sustainability

MDPI

Article

Data-Driven Analysis of Privacy Policies Using LexRank and KL Summarizer for Environmental Sustainability

Abdul Quadir Md [1,*], Raghav V. Anand [1], Senthilkumar Mohan [2], Christy Jackson Joshua [1], Sabhari S. Girish [1], Anthra Devarajan [1] and Celestine Iwendi [3]

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
[2] School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, India
[3] School of Creative Technologies, University of Bolton, Bolton BL3 5AB, UK
* Correspondence: abdulquadir.md@vit.ac.in

Abstract: Natural language processing (NLP) is a field in machine learning that analyses and manipulate huge amounts of data and generates human language. There are a variety of applications of NLP such as sentiment analysis, text summarization, spam filtering, language translation, etc. Since privacy documents are important and legal, they play a vital part in any agreement. These documents are very long, but the important points still have to be read thoroughly. Customers might not have the necessary time or the knowledge to understand all the complexities of a privacy policy document. In this context, this paper proposes an optimal model to summarize the privacy policy in the best possible way. The methodology of text summarization is the process where the summaries from the original huge text are extracted without losing any vital information. Using the proposed idea of a common word reduction process combined with natural language processing algorithms, this paper extracts the sentences in the privacy policy document that hold high weightage and displays them to the customer, and it can save the customer's time from reading through the entire policy while also providing the customers with only the important lines that they need to know before signing the document. The proposed method uses two different extractive text summarization algorithms, namely LexRank and Kullback Leibler (KL) Summarizer, to summarize the obtained text. According to the results, the summarized sentences obtained via the common word reduction process and text summarization algorithms were more significant than the raw privacy policy text. The introduction of this novel methodology helps to find certain important common words used in a particular sector to a greater depth, thus allowing more in-depth study of a privacy policy. Using the common word reduction process, the sentences were reduced by 14.63%, and by applying extractive NLP algorithms, significant sentences were obtained. The results after applying NLP algorithms showed a 191.52% increase in the repetition of common words in each sentence using the KL summarizer algorithm, while the LexRank algorithm showed a 361.01% increase in the repetition of common words. This implies that common words play a large role in determining a sector's privacy policies, making our proposed method a real-world solution for environmental sustainability.

Keywords: natural language processing; privacy; text summarization; high weightage; common word reduction; LexRank; Kullback Leibler summarizer

check for updates

Citation: Md, A.Q.; Anand, R.V.; Mohan, S.; Joshua, C.J.; Girish, S.S.; Devarajan, A.; Iwendi, C. Data-Driven Analysis of Privacy Policies Using LexRank and KL Summarizer for Environmental Sustainability. *Sustainability* **2023**, *15*, 5941. https://doi.org/10.3390/su15075941

Academic Editor: Andreas Kanavos

Received: 6 February 2023
Revised: 7 March 2023
Accepted: 12 March 2023
Published: 29 March 2023

1. Introduction

As technology keeps advancing, the amount of data generated every day keeps increasing exponentially. Approximately around 2.5 quintillion bytes of data are generated every day, which is a vast quantity. Owing to the introduction of artificial intelligence and its subset of machine learning, we can now process large chunks of data in a much faster and more efficient way. Natural language processing (NLP) is a field in machine learning that analyzes and manipulates huge amounts of data, and potentially generates human language. There are a variety of applications of NLP like sentiment analysis, text summarization, spam filtering, language translation, and much more.

To explore the field of natural language processing, a thematic evolution of the same was done to evaluate the growth of using techniques to handle and manipulate data [1,2]. The plot in Figure 1 shows the timeline of how NLP developed over time from its initial stages.

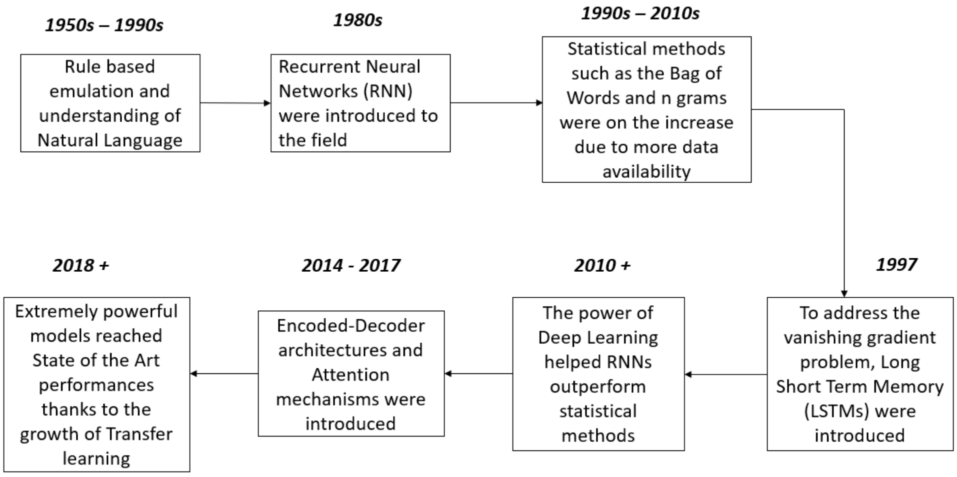
1950s – 1990s
Rule based emulation and understanding of Natural Language
1980s
Recurrent Neural Networks (RNN) were introduced to the field
1990s – 2010s
Statistical methods such as the Bag of Words and n grams were on the increase due to more data availability
1997
To address the vanishing gradient problem, Long Short Term Memory (LSTMs) were introduced
2010 +
The power of Deep Learning helped RNNs outperform statistical methods
2014 - 2017
Encoded-Decoder architectures and Attention mechanisms were introduced
2018 +
Extremely powerful models reached State of the Art performances thanks to the growth of Transfer learning

Figure 1. Evolution of NLP.

