of 1000 pairs divided between training and validation. On the other hand, the BART transformer was trained and fine-tuned on the CNN Daily Mail (https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail (accessed on 31 July 2022)) dataset which contains 286,817 training pairs, 13,368 validation pairs, and 11,487 test pairs (total: 311,672). This dataset is very large and rich compared to the dataset we manually created for this project. Natural language processing models like LSTMs require much more data to give concrete results. Therefore, we propose to increase the data to improve the results in the future.

Linjordet et al. [40] conducted a study on the effect of training data size on LSTM model performance. The models they studied were trained with the same input parameters and different training data sizes to show the effect of data size on the prediction performance. After briefly examining the effects of dataset size on deep neural networks, the consequences of the logarithmic reduction in available training data (10 percent vs. 100 percent) mainly indicate a failure in the model's ability to generalize.

Finally, although this LSTM model did not give us the expected results, we consider it a "proof of concept"; that is, we prepared the necessary architecture for the client and proved that the solution gives acceptable results with less data. Furthermore, if the client wants to improve the result, they just need to train the model with more data using the existing architecture. In Table 5, we summarize the results of our study. We observed that the BART model contains a larger dataset compared to the LSTM model, which allows it to have a higher generalization capacity. However, as mentioned earlier, during the visualization in **TRANSPVIS**, we noticed that the BART model generates longer texts compared to the LSTM model. This necessitates the use of IDs for information elements instead of the original text. Additionally, in this scenario, the expert has the ability to update and make the texts more concise. Underfitting occurs in the LSTM model due to a dataset size that needs to be increased.

Table 5. Comparison between the results of machine learning models.

Metric	BART Model	LSTM Model
Dataset size	311,672 pairs	1000 pairs
Generalization capacity	Good	Poor
Summary length	Long	Short
Underfitting (high bias, low variance)	No	Yes
Overfitting (low bias, high variance)	No	No

6.3. Discussion

The use of natural language processing (NLP) or artificial intelligence (AI) in transparency tasks has a fascinating effect. Transparency is becoming an increasingly important non-functional requirement for information systems (for example, the cost-of-living crisis in England has led many people to demand more transparency from the energy sector), but at the same time, it has always been the job of journalists and investigators to read several pages of transparency documents to provide this information to the public. NLP and AI with machine learning can therefore play a crucial role in making transparency more easily and directly accessible to everyone. On the other hand, visualization is essential for communicating information to people to facilitate the understanding of different aspects of transparency. **TranspVis** has brought all these elements together (i.e., the use of NLP as well as the clear visualization of transparency elements). Having a precision of up to 0.50 for close classes and small datasets provides encouraging signs to further explore NLP in the context of software conditions or policies. However, the recall values show some limitations due to data balance, size, and similarity, which can be improved in the future. One approach to enhance the recall is to incorporate more data or utilize more techniques for obtaining additional data and validation. Expert evaluations indicate that **TranspVis** enables insights into transparency requirements in software conditions and benefits a large number of users.

7. Conclusions

In this study, we have developed a visual analytics prototype that combines the **TranspVis** visualization tool with machine learning and natural language processing techniques to address transparency requirements. The proposed framework consists of two classification stages, where the naïve Bayes classifier is utilized. The first stage aims to determine the presence of IEs in transparency documents, while the second stage focuses on identifying the specific types of IEs, such as data, process, and policies. To handle the issue of text length, we have incorporated text summarization techniques, namely BART and LSTM, which effectively reduce the size of the text. The results obtained from our experiments are highly promising, highlighting the potential of our approach, with binary classification achieving an accuracy of 0.70 and multi-class classification achieving an accuracy of 0.62. The improved **TranspVis** tool simplifies the process of identifying the three classes directly without the need to refer to software policies.

8. Limitations and Future Works

This work has focused on the extraction of information elements, specifically into three classes: data, policy, and process. However, future research should consider the inclusion of transparency actors to visualize the flow of information among relevant stakeholders. It is important to note that limitations were encountered due to the size of the available dataset. To overcome this, we have outlined plans to expand the dataset and seek validation from additional experts in transparency analytics. Additionally, the application of machine learning models holds promise for extracting relationships from the data. Overall, this study serves as an initial step towards gaining insights into software policies and conditions,

providing a solid foundation for further investigations into a broader range of classes and relationships.

