mathematics

MDPI

Article

Really Vague? Automatically Identify the Potential False Vagueness within the Context of Documents

Xiaoli Lian [1], Dan Huang [1], Xuefeng Li [1], Ziyan Zhao [1,*], Zhiqiang Fan [2] and Min Li [2]

[1] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
20373526@buaa.edu.cn (X.L.)
[2] North China Institute of Computing Technology, Beijing 100083, China
* Correspondence: zhaoziyan@buaa.edu.cn

Abstract: Privacy policies are critical for helping individuals make decisions on the usage of information systems. However, as a common language phenomenon, ambiguity occurs pervasively in privacy policies and largely impedes their usefulness. The existing research focuses on the identification of individual vague words or sentences, without considering the context of documents, which may cause a significant amount of false vagueness. Our goal is to automatically detect the potential false vagueness and the related supporting evidence, which illustrates or explains the vagueness, and therefore probably assist in alleviating the vagueness. We firstly analyze the public manual annotations and define four common patterns of false vagueness and three types of supporting evidence. Then we propose the approach of the *F·vague-Detector* to automatically detect the supporting evidence and then locate the corresponding potential false vagueness. According to our analysis, about 29–39% of individual vague sentences have at least one clarifying sentence in the documents, and experiments show good performance of our approach, with recall of 66.98–67.95%, precision of 70.59–94.85%, and F_1 of 69.24–78.51% on the potential false vagueness detection. Detecting the vagueness of isolated sentences without considering their context within the whole document would bring about one-third potential false vagueness, and our approach can detect this potential false vagueness and the alleviating evidence effectively.

Keywords: privacy policy; potential false vagueness; NLP; vagueness

MSC: 68-04; 68P27

check for updates

Citation: Lian, X.; Huang, D.; Li, X.; Zhao, Z.; Fan, Z.; Li, M. Really Vague? Automatically Identify the Potential False Vagueness within the Context of Documents. *Mathematics* **2023**, *11*, 2334. https://doi.org/10.3390/math11102334

Academic Editor: Antanas Cenys

Received: 4 April 2023
Revised: 9 May 2023
Accepted: 10 May 2023
Published: 17 May 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Nowadays, various network-based systems surround us and collect our personal information for the purpose of improving their services. Privacy policies work as the official documents telling us how the systems collect, use, and share our personal data. Furthermore, they can also be viewed as technical documents stating the multiple privacy-related requirements that the related system should satisfy to handle the personal data properly [1]. Privacy policies have received significant attention by the requirement engineering community because they should be consistent with system behaviors [2–6].

With the increase in people's awareness of privacy protection, more companies are sued and fined due to leaking or using people's data illegally. For example, eight companies, including Instagram, Foursquare, Kik, Gowalla, Foodspotting, Yelp, Twitter, and Path, have contributed USD 5.3 million to a payment pot for consumers because their "find friends" feature passed users' personal information to other service providers without notification and user consent. Actually, not only companies but also states are starting to pay more attention to privacy policies. For example, the European Union's *General Data Protection Regulations* clearly stipulate that *data controllers must inform users of the collection, storage, and use of personal data in a transparent and concise language in a timely and truthful manner*. So the privacy policy should be crystal clear for customers.

However, privacy policies are usually presented through natural language (NL) statements, and therefore, vagueness is inevitable due to the nature of NL. According to the empirical research of the Factorial Vignette Survey of Jaspreet Bhatia et al. [7], the increase in uncertainty (such as ambiguity or vagueness) will reduce the user's acceptance of privacy risks, thereby reducing their willingness to share personal information. A lot of research has yielded great results so far. For the sake of simplicity, these works can be classified into two streams: vagueness prevention and vagueness detection. For the first class, some organizations propose templates, such as *privacy policies* (https://www.privacypolicies.com/blog/privacy-policy-template/, accessed on 4 April 2023) and *TermsFeed* (https://www.termsfeed.com/blog/sample-privacy-policy-template/, accessed on 4 April 2023), in order to specify the constituent parts of privacy policies. Regarding vagueness detection, the existing research focuses on detecting vague words [7,8] and individual sentences [9]. However, in any case, the words or sentences are within the context of whole documents. Due to the phenomenon of long-distance dependencies, lots of vague words or sentences are probably explained by the statements in other places, even in different paragraphs or sections.

For example, according to Lebanoff et al. [8,9], *personal information* is annotated as a common vague phrase in the privacy policies of the top websites ranked on Alexa.com accessed on 4 April 2023 [10]. However, according to our observations, most of these policies contain the explanation of this important phrase scattered in different parts of the documents, some of which appear around its first appearance and some in the specific glossary section. For instance, in the crowdsourcing annotations of Lebanoff et al. [9], *personal information* appears 16 times in the overall 63 annotations of vague terms (≈25.4%) in the privacy policy of *Groupon* (updated on 13 September 2012). However, the document contains a specific section, *Glossary of Terms*, which provides a detailed explanation of this phrase, including its definition and compositions. We show the example of Groupon in Figure 1, in which we mark each appearance of *personal information* with red font and its definition with a green background. We believe that the degree of vagueness of this phrase given by the annotators will be mitigated greatly with the assistance of this definition. In other words, it is quite possible that a large portion of vague words or sentences detected by these approaches is actually false vagueness.

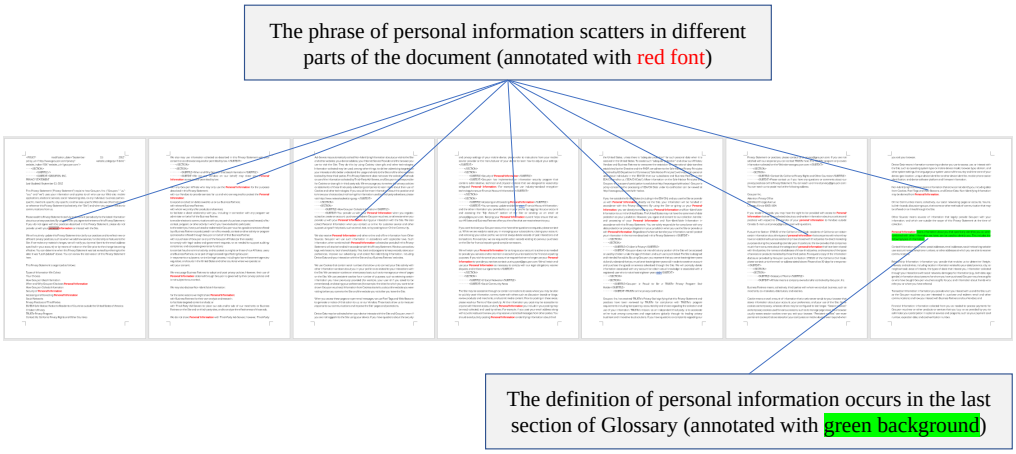


Figure 1. In the privacy policy of Groupon, the phrase of *personal information* is annotated as vague 16 times by the crowdsourcing annotators in [9], and its definition occurs in the last section, long and different distances from the occurrences of the phrase.