In this paper, we put forward a novel approach to implementing natural language processing on the privacy policies of various companies that extracts the important information that must be obligatorily read by any customer before accepting the terms and conditions. A privacy policy is a legal document that must be read by every customer before they accept the terms of the company, be it downloading software in a system or agreeing to buy a house, or any agreement for that matter, it is important to peruse the documents thoroughly before accepting the terms. However, when it comes to reading the privacy policy, it consists of hundreds and thousands of words, out of which every sentence may not be important, though they are part of the terms. The majority of people skip reading the privacy policy because it consists of a large number of words, and it would be too time-consuming to read the entire document.

According to a survey conducted by Statista in the year 2013, with a thousand respondents, only 9.7% of the respondents read the entire privacy policy document and 29.4% of the population never read the privacy policy document. Another survey carried out in the year 2008 with respect to privacy policy documents stated that people take around 10 min to read a medium-sized privacy policy containing 2514 words on average at a rate of 250 words per minute, and people tend to get the task done without reading through the entire policy. They also stated that US internet users visit 66 unique sites per month at work and 119 unique sites at home and it is a known fact that we usually encounter a privacy policy or cookie policy whenever we visit a new website. This positions us in danger since we are used to accepting all sorts of privacy policies over the internet. It is important to know certain key sentences that hold the maximum weightage in a privacy policy that may be of cardinal importance, and the customers must skim through those lines correctly before accepting the terms and conditions. Therefore, using web scraping of privacy policy data and natural language processing algorithms, we plan to extract the sentences in the privacy policy document that is of high importance and display them to the customer so that it can save the customer's time, while also providing them with only the important lines that they need to know before signing the document. Certain issues came up while doing the web scraping process. The main objective of the process is to obtain various important common words from various privacy policies belonging to a particular industry or sector. Unnecessary non-important short words like "as", "the", etc., were also added to the common word list obtained from various sources. Auxiliary verbs like "am"," shall", etc., were also another concern. Since the main purpose of the common word reduction process is to extract the common important words from various privacy policies, these words were removed by using a dictionary called contraction_mapping. This dictionary contains auxiliary verbs and is checked with the common word list to see if the words in the dictionary are presented in them. If they are found, then they are removed from the list, and the words also pass through an algorithm where words of a length less than or equal to three are removed. This preprocessing step makes sure that no non-important words are retained after the common word reduction process is complete. The second section of the paper talks about the various related work done in the field of text summarization in the privacy policy. The third section talks about the proposed work and the various algorithms used to get the summarized text, followed by the fourth section that explains the experimental setup and the results obtained, after which the final section concludes the paper with future work and directions.

2. Literature Survey

There are plenty of works done in the field of NLP and its various divisions. This section discusses the different, existing, and improvised works regarding natural language processing and focusses on the aspect of text summarization and its methods, which are the points to focus in this paper. In [3], the authors used machine learning algorithms such as a support vector machine (SVM), a Naive Bayes classifier, a Hopfield network algorithm, and a few more to classify the sentiments in the texts. All the above-mentioned algorithms either make the entire data as clusters or use gradient descent to approximately map the particular text to one cluster to classify the sentiment of the text. They have also noticed that the results generated by techniques of machine learning are more improved and precise in comparison to earlier techniques like human-generated baselines. Comparing both the techniques of machine learning, the SVM model's results are better than the Naive Bayes classifier, although the difference was not very large, thereby helping us to efficiently classify people's emotions like anger, happiness, sadness, and sarcasm through texts. The authors in [4–7] came up with a solution to retrieve and efficiently summarize valuable information using natural language recognition and natural language generation. They proposed the use of a bi-directional long short-term algorithm (LSTM) through the method of Elmo embedding which captures the context-dependent aspects of the word meaning and aspects of the word. This method involves converting raw text into vectors which helps in improving accuracy, so more focus can be put into extractive text summarization than abstractive text summarization. The disadvantage of this procedure is that the preprocessing technique Elmo embedding is hard to implement and the title for the documents has not been used for extensive text retrieval. In [8], the authors put forwards a model that uses two important concepts: human domain experts, where keywords are provided, and the knowledge that is automatically learned by the machine learning model. The authors have used an ensemble approach to give a solution to the problem where they have classified important sentences based on "statement categories". The keywords are manually crafted into several patterns for five statement categories in the domain of privacy notices. Repeated incremental pruning to produce error peduction (RIPPER), a classification algorithm, was used to classify the sentences into one of those five categories [9]. Then the component on the next level called the "Combiner Classifier" obtained the output of the two level-1 matches (keyword classification and RIPPER algorithm) and produced the final result. It was seen that 6 out of 10 sentences were classified correctly and the precision and accuracy of this approach were decent. Due to considerable error, the model needed more refinement to give a better accuracy.