Author Contributions: Conceptualization, S.F. and K.B.; data curation, S.F. and M.H.; methodology, S.M.; software, K.B.; supervision, S.M. and M.H.; validation, S.F., M.H., K.C. and S.M.; writing—original draft, S.F. and K.C.; writing—review and editing, S.F., M.H., S.M. and K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R196), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R196), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hosseini, M.; Shahri, A.; Phalp, K.; Ali, R. Engineering transparency requirements: A modelling and analysis framework. *Inf. Syst.* **2018**, *74*, 3–22. [CrossRef]
2. Fadloun, S.; Meshoul, S.; Hosseini, M.; Amokrane, A.; Bennaceur, H. Visualization System for Transparency Requirement Analytics. *Appl. Sci.* **2022**, *12*, 12423. [CrossRef]
3. Pang, S.; Pang, C.; Zhao, L.; Chen, Y.; Su, Z.; Zhou, Y.; Huang, M.; Yang, W.; Lu, H.; Feng, Q. SpineParseNet: Spine parsing for volumetric MR image by a two-stage segmentation framework with semantic image representation. *IEEE Trans. Med. Imaging* **2020**, *40*, 262–273. [CrossRef] [PubMed]
4. Kujur, A.; Raza, Z.; Khan, A.A.; Wechtaisong, C. Data Complexity Based Evaluation of the Model Dependence of Brain MRI Images for Classification of Brain Tumor and Alzheimer's Disease. *IEEE Access* **2022**, *10*, 112117–112133. [CrossRef]
5. Perdios, D.; Vonlanthen, M.; Martinez, F.; Arditi, M.; Thiran, J.P. CNN-based ultrasound image reconstruction for ultrafast displacement tracking. *IEEE Trans. Med. Imaging* **2020**, *40*, 1078–1089. [CrossRef] [PubMed]
6. Khan, A.A.; Faheem, M.; Bashir, R.N.; Wechtaisong, C.; Abbas, M.Z. Internet of things (IoT) assisted context aware fertilizer recommendation. *IEEE Access* **2022**, *10*, 129505–129519. [CrossRef]
7. Choutri, K.; Fadloun, S.; Lagha, M.; Bouzidi, F.; Charef, W. Forest Fire Detection Using IoT Enabled UAV and Computer Vision. In Proceedings of the 2022 International Conference on Artificial Intelligence of Things (ICAIoT), Istanbul, Turkey, 29–30 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
8. Zhou, Z.H. *Machine Learning*; Springer Nature: Berlin/Heidelberg, Germany, 2021.
9. Fadloun, S.; Sallaberry, A.; Mercier, A.; Arsevska, E.; Poncelet, P.; Roche, M. Integration of Text-and Web-Mining Results in E pidVis. In Proceedings of the Natural Language Processing and Information Systems: 23rd International Conference on Applications of Natural Language to Information Systems, NLDB 2018, Paris, France, 13–15 June 2018; Proceedings 23; Springer: Berlin/Heidelberg, Germany, 2018; pp. 437–440.
10. Kerren, A.; Stasko, J.T.; Fekete, J.; North, C. (Eds.) *Information Visualization—Human-Centered Issues and Perspectives*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 4950.
11. Yuan, J.; Chen, C.; Yang, W.; Liu, M.; Xia, J.; Liu, S. A survey of visual analytics techniques for machine learning. *Comput. Vis. Media* **2021**, *7*, 3–36. [CrossRef]
12. Fung, A. Infotopia: Unleashing the democratic power of transparency. *Pol. Soc.* **2013**, *41*, 183–212. [CrossRef]
13. Albu, O.B.; Flyverbom, M. Organizational transparency: Conceptualizations, conditions, and consequences. *Bus. Soc.* **2019**, *58*, 268–297. [CrossRef]
14. Hosseini, M.; Shahri, A.; Phalp, K.T.; Ali, R. Transparency as a requirement. In Proceedings of the Joint Proceedings of REFSQ-2015 Workshops, Research Method Track, and Poster Track Colocated with the 21st International Conference on Requirements Engineering, Essen, Germany, 23 March 2015; Foundation for Software Quality (REFSQ): Birmingham, UK, 2015.
15. Hosseini, M.; Shahri, A.; Phalp, K.; Ali, R. Towards engineering transparency as a requirement in socio-technical systems. In Proceedings of the 2015 IEEE 23rd International Requirements Engineering Conference (RE), Ottawa, ON, Canada, 24–28 August 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 268–273.
16. Bannister, F.; Connolly, R. The trouble with transparency: A critical review of openness in e-government. *Policy Internet* **2011**, *3*, 1–30. [CrossRef]
17. Keim, D.; Andrienko, G.; Fekete, J.D.; Görg, C.; Kohlhammer, J.; Melançon, G. Visual analytics: Definition, process, and challenges. In *Information Visualization*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 154–175.