Therefore, the purpose of this study is to automatically identify the supporting sentences, as they can explain the individual words or sentences to a certain degree; these sentences would be determined as vague without considering the very long-distance dependencies, i.e., the paragraph- or section-level dependencies. However, we cannot promise

that these words or sentences with supporting sentences are false vagueness since it is really challenging to automatically decide if the explanation is clear enough to eliminate the vagueness. So we call these words or sentences *potential false vagueness* in this work.

To achieve our goal, we propose an approach called the potential False vagueness Detector for privacy policies (referred to as *F·vague-Detector*). To be specific, we firstly analyze the vague annotations in the public manuals of the privacy policies of 15 popular websites, randomly selected from the dataset of Lebanoff et al. [9], and define four common patterns of supporting sentences based on the principle of grounded theory [11]. Then we establish a series of heuristic rules on the grounds of NLP results, such as part of speech and semantic dependencies, and design a set of algorithms for detecting these types of supporting sentences and locating the corresponding potential false vague words and sentences. Experiments with 30 privacy documents show the great performance of our *F·vague-Detector*, which yields an F1-measure of 75.41% on the training set for the potential false vague sentence identification and 67.51% on the testing set. Meanwhile, for the supporting sentence identification, the F1-measure reaches 73.92% on the training set and 69.24% on the testing set.

The contributions of this study can be summarized as follows:

- We present the first study on considering the vagueness of privacy policies within the context of the whole documents, rather than individual words or sentences.
- We categorize the patterns of sentences which can provide support to alleviate the vagueness in the (potentially) false vague sentences or phrases through exemplification or explanation, more or less.
- We present the first study on detecting the potential false vagueness and the related supporting evidence, which can help to improve the results of vagueness detection in privacy policies of the existing work.
- Experiments show great performance of our *F·vague-Detector* in both potential false vagueness and the supporting evidence detection.

The remainder of this paper is organized as follows. In Section 2, we make a simple survey about the definition of *vagueness* and the existing research on vagueness prevention in privacy policies. In Section 3, we depict the whole framework of this work. In Section 4, we report the definition of four types of potential false vagueness and three patterns of supporting evidence. Then in Sections 5 and 6, we report the detailed procedure of our approach on the identification of the potential false vagueness and the related supporting evidence. Finally, in Sections 7 and 8, we discuss the threats to validity and give the conclusion of our work.

2. Related Work and Background

In this section, we firstly give the definitions of vagueness in the existing academic research and our concerns. Then we provide a simple review of the related work on vagueness prevention and detection in privacy policies.

2.1. The Definition of Vagueness

Van deemter [12] believes that "if a concept or word does not have a clear boundary, then it is ambiguous", e.g., "big", "many", and "rarely".

Keefe et al. [13] think that vague words will cause "sorites paradox", which means that subtle changes in objects do not affect the applicability of vague terms. For example, if a room is now "bright", then after dimming the light a little bit, it can also be considered bright. If you continue dimming the light a little bit until it goes out completely, the room can also remain "bright". So the word "bright" is ambiguous.

L.A. Zadeh [14] proposed the concept of "vague set", which is a class of objects with continuous subordinate relationships. Such a set is characterized by a degree of membership (feature) function that assigns a degree of membership between 0 and 1 to each object. Objects encountered in the real physical world often do not have clear boundaries. For example, the "tall man" is a vague set which does not have an exact scope in mathematics.

RMkempson [15] proposed that the vagueness of nominal terms is caused by the lack of accurate descriptions. A word's meaning is clear, but it is only a general reference, e.g., "neighbor" without the specification of "race", "age", and so on. This kind of ambiguity can be eliminated by adding some qualifiers, such as "a laughing five-foot professor of Welsh philosophy of research".

Since we mean to detect the potential false vagueness in the existing research, the vagueness in this study has a general and broad definition. In other words, vagueness indicates that the meaning of the NL elements (i.e., words, phrases, or sentences) differs for different people.

2.2. Preventing Vagueness in Privacy Policy

Writing a privacy policy in formal language is a good way to avoid ambiguity. Cranor [16,17] uses the platform for privacy preferences (P3P) to introduce a standard machine-readable format for website privacy policies. The P3P specification defines a standard method for encoding website privacy policies in XML format and a mechanism for locating and transmitting P3P policies. The P3P format regulates the content of the privacy policy, including user data, third-party service data, business data, and dynamic data, as well as the set of elements that all data should include. The P3P format defines 11 purpose sub-elements to represent data usage. In addition, each purpose child element has a required attribute that indicates whether the data can always be used for this purpose based on opt-in or opt-out. At present, many companies have developed software to build privacy policies in the P3P format [18] and provided verification procedures to ensure that the privacy policies are free of grammatical errors.

Although the P3P format can guarantee the accuracy of the privacy policy, it requires writers to regulate the language more strictly and use it properly [8]. The P3P format requires a certain programming background, and the general writer is the company's legal consultant or consulting expert, so the P3P format privacy policy is not widely used, and more privacy policies are still written in natural language.

As a regulation on the processing of personal data and on the free movement of such data, the General Data Protection Regulation (GDPR) (https://gdpr-info.eu/, accessed on 4 April 2023) which entered into force in May 2018 in the European Union, specifies the content of privacy policies, including information on the company, type of personal data, purpose of processing, storage period, transfer third country, source of personal data, rights to withdraw, automated decision making, cookie policy, etc. [19].

Sadeh et al. [20] described an "available privacy policy project" which aims to semi-automatically extract important content from privacy policies for presentation to end users in a structured and understandable format. It is proposed that cutting-edge technologies such as crowdsourcing, machine learning, and natural language processing can be used to solve related problems. Under this project, Ammar et al. [6,21–23] studied the use of crowdsourcing-labeled data in privacy policies and carried out some classification of the data information, e.g., whether the privacy policy is transparent on law enforcement requests in [6]. Liu et al. [10] group the text segments (e.g., paragraphs or sections) by the issues they address (without pre-specifying). Sathyendra et al. [24] studied how to automatically detect user opt-out choices in privacy policies so that users can make informed decisions. They can allow users to quickly jump to the paragraphs of text related to certain key privacy practices, alleviating the "too long to read" challenge. This project can detect whether there is a description of some important content in the privacy policy, but the accuracy of the description is not guaranteed.

2.3. Automatic Detection of Vagueness in Privacy Policy

Since most privacy policies are still described in natural language, the work of ambiguity detection in privacy policies is mostly directed to natural language. Current research includes both manual detection [7] and automatic approaches [8,9].

Jaspreet Bhatia et al. [7] derived and classified 26 ambiguous terms based on a manual analysis of 15 website privacy policies. Through a pairwise comparison experiment and Bradley–Terry model, they obtained the hierarchical order between single vague terms and combined ones. This hierarchical order is capable of predicating the degree of ambiguity of the combination of various types of information and different information behavior. This work allows users to manually or automatically locate the sentences containing vague terms.

Fei Liu et al. [8] decoded the ambiguity of privacy policies from the perspective of natural language processing, and they attempted to discover grammatically and semantically related vague terms based on deep neural networks.