The different techniques for automatic text summarization have been analyzed and reviewed. Different preprocessing techniques like segmentation of chapter, paragraph, sentence, word, and lemmatization, term weight, word frequency, sentence by term matrix, sentence selection, normalize, post tagging, proper noun set, and bag-of-words were enacted, and then features were used which indicated the text to be extracted. After the initial stage, they moved on to use different techniques, namely fuzzy-based, machine learning, statistics, graphics, topic modelling, and rule-based. Since various papers were analysed, it was easier for them to figure out which algorithm would give a better result and how to proceed with the entire research. It also gave them a fair idea about the difficulties faced by the people that they were searching for, which helped them figure out solutions to the problems [10–14]. The ease and efficiency in the extraction of information through the analysis of extractive and abstract summarization are facilitated. The primary steps followed included identification, interpretation, and summary generation, while the methods utilized extend to the cluster method, a machine learning approach, fuzzy logic, and so on. As far as extractive text summarization is concerned, the two-step pre-processing and processing step was where the features influencing the relevance of the sentence was decided. The advantage of this is that the conclusion was drawn through the analysis of different methods, but it did not cover the ambit of hybrid methods. It was effectively concluded that the summarization which uses graph-theoretic, neural networks, fuzzy logic, and cluster has an effective summary [15]. In [16–18], the authors highlight the statement of the problem as pertaining to the rise in data and the concerns of high storage space requirements. A suggested alternative was to summarize the data which would enable the people to read easily, as well as take up less storage space. This mandated a model which would condense the text data into a compact format. The researchers thus proposed SpaCy—the open-source library of python, a quintessential process in the commencing phase of the extractive approach—natural language processing [19–21]. The initial step was that the text was preprocessed through the SpaCy library wherein the words and subsequent sentences were tokenized. Henceforth, any machine learning algorithm like a recurrent neural network (RNN) was used. This model drew inspiration from Natural Language Toolkit (NLTK) and remedies the issue of time delays by utilizing python's library space and time efficiency. By combining the best approaches, it is well-rounded and object-oriented. However, the emphasis on speed came at the sacrifice of accuracy which might have affected the summarized text. In [22–24], the authors present a clear understanding of the pros and cons of the semantic- and structure-based approaches for abstractive text summarization, extractive text summarization techniques, and performance-based comparisons for existing summarizers. It is concluded that abstractive text summarization requires more learning and analysis because it is more complex but proves to be better than the extractive approach as it provides a more appropriate and meaningful summary of the text. It is also noted that less research work is done in the domain of abstractive text summarization on Indian languages and hence, there is more scope for exploration of better techniques.

An automated text generation system that generates short but valuable sentences from lengthy judgments is proposed. Based on the input type of case- civil or criminal, two approaches are followed namely single document untrained approach and multi-document trained approach. Both of these use a technique called latent semantic analysis (LSA). The results were partially assessed using Recall-Oriented Understudy for Gisting Evaluation (ROGUE) and the results were accurate. This model was even approved by professional lawyers as well. The authors have also identified an issue of continuity with the LSA technique and also base their model on entire concepts and not just similar words [25]. In [26], the authors have introduced an automated summarization evaluation (ASE) model that is highly conditional to the features of the source text, thereby giving a text-based model of quality [27]. Different approaches have been used for the ASE model to evaluate quality including the overlap of n-grams between source and summaries, examining the lexical and semantic overlap between source texts and their summaries, and the use of rhetorical devices in summaries including connectives. One other approach that was looked into was the LSA method. The NLP tools that were used to collect different information like lexical sophistication, syntactic complexity, etc., are the Tool for the Automatic Analysis of Lexical Sophistication (TAALES), Tools for the Automatic Analysis of Text Cohesion (TAACO), the Tool for the Automatic Analysis of Lexical Diversity (TAALED), and the Tool for the Automatic Analysis of Syntactic Sophistication and Complexity (TAASSC). After a thorough analysis of different approaches, the results proved to be accurate. Similarly, [28] has also analyzed the various text summarization algorithms. The authors of [29] have analyzed algorithms separately and in combination with one other. They have done a check and comparison for accuracy using other concise and better summaries and use ROGUE and Term Frequency–Inverse Document Frequency (TF_IDF) for this purpose. After analysis, it was concluded that the summaries obtained from using different algorithms were not very accurate and the study is still ongoing. Some suggestions of using the generative adversarial network (GAN's) model [30] and transfer learning were put forth. Moreover, the authors [31] discussed many issues related to recent works on natural language processing and security.

3. Problem Statement

Privacy documents are important and legal, so they are an essential element of any agreement. Despite the length of these documents, the important points still need to be read thoroughly. It might be difficult for a customer to understand all the details of a privacy policy if they lack time and knowledge. In this frame of mind, a novel approach to optimally select the important sentences of the privacy policies, defined in this paper as the Common Word Reduction (CWR) process, forms the main backbone of the architecture. The application of this process helps to find commonly used words in a particular sector's privacy policy, which are rendered important because of its frequent usage in such documents. Extractive text summarization techniques—the Kullback–Leibler algorithm and the LexRank algorithm—have been applied on top of the above-obtained sentences to display the sentences that carry heavy weightage, consequently minimizing the total count of the sentences in the privacy policy.

4. Materials and Methods

Data preprocessing is the first and foremost process in any aspect of machine learning and natural language processing models. Since there is no raw text containing a purely privacy policy dataset, the scraping of the data from different websites is introduced to clear all the Hypertext Markup Language (HTML) tags of the particular web pages to get clear data of privacy policy, following which the CWR process described earlier is applied to the data to optimize the privacy policy summary. KL summarizer and LexRank algorithms are used to further optimize it. The framework established for the process is described in the following sub-sections.

4.1. Common Word Reduction Process

The dataset used to formulate the CWR process contains the links of the websites containing various policies. The algorithm that is introduced will first web scrape data from various privacy policies from a chosen industry, remove punctuation marks and special characters, and bring out all the important words related to the privacy policy using the NLTK library. The data used for this purpose help to know the important words that are typically meant to be used in the privacy policy documents about the chosen sector, which essentially is the end-goal optimization of this work.

4.2. Text Summarization

After the completion of the common word reduction process on the training data, the frequently stated common words that occur in sentences are extracted and stored. This is further used for extracting sentences that contain these words from the privacy policies of the testing data. After the important sentences are extracted, the extractive text summarization algorithms are applied on top of this to further reduce the summary optimally, and the important sentences of the privacy policy as part of the test data are obtained.