18. Fadloun, S.; Meshoul, S.; Choutri, K. CircleVis: A Visualization Tool for Circular Labeling Arrangements and Overlap Removal. *Appl. Sci.* **2022**, *12*, 11390. [CrossRef]
19. Chatzimparmpas, A.; Martins, R.M.; Jusufi, I.; Kucher, K.; Rossi, F.; Kerren, A. The State of the Art in Enhancing Trust in Machine Learning Models with the Use of Visualizations. In *Computer Graphics Forum*; John Wiley Sons: Hoboken, NJ, USA, 2020. [CrossRef]
20. Fadloun, S.; Morakeb, Y.; Cuenca, E.; Choutri, K. TrajectoryVis: A visual approach to explore movement trajectories. *Soc. Netw. Anal. Min.* **2022**, *12*, 53. [CrossRef] [PubMed]
21. Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-trained models for natural language processing: A survey. *Sci. China Technol. Sci.* **2020**, *63*, 1872–1897. [CrossRef]
22. Shen, L.; Shen, E.; Luo, Y.; Yang, X.; Hu, X.; Zhang, X.; Tai, Z.; Wang, J. Towards Natural Language Interfaces for Data Visualization: A Survey. *IEEE Trans. Vis. Comput. Graph.* **2021**, *29*, 3121–3144. [CrossRef] [PubMed]
23. Lavigne, V.; Gouin, D. Visual analytics for cyber security and intelligence. *J. Def. Model. Simul.* **2014**, *11*, 175–199. [CrossRef]
24. Zhang, Y.; Zhang, J.; Zhang, B. Visual analysis of cybersecurity situational awareness. In Proceedings of the 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 18–20 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 685–688.
25. Wang, X.; Chen, W.; Chou, J.; Bryan, C.; Guan, H.; Chen, W.; Pan, R.; Ma, K. GraphProtector: A Visual Interface for Employing and Assessing Multiple Privacy Preserving Graph Algorithms. *IEEE Trans. Vis. Comput. Graph.* **2019**, *25*, 193–203. [CrossRef] [PubMed]
26. DeHart, J.; Stell, M.; Grant, C. Social Media and the Scourge of Visual Privacy. *Information* **2020**, *11*, 57. [CrossRef]
27. Chou, J.K.; Wang, Y.; Ma, K.L. Privacy preserving visualization: A study on event sequence data. In *Computer Graphics Forum*; John Wiley Sons: Hoboken, NJ, USA, 2019; Volume 38, pp. 340–355.
28. Ghazinour, K.; Majedi, M.; Barker, K. A model for privacy policy visualization. In Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference, Seattle, WA, USA, 20–24 July 2009; IEEE: Piscataway, NJ, USA, 2009; Volume 2, pp. 335–340.
29. Ghazinour, K.; Albalawi, T. A usability study on the privacy policy visualization model. In Proceedings of the 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Auckland, New Zealand, 8–12 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 578–585.
30. Jutla, D.N.; Bodorik, P.; Ali, S. Engineering privacy for big data apps with the unified modeling language. In Proceedings of the 2013 IEEE international congress on big data, Santa Clara, CA, USA, 27 June–2 July 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 38–45.
31. Schonlau, M.; Guenther, N. Text mining using n-grams. *Stata J.* **2017**, *17*, 866–881. [CrossRef]
32. Karabiber, F. TF-IDF, Term Frequency-Inverse Document Frequency. 2020. Available online: https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/ (accessed on 31 July 2022).
33. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7871–7880.
34. De Bruyn, M.; Lotfi, E.; Buhmann, J.; Daelemans, W. BART for Knowledge Grounded Conversations. In Proceedings of the KDD Workshop on Conversational Systems Towards Mainstream Adoption (KDD Converse' 20), San Diego, CA, USA, 24 August 2020; Volume 2666.
35. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef] [PubMed]
36. Patel, R.M.; Goswami, A.J. Abstractive Text Summarization with LSTM using Beam Search Inference Phase Decoder and Attention Mechanism. In Proceedings of the 2021 International Conference on Communication, Control and Information Sciences (ICCISc), Idukki, India, 16–18 June 2021; IEEE: Piscataway, NJ, USA, 2021; Volume 1, pp. 1–6.
37. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1, pp. 4171–4186.
38. Angela, F.; David, G.; Michael, A. Controllable Abstractive Summarization. In Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, Melbourne, Australia, 15–20 July 2018; pp. 45–54.
39. Murphy, K.P. *Naive Bayes Classifiers*; University of British Columbia: Vancouver, BC, Canada, 2006; Volume 18, pp. 1–8.
40. Linjordet, T.; Balog, K. Impact of training dataset size on neural answer selection models. In Proceedings of the Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, 14–18 April 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 828–835.