Logan Lebanoff et al. [9] proposed an automatic privacy policy ambiguity detection algorithm based on deep neural network models. They firstly constructed a corpus with artificially labeled vague words and sentences. Then they designed their algorithms and showed that the algorithm considering the sentence-level context of vague terms outperforms the context-free models in predicting vague words.

From these related works, we can find that all of these works consider the vagueness of privacy policies from the perspective of single words and sentences. One primary reason is that words and sentences are not only the basic units of documents but also the source of vagueness. The privacy policies with more vague words and sentences of high ambiguity probably contain serious vagueness [9]. We therefore concentrate on the vagueness of these two units too. However, the whole document presents a complete semantic structure, and the sentences of the document always have some kind of relationship. Detecting the vagueness of isolated sentences is obviously inappropriate while ignoring its context in the whole document, i.e., the semantic dependency between different sentences.

Furthermore, there is some research on preventing and detecting the vagueness or ambiguity in other artifacts, such as software requirements [25–30] and fake news [31]. Similarly, their approaches include providing restricted NL for requirement description [25] for ambiguity prevention and detecting vague words/phrases [27] and sentences [26,29,30]. However, we have not yet found any research that considers the ambiguity of these elements within the context of the whole document.

3. The Framework of Our *F·Vague-Detector*

Let the algorithm F obtain the vague description of privacy policy Y in the results of the current vagueness detection research as the set $V = \{V_1, V_2, ..., V_n\}$. If there is a set of sentences $S = \{S_1, S_2, ..., S_m\}$ which can explain or illustrate the element V_i, then V_i may be misjudged. V_i is referred to as the potential false vagueness, and S_i is a supporting sentence (evidence). The goal of this study is to identify the potentially false vague sentences and their supporting sentences in the detection results of the latest research scheme (i.e., vague words and vague sentences automatically detected by deep neural network models) from the privacy policy document, thereby assisting to adjust the automatic detection of the current research and helping people to improve the quality of privacy policies.

The overview of our approach is depicted in Figure 2. To be specific, our work includes three phases. In Phase I, we obtain all potential vagueness (i.e., vague phrases and sentences) from manual annotation or the current vagueness detection algorithm. In this present work, we select the crowdsourcing annotations in the work of Lebanoff et al. [9] due to two considerations. Firstly, the quality of manual annotations is trustworthy. They assigned five people for each single sentence annotation and obtained the selected times of the vague terms and the average vagueness value of the sentence. Second, we expect to obtain the ratio of potential false vagueness and the characteristics of that false vagueness for automatic detection, and there is obviously still a gap between the algorithm's performance and the manual results. For example, in the work of Lebanoff et al. [9], the F-Measure is about 60% for vague term detection and 52% for sentence vagueness classification. Then in the Phase II, we define the patterns of supporting sentences of the potentially false vagueness by manually analyzing a random sample from Phase I,

by following the principle of grounded theory [11]. Finally, in Phase III, we identify the potentially false vagueness and the supporting evidence automatically. In particular, we firstly define the heuristic rules for identifying the different patterns of supporting sentences and propose the corresponding algorithms for the automatic identification of these supporting sentences. Then we match the identified supporting sentences with all candidate potential vagueness.

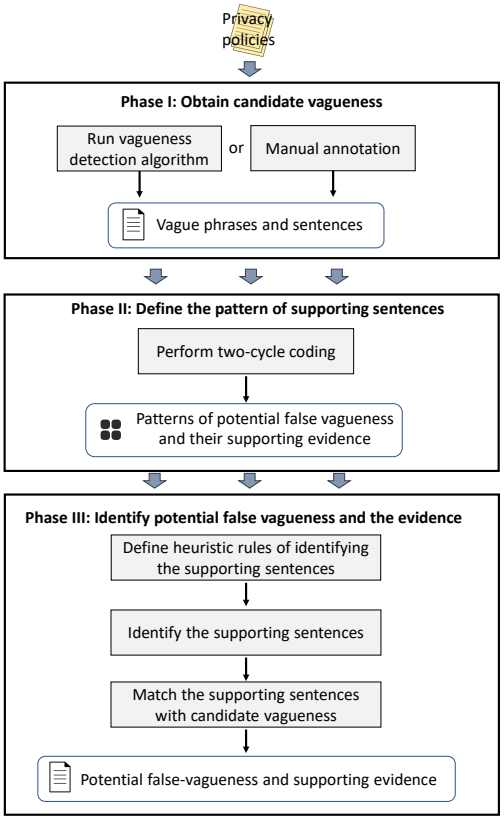


Figure 2. The overall framework of this study.

3.1. Data Source

We use the corpus constructed by Logan Lebanoff [9], which includes 133 K words and 4.5 K sentences from 100 website privacy policies and the related manual vagueness annotations (i.e., the vague terms and the degree of vagueness of sentences). These privacy documents are lengthy, containing on average 2.3 K words. They hired annotators from Amazon Mechanical Turk and assigned individual sentences to five people for annotations using crowdsourcing. For each vague term, they marked the number of annotators who regarded the term as vague. The score of the degree of vagueness of the sentence is from 1 to 5, and the corresponding value is from very clear to very vague. Then they took the average scores of the five annotators, and the average value of the ambiguity of each sentence is distributed in four intervals of [1, 2], (2, 3], (3, 4], and (4, 5], corresponding to "clear", "some vague", "vague", and "extremely vague", respectively.

In this pilot study, we randomly select 30 companies' privacy policy documents from the corpus, 15 for manual analysis and 15 for testing. Since we mean to study the characteristics of vague statements in privacy policies, we therefore firstly filter the non-vague ones in the dataset, which are assessed as clear by the annotators and for which the average degree of vagueness is less than 2 points.

3.2. Research Questions

In this present work, we aim to address the following three research questions.

- RQ1: What is the proportion of vague sentences in privacy policies, which have a minimum of one clarifying sentence to reduce ambiguity, as identified in Lebanoff's manual annotations [9]?
- RQ2: Do the supporting sentences present some patterns? What are the common patterns?
- RQ3: To what extent can our approach automatically identify the potentially false vagueness and at least one piece of supporting evidence (sentence)?

4. Defining the Patterns of Supporting Sentences

We detect the potentially false vague sentences (or phrases) by identifying their supporting evidence (i.e., supporting sentences in this study). Once any supporting evidence is found, the corresponding sentence is deemed as potential false vagueness. To automatically identify the supporting evidence, we need to firstly analyze the characteristics. In this section, we try to find the common patterns of these supporting sentences.

4.1. Manual Data Labeling and Analysis Process

We analyze and label 15 random training privacy policies based on grounded theory [11], which is widely used in qualitative analysis research [7,32]. Saldana summarized the ground-based theory of two-cycle coding for data analysis in 2012 and proposed 32 data processing strategies [33]. The first stage is the initial data labeling, focusing on the basic characteristics of the data, such as attributes, structure, and so on. The second stage is about the analysis strategy, the goal of which is to achieve classification, priority determination, synthesis, abstraction, conceptualization, and theoretical analysis based on the labeled data in the first stage. The usage of this two-cycle coding method can ensure the relative objectivity of the data labeling process and the relative accuracy of the conclusions [32].