4.3. Common Word Reduction Process

4.3.1. Kullback–Leibler Algorithm

This extractive text summarization technique uses a greedy approach to make a summary of the sentences. As long as the divergence is low, the sentences are added to the summary and once the sentence count reaches the limit set by the user, the process stops. The algorithm is efficient when there is less divergence in the sentences of the document, and consequently, the summary of the privacy policy produced will be more.

Let P and Q be probability distributions. The purpose of KL divergence is to find the divergence between the probabilities of P and Q given in Equation (1):

$$KL\ Divergence(P, Q) = \sum_{i=1}^{n} \left[P_i \times loglog\left(\frac{Pi}{Qi}\right) \right] \tag{1}$$

where $P = (x1 : P1, x2 : P2, x3 : P3 \ldots)$ with $\sum Pi = 1$ and $Q = (y1 : Q1, y2 : Q2, y3 : Q3 \ldots)$ with $\sum Qi = 1$.

The KL divergence value always has a value greater than or equal to zero.

Different forms of the divergence when there is the transformation from variable x to variable $y(x)$ can be represented using the following Equation (2):

$$\begin{aligned} D_{KL}(P\|Q) &= \int_{x_a}^{x_b} P(x) log\left(\frac{P(x)}{Q(x)}\right) dx = \int_{y_a}^{y_b} P(y) log\left(\frac{P(y)\frac{dy}{dx}}{Q(y)\frac{dy}{dx}}\right) dy \\ &= \int_{y_a}^{y_b} P(y)\ \ log\left(\frac{P(y)}{Q(y)}\right) dy \end{aligned} \tag{2}$$

4.3.2. LexRank Summarization Algorithm

The LexRank extractive text summarization technique follows a graph-based approach to summarising texts. It is an unsupervised method that uses the concept of eigenvector centrality in graphs for a given representation of sentences.

Based on the eigenvector centrality that is calculated, the sentences are placed on the vertices of the graph and the weights of each edge are calculated using the cosine similarity function.

The computation of the cosine similarity function is done in a way that for each word occurring in a sentence, the corresponding value in the matrix representation is the count of a particular word in the sentence multiplied by the inverse document frequency of the word, which is calculated using the following Equation (3):

$$idf - modified - cosine(x, y) = \frac{\Sigma_{w \in x,y}\ tf_{w,x} tf_{w,y} (idf_w)^2}{\sqrt{\Sigma_{x_i \in x}\ (tf_{x_i,x} idf_{x_i})^2} \times \sqrt{\Sigma_{y_i \in y}\ (tf_{y_i,y} idf_{y_i})^2}} \tag{3}$$

On applying the above two extractive summarization techniques after the common word-reduction process, an analysis of the sentences obtained, with respect to the common words obtained from various sources of the training data, is made with respect to each algorithm to prove the significance of the reduction process. Figure 2 represents the architecture of the entire process.

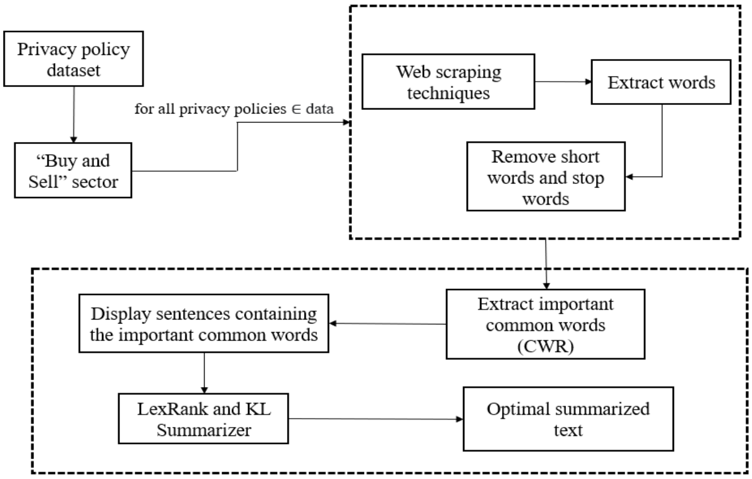
Privacy policy dataset
"Buy and Sell" sector
for all privacy policies ∈ data
Web scraping techniques
Extract words
Remove short words and stop words
Extract important common words (CWR)
Display sentences containing the important common words
LexRank and KL Summarizer
Optimal summarized text

Figure 2. Proposed architecture and workflow of the text summarization process.

4.4. Privacy Policy Dataset

The South Asian privacy policy dataset [32] has been used for performing the common word reduction process which contains the links to various privacy policy websites.

The first column of the dataset contains the specific sector that the website belongs.

The second column contains the sources from where the website links were collected.

The third column specifies the name of the website belonging to the sector.

The fourth column contains the website Uniform Resource Locator (URL) and,

The fifth and final column contains the URL of the privacy policy website that will be used for the common word reduction process.

4.5. Web Scraping Data

On choosing a particular sector to get started from, a web scraping process for each privacy policy belonging to the chosen sector is performed using various NLP libraries such as Natural Language Toolkit (NLTK), BeautifulSoup, and stopwords.

In Algorithm 1, the website link corresponding to a privacy policy of the chosen sector is passed as the parameter to the function and after the completion of decoding of the file, the file is returned as a readable format to the machine for further processing.

Algorithm 1: Open the URL link and decode the Unicode Transformation Format-8 (UTF-8) encoding

```
Step 1    Initialize function with parameter
    Read link of privacy policy ∈ chosen sector
Step 2    Request access
    For each link ∈ passed parameter, do
        file ← Request urlopen from urllib for given link
myfile ← Read file contents (file)
file ← myfile. decode ('UTF-8')
EndFor
Step 3    Return decoded file to the variable that is called the function
```

4.6. Extract Words Present in Each Privacy Policy

After extracting the content from the web for each link in the sector, the returned file after the pre-processing done in Algorithm 1 includes HTML tags, special characters, and other auxiliary verbs that are unnecessary for the common word reduction process. Hence, they are removed using regex and contraction mapping. This is an important step because we need only the important weighted words for the common word reduction process, which will be implemented further.