According to the two-cycle coding, we firstly used one data processing strategy, *attribute coding*, focusing on analyzing whether the vague statement in a privacy policy has a *potentially false vague* attribute. In other words, for each vague sentence selected by annotators, we scanned the whole document to determine whether there is any supporting sentence for clarifying the vagueness to a certain extent. This manual analysis work involves the first two authors. They independently and carefully read the full text of the 15 policies and gave their answers. They needed to list each potential false vague sentence and their supporting sentences in the answer sheet.

In the second cycle, we employed the strategy of *pattern coding* with the purpose of reaching an agreement on the results of the first cycle and also exploring the patterns of these supporting sentences for later automatic detection. In particular, the first two authors discussed and analyzed the records one by one in their answer sheets. One record would be retained if (a) it occurs in the two answer sheets or, (b) though it only occurs in one answer sheet, the other author can be persuaded. They then categorized the remaining records by their relationship between the supporting sentence and the potentially false vague sentence. After this work, they developed a classification guide book together. Then, we recruited another annotator (referred to as *annotator_C*) and asked him independently to read the 15 privacy policies, to record the potentially false vague sentences and their supporting sentences, and to define the patterns of their relationships.

Finally, the three annotators (i.e., the first two authors and *annotator_C*) conducted a joint discussion. They compared and analyzed the annotation results of *annotator_C* with those of the two authors. In the end, they improved the annotation on potential false vagueness and refined the categories.

4.2. The Results of Manual Annotations: The Patterns of Supporting Sentences of Potentially False Vagueness

We divide the potential false vague sentences into four categories according to the relationship with their supporting sentences: potential false vagueness supported by some supplemental statements (referred to as supplementary supporting pattern), supported by example statements (referred to as exemplification supporting pattern), supported by the interpretation of specific terms (referred to as interpretation supporting pattern), and the potential false vagueness describing some kinds of domain phenomenon.

We would like to introduce the phenomenon-related vagueness first. In almost all of the privacy policies, there are some statements describing the infrastructure beyond the current information system. For example, lots of privacy policy documents involve a general description of browser behavior. Because browsers work as the common infrastructure of all web systems and because there are several browsers, it is almost impossible for these companies to spend a lot of space in privacy policies on the description of browsers. People would obviously not spend too much time on too lengthy a privacy policy.

For example, in the privacy policy of *Horoscope company*, there is one sentence that is marked manually as vague in the work of Lebanoff et al. [9]:

- **Vague sentence:** *Although Google Analytics plants a persistent Cookie on your web browser to identify you as a unique user the next time you visit the Website , the Cookie can not be used by anyone but Google* (from the privacy policy of Horoscope company).
- **Vague words manually labeled:** *persistent cookie, unique, persistent*

Horoscope company should not spend lots of words on the explanation of common technical terms such as *persistent cookie*. We regard this kind of sentence as potential false vagueness since the related vagueness can be mitigated through simple learning through a search engine. We put this kind of false vagueness out of our research since the scope of these technical concepts is too broad and the privacy documents seldom include the supporting sentences for them.

Now we introduce the other three types of potential false vagueness and their supporting sentences.

- **Supplementary supporting pattern**

Some statements are so complex that they are composed of several separate items (e.g., bullet items). The following is one example from the privacy policy of Turkish Airlines:

The purposes for which Data may be used are as follows:

(a) processing your bookings / orders and managing your account with us;
(b) marketing our services or related products;
(c) for compiling aggregate statistics about you and maintaining contact lists for correspondence, commercial statistics and analysis on Site usage;
(d) for identity, verification and records.

The above paragraph is processed as five individual sentences (i.e., the first starting sentence and the following four bullet items) in the current research [9]. From the perspective of single sentences, they are indeed ambiguous because their semantics are incomplete. However, these sentences are complementary to each other. The ambiguity of these statements as a whole must be mitigated.

This kind of sentence usually contains relatively obvious syntactic or paragraph structure characteristics, such as a colon and labels (i.e., numbers or alphabetic characters), which will be discussed in Section 5.2.

- **Exemplification Supporting Pattern**

When it comes to stating an important fact or something that is difficult to understand, people always give examples. Obviously, as an aid to comprehension, a proper example assists in mitigating the degree of vagueness to a great extent. The example can be for a specific word or the whole sentence. We refer to as the *exemplification supporting pattern* the kind of sentences which carry examples for explaining *the whole sentence*.

Take one sentence from the privacy policy of *Groupon* labeled as vague in [9] as an example:

- **Vague sentence:** *All the information you post may be accessible to anyone with Internet access, and any Personal Information you include in your posting may be read, collected, and used by others.*
- **One exemplification supporting sentence:** *For example, if you post your email address along with a public restaurant review, you may receive unsolicited messages from other parties.*

The identification of the occurrence of this exemplification supporting sentence and the corresponding potentially false vagueness will be discussed in Section 5.3.

- **Interpretation Supporting Pattern**

The interpretation supporting pattern refers to the type of sentence that explains a specific vague word in the vague sentence in the ways of definition or exemplification. The difference between this pattern and the *exemplification supporting pattern* is that this focuses on the explanation of individual words rather than vague sentences in the *exemplification supporting pattern*.

Take one sentence from the privacy policy of *Rockstar Games, Inc.*, which is marked as vague in [9], as an example:

- **Vague sentence:** *We will retain your Personal Information for as long as your account is active or as needed to provide you services and to maintain a record of your transactions for financial reporting purposes.*
- **Vague word:** *Personal Information*
- **One interpretative supporting sentence:** *Personal Information means the information about you that specifically identifies you or, when combined with other information we have, can be used to identify you.*

In this example, the supporting sentence gives the definition of the vague word *personal information* in the vague statement. According to the manual annotations of this sentence, *personal information* is the most vague word. The degree of vagueness of the whole sentence is therefore expected to be mitigated with the assistance of this definition.

We calculate the ratio of the sentences of these four patterns in the 15 training privacy policies, as shown in Figure 3. We can observe that in all the potential false vague sentences (271 sentences), the biggest part (about 64%) can be explained by the sentences with the *interpretative supporting pattern*. In other words, the vague words in these sentences can be clarified to some extent. Actually, some vague words, such as personal information, which have been explained in other places, occur frequently and almost all of the sentences containing these vague words are also marked as vague. The second is general knowledge (about 16%). The third is the category that can be supported by supplementary sentences (about 15%). The remaining potential false vagueness (about 5%) is illustrated by examples.

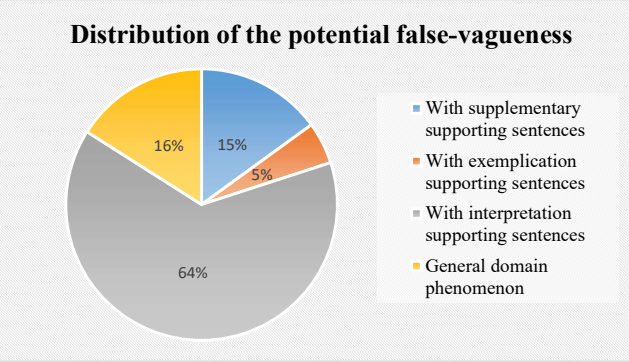


Figure 3. The pattern distribution of potential false vagueness sentences in the 15 random samples.

4.3. Addressing RQ1 and RQ2

Until now, we have addressed the first research question **RQ1:** *What is the proportion of vague sentences in privacy policies, which have a minimum of one clarifying sentence to reduce ambiguity, as identified in Lebanoff's manual annotations [9]?* and the second one **RQ2:** *Do the supporting sentences present some patterns? What are the common patterns?*

4.3.1. Experimental Data

Just as the description in Section 3.1, we randomly selected 30 privacy policies and their corresponding manual annotations from the corpus of Lebanoff et al. [9]. These 30 privacy policies were randomly divided into two parts, i.e., training and testing datasets. We explored the patterns of supporting sentences and their identified rules from the training dataset and evaluated their applicability in the testing set.

The *training dataset* includes 15 privacy policies for manual analysis to obtain patterns of supporting sentences and the heuristic rules for identifying potential false vague sentences and their supporting evidence. There are 1380 sentences in total, of which 271 are marked as vague and 106 pairs of ⟨potential false vague sentence, supporting sentence⟩ are collected. In the 106 pairs, there are 22 ⟨potential false vague sentence, supplementary supporting sentence⟩ pairs, 6 pairs of ⟨potential false vague sentence, exemplification supporting sentence⟩, and 78 pairs of ⟨potential false vague sentence, interpretative supporting sentence⟩.

The *testing dataset* includes 15 privacy policies too, which do not overlap with those in the training dataset. Our two authors annotated these 15 policies independently to find the potentially false vagueness and the supporting evidence. They then categorized the patterns of this supporting evidence by following our guidelines. Finally, these two authors discussed and reached an agreement on the final results. We will evaluate the generality of our *F·vague-Detector* on the testing dataset. Thus, the quality of these annotations is critical and we calculated the inner agreements given in Section 4.3.2. There are 1169 sentences in all, and 269 are marked as vague in [9]. According to the annotations of our authors, there are 78 pairs of potential false vague sentences and their supporting evidence, of which there are 19 pairs with supplementary supporting sentence, 14 pairs with exemplification supporting evidence, and 45 pairs with interpretative sentences.

4.3.2. Inner Agreement Analysis on the Supporting Pattern Annotations on Testing Dataset

Before showing the analysis results about the ratio of potentially false vagueness and the patterns of supporting evidence, we need to evaluate the quality of the manual annotations. We focus on the annotations of the testing dataset here because we want to check the quality of our guidebook and relatively reliable annotations should be achieved on the training dataset due to the two-round discussion (i.e., the two authors and one annotator with them).

We used *Cohen's Kappa* to measure the inner agreement between the manual labeling results of the first two authors on the testing data. Cohen's Kappa coefficient κ is a statistic used to measure the inter-rater agreement among evaluators of qualitative (classified) items [34]. We selected κ because it takes the possibility of the agreement by chance into account and therefore it is more robust than the simple percentage statistics.

Assume that there are two evaluators, a and b, and each divides N samples into C mutually exclusive categories. *Kappa* can be defined as

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \tag{1}$$

p_o is the probability of the observed agreement between the evaluators, which is calculated as the sum of the samples (S_i) whose labels by the two annotators are consistent divided by the total number of samples N.

$$p_o = \frac{1}{N}\sum_{i=1}^{C} S_i \tag{2}$$

p_e is the probability of the random consistency, which means that the two evaluators give the labels randomly and the results are just consistent.

$$p_e = \frac{1}{N^2}\sum_{i=1}^{C} n_{ia} n_{ib} \tag{3}$$

where C represents the number of mutually exclusive classifications, n_{ia} represents the number of samples in category i given by evaluator a, and n_{ib} depicts that given by evaluator b.

The range of *kappa* is [−1,1], and a higher value means better agreement.

We expect to measure the agreement on the judgment of the potential false vagueness and the existence of specific supporting evidence type. To be specific, for each of the 269 sentences that are marked as vague manually in Lebanoff et al. [9], we checked the result of our two annotators (noted as *annotator_a* and *annotator_b* in Table 1). If the annotator regards this sentence as potential false vagueness and meanwhile finds at least one supporting sentence of the specific type, we mark it as "yes"; otherwise, we mark it as "no". The result is shown in Table 1.

Table 1. *Kappa*: Inner agreement of the manual annotations on potential false vagueness and the existence of specific type of supporting evidence.

		Supplementary Supporting		**Exemplification Supporting**		**Interpretation Supporting**		**Overall**	
		Authors		**Authors**		**Authors**		**Authors**	
		Yes	**No**	**Yes**	**No**	**Yes**	**No**	**Yes**	**No**
annotator_c	yes	19	0	12	2	51	9	82	11
	no	0	519	0	255	10	199	10	704
kappa		1.0		0.92		0.80		0.87	

We can observe that best agreement has been achieved on annotating the existence of the evidence on the sentence with the *supplementary supporting pattern* and the related potential false vagueness (i.e., perfect agreement). The reason is that there are very obvious features identifying this kind of evidence and the distance between the false vagueness and the evidence in documents is often close. The second best agreement is with the *exemplification supporting pattern* (i.e., *kappa* is about 0.92), followed by the annotation of the *interpretation supporting pattern* (*kappa* is about 0.80). Besides the three supporting patterns, we also calculated the overall *kappa*, which measures the agreement on the overall 807 records (i.e., 269 × 3). The overall *kappa* is about 0.87. According to Viera et al. [35], perfect agreement is reached for identifying all of the three kinds of evidence supporting potential false vagueness (i.e., $kappa > 0.80$).

4.3.3. Addressing RQ1 and RQ2

From the annotation results on the training and testing datasets (i.e., 30 privacy policies) in Section 4.3.1, we can make the following conclusions:

- For the training dataset, in the isolated sentences that are marked manually as vague in [9], about 39% are potentially false vague which have at least one supporting sentence. These supporting sentences cover all three patterns that we focus on in

this study; supplementary support shows 20.75%, exemplification shows 5.66%, and interpretative support shows 73.58%.

- For the testing dataset, about 29% are isolated sentences that are manually marked as vague in [9], are potentially false vague, and have at least one supporting sentence. These supporting sentences cover the all three patterns we focus on too. In general, supplementary support shows 24.36%, exemplification support shows 17.95%, and interpretative support shows 57.69%.

Therefore, we can address the first two research questions.

Addressing RQ1: By analyzing the crowdsourcing annotations on a random sample of 30 privacy policies in [9], about 29–39% of the isolated sentences selected as vague by people are potentially false vague, and they have at least one piece of supporting evidence.