The algorithm makes use of an in-built package, BeautifulSoup, to preprocess the strings obtained in the decoded file from Algorithm 1 and stores the words whose length is greater than or equal to three. This essentially eliminates unnecessary small words like prepositions and connectives. This process is repeated for every privacy policy website from the chosen sector and the words extracted from each privacy policy are stored in separate lists, with comma as the delimiter.

4.7. Extract Common Words in the Lists and Apply the Common Word Reduction Process

After performing web scraping and the cleaning process on all the privacy policies used for extracting the words, the common words from all these privacy policies are extracted and stored in a separate list, and it will be used for displaying the sentences containing such common words from the testing data containing the privacy policies before being fed to the standard NLP algorithms, i.e., the common word reduction process.

This algorithm extracts the common words present in all the lists after performing Algorithm 1 and Algorithm 2, which contain the words from various privacy policies of the chosen sector, and it is stored separately. The "Summary" list obtained as the end result in the above algorithm contains the sentences which contain the common words present in the stored common word list.

Algorithm 2: Removing the short words, stop words and placing comma as the delimiter to separate each word

```
Step 1    Initializing the function that receives the contents from the previous step
Step 2    Cleaning the content
    Stop_words ← Distinct elements in stopwords.words ("English")
    For each file content ∈ passed parameter, do
        FullString ← Lowercase (file content)
        Modified_String_1 ← BeautifulSoup(FullString, "lxml").text
        Modified_String_2 ← re.sub(r'\([^)]*\)', ", Modified_String_1)
        Modified_String_3 ← re.sub("",", Modified_String_2)    # For removing special characters
        Modified_String_4 ←  Join ' ' with
    ([contraction_mapping[word] if word ∈ contraction_mapping else word for word ∈ newString.split(" ")])
    EndFor
    # Removing auxiliary verbs by having a dictionary named contraction mapping that contains such words
Step 3    Identifying tokens
Tokens ← [words for each word ∈ newString.split() if not words in stop_words]
Step 4    Removing short words
Long_words ← [ ]
For each_word ∈ Tokens:
if len(each_word) >= 3: #removing short word
Long_words.append(each_word)
EndFor
    Return (" ". join(long_words)). strip()
Step 5    Function to place delimiters to separate the words
    for each file content ∈ passed parameter do
        Comma_separated_file ← file.replace(" ", ",")
    EndFor
Step 6    Return comma_separated_file
```

4.8. Applying NLP Algorithms to Optimally Further Display Important Sentences

LexRank algorithm and the KL Summarizer extractive text summarization techniques are applied on top of the weighted sentences that we obtained through the common word reduction process. This helps us to further get more important sentences to be displayed for the customer even though the sentence count gets reduced.

This algorithm applies the extractive NLP summarization techniques to further summarize the sentences extracted after Algorithm 3. In Algorithm 4, to test the significance of the CWR process, Step 5 presents an equation which establishes the importance of the extraction of common words from various privacy policies, and how its presence is important in the summarized document.

Algorithm 3: Extracting common words and using test data to print sentences containing such common words

```
Step 1    Extracting common words
Common_word_set ← Distinct(word_list_1) & Distinct(word_list_2) & . . . Distinct(word_list_n)
Common_word_list ← list(common_word_set)
Step 2    To store sentences from the privacy policy that contain the words (either as a
subset or as a whole) from the common list of words we formed above
Summary ← [ ]
For each index ∈ range (length (privacy_policy_test_sentences)):
For each word ∈ Common_word_list:
if word ∈ privacy_policy_test_sentences[index]:
Summary ← privacy_policy_test_sentences[index])
```

Algorithm 4: Applying extractive text summarization techniques

```
Step 1    Getting the summarized privacy policy sentences combined back from the
Summary list
          Policy_sentences = " "
For each sentence ∈ Summary:
Policy_sentences += sentences
Policy_sentences += "."    # To separate sentences with a full stop
Step 2    Applying the LexRank summarizer algorithm
          From the Sumy summarizers library, import LexRankSummarizer
My_parser ← PlaintextParser.from_string(Policy_sentences,Tokenizer('English))
Lex_rank_summarizer ← LexRankSummarizer()
Lexrank_summary ← Lex_rank_summarizer(my_parser.document,sentences_count=set_count)
Step 3    Applying the KLSummarizer algorithm
          My_parser ← PlaintextParser.from_string(Policy_sentences,Tokenizer('English))
Kl_summarizer ← KLSummarizer()
Kl_summary ← Kl_summarizer(parser.document,sentences_count=set_count)
Step 4    Check the count of the common words present in the final output
          Common_words_count=0
For each i ∈ final_policy_list:
                    For each j ∈ common_word_list:
                              if j ∈ i:
          Common_words_count +=i.count(j)
Step 5    Prove the significance of common words present in the finally obtained
sentences by calculating the percentage increase of the repetition of common words using the
length of the common word list
Percentage increase in usage of common words:
$((Common\_words\_count - Length\ (Common\_word\_list)) \div Length\ (Common\_word\_list)\ ) \times 100$
```

5. Evaluation and Analysis

5.1. Experimental Setup and Results

The experiment is conducted on a windows machine with an Intel-i7 Quadcore processor of 1.50 GigaHertz (GHz) with 8 GB Random Access Memory (RAM). The environment used for this experiment is Visual Studio Code and the web interface of the Jupyter notebook. The privacy policy data is collected from the particular company's website. These website links are available in the dataset where they are classified into each company sector such as Buy and Sell, E-commerce, etc. The excel file is read using the pandas library in Python and the extracted URL is scrapped using the urllib library. Other preprocessing libraries and natural language processing libraries explained in the algorithms section are also required to proceed with further experimentation and analysis of results.