Addressing RQ2: By analyzing the crowdsourcing annotations on the random 30 privacy policies of websites used worldwide in [9], we find there are at least three patterns of the supporting evidence, and they commonly occur in both the training and testing datasets. To be specific, interpretative supporting sentences occur most, while the second most frequent is the supplementary, followed by the exemplification supporting type.

5. *F·Vague-Detector*: Our Approach for Automatically Detecting the Potential False Vagueness and The Supporting Evidence

The primary target of this study is to automatically identify the potential false vagueness from the automatic detection results of state-of-the-art methods. In order to achieve this goal, we propose our approach, the *F·vague-Detector*, which detects the potentially false vague sentences by identifying their clarifying evidence. In other words, in case any supporting sentence is detected, we think the related "vague" sentences are potentially false vague. Therefore, the critical part is the identification of the three types of supporting sentences, which will be introduced in detail in this section.

In general, with the purpose of identifying supporting sentences of the three patterns, we firstly analyze their characteristics from the aspects of the text pattern (such as keywords and punctuation) and lexical and syntax features and design the automated algorithm for each type. Then we design algorithms to associate the "vague" sentences in the existing results with this supporting evidence to determine the potentially false vague sentences.

5.1. Identifying Interpretive Supporting Sentences and Their Potentially False Vague Evidence

We categorize the sentences that explain or illustrate the vague words in other "vague" sentences as interpretative support to the vague sentence containing these vague words. The identification of this kind of supporting sentence is most difficult because the potential false vagueness and the interpretative sentences usually scatter in different sections of the privacy policy, and furthermore, there may be synonyms of the vague words whose identification is tough using state-of-the-art methods [36]. Furthermore, interpretative supporting evidence occurs most frequently according to the analysis in the above section; the identification of potentially false vagueness and this evidence type is vitally important for our study.

In general, this work includes three steps. Actually, the idea is intuitive. We firstly extract all interpretative sentences from the privacy policies which are capable of explaining or defining some specific words of other sentences. Then we identify and extract the interpreted word from the interpretative sentence. Finally, we match the explained word and the vague word in the existing answer sheet on vagueness detection. We would like to explain the concept of the interpreted word with one example. The sentence *Personal information means the information about you that specifically identifies you or, when combined with other information we have, can be used to identify you* in Section 4.2 is referred to as an interpretative sentence and *personal information* in this sentence is the interpreted word.

Step 1: Identify all interpretative sentences from the privacy document. We perform feature analysis on a random sample, define the rules, and then propose an algorithm that identifies all candidate sets of interpretative sentences from the privacy policy.

In order to annotate the features of interpretative sentences, we perform a two-cycle coding. The procedure is just like that in Section 4. We will not give the duplicated details here. The manual analysis results in two types of interpretation: exemplification and definition.

The exemplification for a specific word usually occurs in the clause containing keywords such as *include*, *including*, and *such as*. The defining sentence usually presents an explicit structure such as *noun phrase + mean(s)* or *noun phrase + be + subject structure*. However, because the sentence pattern of *be + subject structure* is widely used in English, it cannot be used as a feature for defining the sentence. Otherwise, lots of false positive results will be returned. We call these feature words/phrases, including *include*, *including*, *such as*, and *mean(s)*, the matching words of interpretative sentences.

With the purpose of evaluating the effectiveness of these matching words, we calculate their coverage in the 15 training samples. There are 95 sentences marked as interpretative supporting by our two authors, and 86 contain at least one of the above matching words. The coverage is about 91%, meaning that these matching words occur frequently in the interpretative sentences.

Although the coverage of matching keywords is very high, not all of the sentences containing matching words are interpretative sentences. The primary reason is the phenomenon of polysemy. For example, the word *means* contains several semantic denotations, such as *methods* as a noun, *to express something* as a verb, or *intend to* as a verb. Thus, we set the part-of-speech filter for the words *mean* and *means*.

Step 2: Extract the interpreted words in the interpretative sentences. We analyze the candidate interpretative sentences from three perspectives, syntax structure, text pattern, and semantic dependencies, to define heuristic rules for extracting the interpreted words in these sentences.

- *Syntax structure-based heuristic rules* Through the analysis on the content and the parsing tree of interpretative sentences, we find some rules about the occurrence of interpreted words.
  The interpreted word and its modifiers usually appear before the matching word (if it exists), and the detailed explanation usually comes after it. The modifiers include noun phrases, subordinate clauses, and so on.
  We analyzed the features for interpreted word identification from the parsing tree of the interpreted sentences too. We find that the interpreted words are usually noun phrases (NPs) or demonstrative pronouns (DETs). Their occurrence in the parsing tree presents the following characteristics:
  Rule I: The modifiers of interpreted words (usually NPs) may be a prepositional phrase (PP) or verbal phrase (VP), and the NP, VP, and PP are usually in the same level of the parsing tree. In addition, the matching words closely follow the modifiers (if they exist). Therefore, if there is a subtree in the parsing tree of the interpretative sentence satisfying the structure of *NP + PP||VP + matching words*, the NP is the interpreted word of this interpretative sentence.
  Rule II: The NP closest to the matching words is usually the interpreted word. There may be multiple NPs with the same distance to the matching words, and some may have a nested relationship between each other. We select the longest one because it carries more complete semantics.
  If *Rule I* and *Rule II* are satisfied, we select the results from *Rule I* as the interpreted words.
  For the noun phrases, we would like to find the extension based on the parsing tree of the current sentence, while for the demonstrative pronouns, we would like to perform coreference resolution in the current sentence and its neighbors with the Stanford NLP tool. To be specific, we use the noun phrase composed of the most words in the coreference chain as its antecedent.
- *Semantic dependency-based heuristic rule* Depending purely on the syntax-based rules without considering the semantic information is not enough for the interpreted words'

extraction, which loses some of the relationship and may cause lots of false positives. We therefore attempt to optimize the extraction results with the usage of the information of semantic dependency parsing (SDP).

Semantic dependency parsing is a method to analyze the binary semantic association between the language units (i.e., word or phrase) in one sentence. The SDP result of the Stanford NLP Group is a collection of triples known as the locations of two single words and their semantic dependency relation.

Rule III: Through analyzing the results of SDP, we find that when the words *include* or *mean* occur as the root verb, the *subject* of the whole sentence is usually the interpreted word. The corresponding relation in SDP is *nsubj* between the core noun of the interpreted word and the matching word. However, since single words as the components of phrases often contain part of the semantics, we need to extend them and find their nested noun phrases. For example, from the sentence "*Personal information includes your email address, last name...*", we can obtain the relation triple of (2,3,nsubj), where the 2 and 3 are the place ID of the words *information* and *includes*. However, *personal information* rather than *information* should be the interpreted word. In order to obtain the accurate interpreted word, we need to find the extension of the core word *information* in this sentence (i.e., personal information).

- *Text pattern-based heuristic rules* Finally, we propose two rules for two special patterns of sentences to optimize the extraction results.

  Rule IV: The candidate interpretative sentence may start with *noun phrase + colon*. In this case, the whole following part of this sentence is used to explain the noun phrase. For example, iin the sentence *Transaction Information: information you provide when you interact with us and the Site, such as the Groupon vouchers you are interested in, purchase and redeem...*, the content after the colon is the explanation of the transaction information.

  Rule V: When the candidate interpretative sentence contains the pattern *following + noun phrase + colon*, the part after the colon is the explanation of the noun phrase. For example, in the sentence *The information collected may include but is not limited to the following personal information: email address, last name...*, the items after the colon are the enumerations of personal information.

In summary, we define five rules for detecting the interpreted words from the candidate interpretative supporting sentence. We assign priorities for them, as shown in Table 2, and the results from the rule of higher priority would be selected if at least two rules are satisfied.

Step 3: Find the potentially false vague sentences that the interpretative sentences' support The purpose of this step is to find the potential false vagueness and the interpretative supporting sentences through matching. The bridge is the vague words in vague sentences and the interpreted words in the interpretative sentences. If the vague word is same as or synonymous with the interpreted word, the original vague sentence is potentially false vague and the interpretative sentence is its supporting evidence.

This work is challenging for a few reasons. The first is synonyms with different forms. For instance, *personal information*, *personally identifiable information*, and *personal data* all occur frequently in privacy policies, but automatically identifying their synonymous relation is not easy. Another difficulty is polysemy. For example, the phrase *other information* has different meanings in diverse contexts, sometimes meaning personal information but sometimes not.

Because the demonstrative pronouns have been replaced into the corresponding noun/phrase in the extraction based on syntax structure rules, we perform the mapping based on the synonymous extraction. To put it simply, we firstly extract the domain terms of privacy policies based on the approach of C-Value [37]. Then we detect the pairs of terms with a synonym relation using the approach of [38]. Finally, we walk through the synonymous list and perform manual optimization of the results. Now, we briefly introduce the process of synonymous detection.

Table 2. Rules for extracting the interpreted word in the candidate interpretative sentences.

Category	Features of the Interpretative Sentence	Interpreted Word	Priority
syntax structure-based	there is at least one NP before matching words	the NP closest to matching words in the parsing tree	1
	there is a subtree satisfying the structure *NP + (NP| VP) + matching words*	the first NP (may be with modifiers)	2
semantic dependency-based	there is an *nsubj* relation between some NN/DET and matching words.	the NP composed by NN or the NP referred to by DET	3
text pattern-based	the sentence contains the linguistic form *NP + colon*	the NP before colon	4
	the sentence contains the linguistic form *following + NP + colon*	the NP between the word "following" and colon	5

NN: noun; NP: noun phrase; DET: demonstrative pronoun.

Domain concept extraction: We employ the C-value to extract the domain multi-term terminologies. We select this approach because it is simple but effective. It considers the length of concepts and their nested relation, so it performs well in extracting the long concepts. Piao et al. [39] show that it is a stable approach for concept extraction. A stream of research on concept extraction includes the improvement of the C-Value, such as by Frantzi et al. [40] and Ventual et al. [41].

We focus on noun phrases because we find that the vague terms having explanation or definition in other places are usually nouns through a primary walkthrough of the results of manual annotations of [9] and the corresponding privacy policies. In particular, the C-value includes two steps: obtain candidate terminology based on a POS tagger and a POS pattern filter, and determine the termhood and unithood of the candidates based on statistical measure.

Synonymy of multi-word terms detection: We employ the approach of Hazem et al. [38] for synonymous detection. The main idea behind their approach is that the synonyms should be in the same context. So they firstly use the technology of word embeddings to obtain the vectors of candidate phrases. Each phrase can be divided into three parts, the starting word, middle word, and ending word, of which the middle one can be empty. We then find the synonyms from the ones having the same starting or ending word. We calculate the cosine similarity between the different parts of these phrases and select the ones above our predefined threshold.

Manual checking and correcting: In order to acquire the list of synonyms in privacy policies, we run this approach on the 1000 privacy documents containing 107 K sentences collected by Liu et al. [8]. However, we find that this approach falsely regards some phrases with different meanings but similar context as synonymous. We therefore perform manual checking and correcting on the extraction results.

5.2. Identifying the Supplementary Supporting Sentences and Their Potential False Vague Evidence

Supplementary supporting evidence refers to the statements that are composed of several items, separated with a colon or other explicit marks, and work as complement to each other. In the current research on vagueness detection, such as [9], these items are processed separately. In the absence of context, these sentences are misjudged as vague.

We find that there are two kinds of incomplete statements: a starting statement and several following items. The starting statement is always an overview or the objective of the following items, while the detailed items are the refinement of the opening one.

We extracted all the incomplete statements from the 15 training privacy policies and analyzed the features from the aspect of text structure. The features are listed in Table 3.

The starting sentence often ends with a colon, indicating that a list of items, explaining or illustrating this present sentence, follows. Identifying the illustrated items is relatively hard, since sometimes the items may be too long to find all of them, especially the last one. Thus we collect a group of features for identifying the items from different aspects, including (i) punctuation feature: the enumerated items usually end with a colon, except that the last one ends with a period; (ii) sequential feature: the items are usually ordered alphabetically or numerically, starting with some kind of ordering marks, such as numbers or letters; (iii) paragraph feature: sometimes there are no obvious ordering marks at the beginning of the items, as the items may be lengthy paragraphs beginning with a few umbrella terms (usually less than four words); (iv) special cases: nowadays no part of an information system is isolated, and they often need third-party services. Generally, when the detailed description of third parties is not involved, their URLs will be directly listed.

Table 3. The features of supplementary supporting sentences.

Types of Incomplete Statements	Features
starting statement	ending with ";"
enumerated items	the last item ends with ".", while the others with ";"
	usually, items are ordered alphabetically or numerically, and start with the ordering marks, such as (a) (b) (c) or (1) (2) (3) and so on
	sometimes, items are arranged as individual paragraphs and start with short umbrella terms.
	if third parties are illustrated, the URLs are probably the main body of the items.

The identification of the supplementary supporting sentences is based on the rules listed in Table 3. To be specific, this process begins with the starting sentence identification from the whole privacy policy. We firstly detect all the statements ending with a colon. Then we determine whether the following text satisfies the features of the illustrated items. If so, the starting sentence and the following text work as an integrated semantic union. Then we check whether this starting sentence or any illustrated item was annotated as vague in the existing result list. If yes, we note it as potential false vagueness. For the starting sentence, all of the illustrated items are the supplement. For any isolated illustrated item, both the starting sentence and the other items are the supporting evidence.

5.3. Identifying the Exemplification Supporting Sentences and Their Potentially False Vague Sentence

Through the manual analysis, we find that most examples for clarifying specific sentences begin with obvious keywords, such as "for example" and "for instance". So, we just locate the sentences starting with these two keywords and collect their previous sentence too. Furthermore, they usually appear after the sentence that is clarified. If the previous ones are determined as vague in the answer sheet, they are potentially false vague and the following sentences are their exemplification supporting evidence.