The result of the experiment includes the analysis of:

1. The difference in the number of sentences after applying the common word reduction process on the raw privacy policy text.
2. The difference in the number of sentences after applying the NLP algorithm on top of it.
3. Minimum scale of increase in the repetition of common words.
4. Maximum scale of increase in the repetition of common words.

The last two points of analysis are to prove the significance of the common word reduction process acquired from different sources of data. From the dataset, in this paper, the Buy and Sell sector has been picked and the privacy policy data related to the sector have been trained for the common word reduction process and the results have been analyzed.

The data have been divided into train and test data, where the common word reduction process will be applied to the training data, and after the extraction of the common words, the reduction process is applied to the test data followed by the application of the extractive text summarization techniques. The privacy policy website links of Ebay, OLX, Limeroad, and GroupOn were used for the training process, i.e., the application of preprocessing algorithms and CWR, while EleB2B, GoCoop were used as testing data. Various common words obtained from the various data sources include 'right', 'products', 'data', 'promotion', 'email', 'take', 'used', 'provide', 'send', 'disclose', 'orders', 'changes', and so on. A total of 59 such important common keywords were collected from the training data. After implementing the common word reduction process, the important sentences of the privacy policy are still retained, while removing relatively less important sentences that can be skipped by the user when going through the privacy policy. The sentence count parameter of both the extractive text summarization algorithms was set as 15.

5.2. Evaluation Metrics

For the common word reduction process, the percentage decrease in the count of sentences before and after applying the process is taken as the evaluation metric, while the count of the common words in the privacy policies and the corresponding percentage increase in the repetition of those common words in the policies have been taken as the metrics mentioned in Equations (4) and (5).

Percentage decrease in the count of sentences after the CWR process:

$$\frac{(InitialSentencesCount - CWReducedSentencesCount)}{InitialSentencesCount} \times 100 \tag{4}$$

Percentage increase in the repetition of common words:

$$\frac{(CommonWordsCount - Length(CommonWordList))}{Length(CommonWordList)} \times 100 \tag{5}$$

5.3. Performance Metrics and Comparison of the Algorithms

The privacy policy of EleB2B showed a maximum reduction in sentences with a 14.63% reduction in the sentence count, followed by Rediff with an 11.86% decrease, and, finally, GoCoop with a 7.93% decrease in the total count of the sentences, as shown in Table 1. The grouped bar plot in Figure 2 shows the sentence count results in the graphical format.

The plot in Figure 3 shows the significance of the CWR by measuring the increase in the repetition of the common words obtained in the test data after applying the extractive text summarization techniques.

The results indicate that in the privacy policy of Rediff, the LexRank algorithm showed a 315.25% increase in the repetition of common words and the KL summarizer showed a 94.91% increase in the same. In the GoCoop privacy policy, there was a 361.01% increase in repetition in the LexRank algorithm in comparison to a 161.01% increase in the KL summarizer technique, and the line plot in Figure 4's visualization of the results in the table compares each algorithm. It is evident from the data and the graph that the LexRank algorithm gives more importance to the common words, even though those words were obtained from various privacy policies. This result highlights the significance of CWR.

Table 1. Comparison of sentence count before and after the common word reduction process.

Test Data	Sentence Count		Percentage Decrease
	Initial	**After CWR Process**	
Rediff	59	52	11.86%
EleB2B	41	35	14.63%
GoCoop	63	58	7.93%

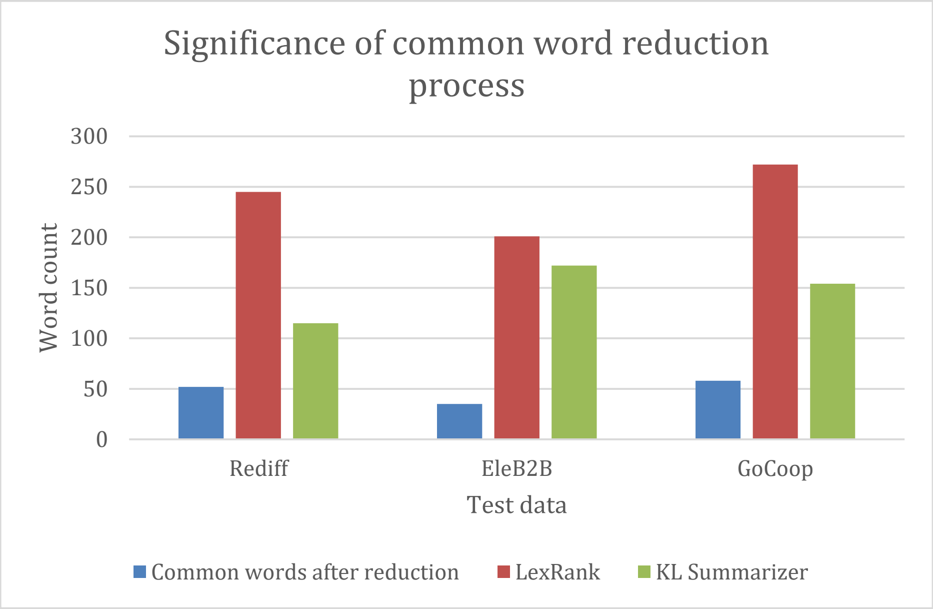

Figure 3. Significance of common word reduction process with different policies and algorithms.

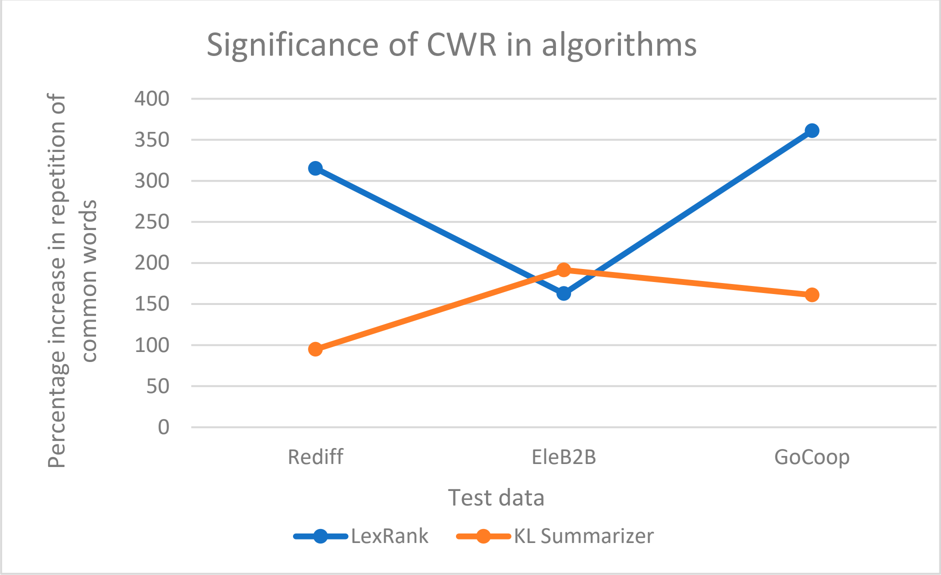

Figure 4. Comparison of common word significance between different algorithms and policies.

6. Discussion and Future Work

In this paper, different privacy policies from a sector Buy and Sell were analyzed from their respective URL links. Information was extracted from them and summarized.

Common words from different privacy policies were initially extracted and the output of the sentences after the common word reduction process were analyzed and the results were discussed. The introduced CWR methodology adds novelty to the existing summarization techniques compared to various works by analyzing the important common words occurring in various privacy policies of a particular sector, and it is implemented to initially extract a summarized text. Two extractive text summarization algorithms, namely LexRank and KL Summarizer were used on top of the previously extracted summarized text to obtain a further reduced well highlighted summary. With respect to the evaluation metrics, the usually implemented metrics are concerned with precision scores, recall, F scores, and the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric, as surveyed in other works. This paper introduces a new metric to understand the significance of the sentences present in the obtained summary. The measurement of the increase in the repetition of common important words obtained through the CWR methodology highlights the importance of the process.

It was inferred that the sentences obtained after applying the CWR process are more significant when compared to giving the privacy policy text to the summarization algorithms in raw form. There was a minimum of 7.93% reduction of sentences and a maximum of 14.63% reduction of sentences using the CWR process, and by applying extractive NLP algorithms on top of that, more significant sentences in a summarized form were obtained. The results after applying NLP algorithms show that there is a minimum of 94.91% increase in the repetition of common words in each sentence in the KL summarizer algorithm and a maximum of 191.52% increase in repetition, while the LexRank algorithm showed a minimum of 162.7% increase in the repetition of common words and a maximum repetition of 361.01%, which emphasizes the significant role common words obtained in a sector's privacy policies have.

The algorithms have currently been experimented on a single sector. As part of a future work, this can be expanded across many sectors and an interface can be created that allows the user to choose to read a privacy policy from a particular sector and get a summarized form of results on a single click, that can essentially save their time of reading through the entire document.

Author Contributions: Conceptualization, A.Q.M., R.V.A., C.J.J. and S.M.; Methodology, A.Q.M., R.V.A., S.S.G. and A.D.; Formal analysis, R.V.A. and S.M.; Investigation, R.V.A., S.S.G. and A.D.; Resources and data collection, S.S.G., A.D. and C.I.; Writing; A.Q.M. and R.V.A. Validation, S.M., C.J.J. and C.I. Funding Acquisition, A.Q.M. All authors have read and agreed to the published version of the manuscript.

Funding: The Article Processing Charges for this research is funded by the Vellore Institute of Technology, Chennai, India.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: All authors acknowledge the Vellore Institute of Technology, Chennai, India.

Conflicts of Interest: The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Sott, M.K.; Nascimento, L.D.S.; Foguesatto, C.R.; Furstenau, L.B.; Faccin, K.; Zawislak, P.A.; Mellado, B.; Kong, J.D.; Bragazzi, N.L. A Bibliometric Network Analysis of Recent Publications on Digital Agriculture to Depict Strategic Themes and Evolution Structure. *Sensors* **2021**, *21*, 7889. [CrossRef] [PubMed]
2. Belfiore, A.; Cuccurullo, C.; Aria, M. IoT in healthcare: A scientometric analysis. *Technol. Forecast. Soc. Change* **2022**, *184*, 122001. [CrossRef]
3. Gupta, P.; Tiwari, R.; Robert, N. Sentiment analysis and text summarization of online reviews: A survey. In Proceedings of the 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 6–8 April 2016; pp. 241–245.

4. Gupta, H.; Patel, M. Study of extractive text summarizer using the elmo embedding. In Proceedings of the 4th International Conference on IoT in Social, Mobile, Analytics and Cloud, Palladam, India, 7–9 October 2020; pp. 829–834.
5. Baharudin, B.; Lee, L.H.; Khan, K. A Review of Machine Learning Algorithms for Text-Documents Classification. *J. Adv. Inf. Technol.* **2010**, *1*, 4–20. [CrossRef]
6. Strubell, E.; Ganesh, A.; McCallum, A. Energy and policy considerations for deep learning in NLP. *arXiv* **2019**, arXiv:1906.02243.
7. Jensen, C.; Colin, D. Privacy policies as decision-making tools: An evaluation of online privacy notices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vienna, Austria, 24–29 April 2004.
8. Singh, A.K.; Shashi, M. Vectorization of Text Documents for Identifying Unifiable News Articles. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 305. [CrossRef]
9. Cambria, E.; White, B. Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]. *IEEE Comput. Intell. Mag.* **2014**, *9*, 48–57. [CrossRef]
10. Adhika, W.; Supriadi, R.; Guruh, S.; Edi, N.; Abdul, S.; Affandy, A.; Setiadi, D.R.I.M. Review of automatic text summarization techniques & methods. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *11*, 21–27.
11. Saeed, M.Y.; Awais, M.; Talib, R.; Younas, M. Unstructured Text Documents Summarization with Multi-Stage Clustering. *IEEE Access* **2020**, *8*, 212838–212854. [CrossRef]
12. Akbik, A.; Bergmann, T.; Blythe, D.; Rasul, K.; Schweter, S.; Vollgraf, R. FLAIR: An easy-to-use framework for state-of-the-art NLP. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), Minneapolis, MN, USA, 2–7 June 2019; pp. 54–59.
13. Babar, S.A. Text summarization: An overview. In Proceedings of the Babar Summarization, Alicante, Spain, 2013; pp. 13–17.
14. Walkowiak, T.; Datko, S.; Maciejewski, H. Bag-of-words, bag-of-topics and word-to-vec based subject classification of text documents in polish-a comparative study. In *Contemporary Complex Systems and Their Dependability. Proceedings of the Thirteenth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, Brunów, Poland, 2–6 July 2018*; Springer International Publishing: Berlin, Germany, 2019; pp. 526–535.
15. Stenetorp, P.; Pyysalo, S.; Topić, G.; Ohta, T.; Ananiadou, S.; Tsujii, J.I. BRAT: A web-based tool for NLP-assisted text annotation. In Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, 23–27 April 2012; pp. 102–107.
16. Jugran, S.; Kumar, A.; Tyagi, B.S.; Anand, V. Extractive automatic text summarization using SpaCy in python & NLP. In Proceedings of the International Conference on Advance Computing and Innovative Technologies in Engineering, Greater Noida, India, 4–5 March 2021; pp. 582–585.
17. Dredze, M.; Jansen, A.; Coppersmith, G.; Church, K. NLP on spoken documents without ASR. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, USA, 9–11 October 2010; pp. 460–470.
18. Meystre, S.M.; Haug, P.J. Comparing natural language processing tools to extract medical problems from narrative text. *AMIA Annu. Symp Proc.* **2005**, *2005*, 525–529. [PubMed]
19. Gaikwad, D.K.; Mahender, C.N. A review paper on text summarization. In Proceedings of the Gaikwad2016ARP IJARCCE, Maharashtra, India, 2016; pp. 234–239.
20. Rajman, M.; Besançon, R. Text mining: Natural language techniques and text mining applications. In *Data Mining and Reverse Engineering: Searching for semantics. Proceedings of the IFIP TC2 WG2. 6 IFIP Seventh Conference on Database Semantics (DS-7), Leysin, Switzerland, 7–10 October 1997*; Springer: New York, NY, USA, 1998; pp. 50–64.
21. Merchant, K.; Pande, Y. NLP based latent semantic analysis for legal text summarization. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics, Bangalore, India, 19–22 September 2018; pp. 1803–1807.
22. El-Kassas, W.S.; Salama, C.R.; Rafea, A.A.; Mohamed, H.K. Automatic text summarization: A comprehensive survey. *Expert Syst. Appl.* **2020**, *165*, 113679. [CrossRef]
23. Joshi, A.; Fidalgo, E.; Alegre, E.; Fernández-Robles, L. SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Syst. Appl.* **2019**, *129*, 200–215. [CrossRef]
24. Crossley, S.A.; Kim, M.; Allen, L.; McNamara, D. Automated summarization evaluation (ASE) using natural language processing tools. In Proceedings of the Artificial Intelligence in Education, Austin, TX, USA, 2019; pp. 84–92.
25. Zhong, L.; Zhong, Z.; Zhao, Z.; Wang, S.; Ashley, K.D.; Grabmair, M. Automatic summarization of legal decisions using iterative masking of predictive sentences. In Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, Montreal, QC, Canada, 17–21 June 2019; pp. 163–172.
26. Sabireen, H.; Neelanarayanan, V. A review on fog computing: Architecture, fog with IoT, algorithms and research challenges. *ICT Express* **2021**, *7*, 162–176.
27. Rahul, K.; Adhikari, S.; Monika, D. NLP based machine learning approaches for text summarization. In Proceedings of the 4th International Conference on Computing Methodologies and Communication, Erode, India, 11–13 March 2020; pp. 535–538.
28. Moens, M.-F. Innovative techniques for legal text retrieval. *Artif. Intell. Law* **2001**, *9*, 29–57. [CrossRef]
29. Moritz, K.; Heinz, H.J. *I/S: A Journal of Law and Policy for the Information Society*; Center for Interdisciplinary Law and Policy Studies: Columbus, OH, USA, 2005; Volume 12, no. 2, pp. 125–132.
30. Trappey, A.J.; Trappey, C.V.; Wu, C.-Y. Automatic patent document summarization for collaborative knowledge systems and services. *J. Syst. Sci. Syst. Eng.* **2009**, *18*, 71–94. [CrossRef]

31. He, S.; Zeng, W.; Xie, K.; Yang, H.; Lai, M.; Su, X. PPNC: Privacy preserving scheme for random linear network coding in smart grid. *KSII Trans. Internet Inf. Syst.* **2017**, *11*, 1510–1532.
32. Javed, Y.; Salehin, K.M.; Shehab, M. A study of South Asian websites on privacy compliance. *IEEE Access* **2020**, *8*, 156067–156083. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.