6. Addressing RQ3: The Effectiveness Evaluation of *F·Vague-Detector*

In this section, we mean to address **RQ3:** *To what extent can we automatically identify the potentially false vagueness and at least one piece of supporting evidence?* In the above Section 5, we propose our *F·vague-Detector* for identifying the potentially false vague sentences and their three kinds of supporting evidence. For each kind of supporting evidence, we design an approach for identifying the related statements and then matching with the potential false vagueness they support. Here we would like to evaluate these approaches separately.

6.1. Metrics

In order to evaluate the extent of the automatic identification of our *F·vague-Detector*, we design four metrics:

UPairs: The number of pairs of ⟨one potentially false vague sentence, one supporting sentence of specific pattern⟩ manually marked.

APairs: The number of pairs of ⟨one potentially false vague sentence, one supporting sentence of specific pattern⟩ identified automatically by *F·vague-Detector*.

CPairs: The number of pairs of ⟨one potentially false vague sentence, one supporting sentence of specific pattern⟩ that occur in both the manual and automatic answer sheets.

RPairs: The number of pairs of ⟨one potentially false vague sentence, one supporting sentence of specific pattern⟩ that are correctly identified by *F·vague-Detector*. Although our authors took significant time on annotating the supporting sentences, we may lose some evidence for the potential false vagueness, especially the weak supporting since most privacy policies are pretty lengthy. Therefore, we need to walk through the entire automatic answer sheet to determine their correctness.

On the basis of the above four metrics, we redefine the Recall, Precision, and F-Measure.

Recall: the ratio of correct automatic identification in all reference answers.

$$recall = \frac{CPairs}{UPairs} \tag{4}$$

Precision: the ratio of correct identification in all automatic results.

$$precision = \frac{RPairs}{APairs} \tag{5}$$

F_1: the harmonic mean of recall and precision.

$$F_1 = \frac{2 * recall * precision}{recall + precision} \tag{6}$$

6.2. Results and Analysis

We conduct statistics on both the training and testing datasets in Section 4.3.1 beyond the above metrics. The training dataset is used to measure the intrinsic effectiveness of the algorithms, and the testing dataset is for the extrinsic evaluation as well as the generality of our approach. To be specific, for each dataset, we firstly evaluate the effectiveness of our *F·vague-Detector* in identifying the pairs of potentially false vague sentences and each pattern of supporting sentence, and then give the overall evaluation. The results can be seen in Tables 4 and 5.

Table 4. Evaluation results of the automatic identification by our *F·vague-Detector* on training dataset: pairs of the potential false vagueness and their supporting sentences.

Supporting Pattern		UPairs	APairs	CPairs	RPairs	Recall	Precision	F_1
Supplementary	**Starting sentences**	11	12	10	12	90.91%	100%	95.24%
	Enum items	11	18	10	17	90.91%	94.44%	92.64%
Exemplification		6	5	5	5	83.33%	100%	90.91%
Interpretative		78	159	46	150	58.97%	94.34%	72.58%
Overall		106	194	71	184	66.98%	94.85%	78.51%

Table 5. Evaluation results of the automatic identification by our *F·vague-Detector* on testing dataset: pairs of the potential false vagueness and their supporting sentences.

Supporting Pattern		UPairs	APairs	CPairs	RPairs	Recall	Precision	F_1
Supplementary	**Starting sentences**	8	7	7	7	87.50%	100%	93.33%
	Enum items	11	18	10	13	90.91%	72.22%	80.50%
Exemplification		14	12	10	12	71.43%	100%	83.33%
Interpretative		45	65	20	40	57.78%	61.54%	59.60%
Overall		78	102	53	72	67.95%	70.59%	69.24%

- From the results of the overall evaluation on the training dataset (i.e., the last row in Table 4), the overall recall on the training set is 66.98%, the precision is 94.85%, and the F_1 value is 78.51%. On the testing dataset (i.e., the last row in Table 5), the overall recall is 67.95%, the precision is 70.59%, and the F_1 value is 69.24%. The results on both these two datasets look good, although there is a certain gap between the results on the training data and testing data.
- For the identification of pairs of ⟨potential false vagueness, the starting sentence pattern⟩:
  - *Identification effectiveness:* On the training set, the recall reaches 90.91%, the precision is 100%, and the F_1 is 95.24% (seen in Table 4), while for the testing set, the recall is 87.5%, the precision is 100%, and F_1 value is 93.33% (seen in Table 5).
  - *Versatility analysis:* The gap between the recall of our *F·vague-Detector* on the training and testing sets is 3.41%, the precision gap is 0.00%, and the F_1 value is 1.91%. The performance of our algorithm on the testing set is very close to that on the training set. This shows that our identification rules of the starting supplementary supporting sentence and the corresponding potential false vague sentences have very strong generality.
- For the identification of the pairs of ⟨potential false vagueness, enumeration supplementary sentence⟩:
  - *Identification effectiveness:* On the training set, the recall reaches 90.91%, the precision is 94.44%, and the F_1 is 92.64% (seen in Table 4). For the testing dataset, the overall recall is 90.91%, the precision is 72.22%, and the F_1 is 80.5% (seen in Table 5).
  - *Versatility analysis:* There is no gap between the recall of our *F·vague-Detector* on the training and testing dataset. For precision and F_1, the gaps are 22.22% and 12.14%. Although the recall values show great generality of our approach on the identification of the enumerated items, there is still space for improvement. By manually analyzing the pairs of ⟨potential false-vagueness, the enumerated items⟩, we found that the primary reason is that judging the ending item of the enumerations is not easy. Sometimes these enumerations and their following

paragraph have the similar text structure and context, and it is hard to distinguish. We will make an improvement on the identification rules in future work.

- For the identification of the pairs of ⟨potential false vagueness, exemplification supporting sentence⟩:
  - *Identification effectiveness:* On the training set, the recall is 83.33%, the precision reaches 100%, and the F_1 is 90.91% (seen in Table 4), while on the testing set, the recall is 71.43%, the precision is 100%, and the F_1 is 83.33%.
  - *Versatility analysis:* The gap between the recall of our approach on the training and testing sets is 11.9%, the precision is 0%, and the F_1 is 7.58%. We can say that the effect of our approach on the identification of the pairs of ⟨potential false vagueness, exemplification supporting sentence⟩ from these two datasets is very close, showing great generality. However, there is still room to improve. By manually analyzing the wrong identification, we found that the primary reason is that our rules are too strong to restrict the sentences containing specific keywords. So we miss some "implicit" exemplifications. We plan to improve our identification algorithm for this kind of sentence in future work.
- For the identification of the pairs of ⟨potential false vagueness, interpretative supporting sentence⟩:
  - *Identification effectiveness:* The recall of our approach on the training set is 58.97%, the precision reaches 94.34%, and the F_1 value is 72.58%; on the testing set, the recall is 57.78%, the precision is 61.54%, and the F_1 is 59.6%.
  - *Versatility analysis:* The gap between the recall of our approach on the training and testing sets is 1.19%, the precision is 32.8%, and the F_1 value is 12.98%. Although the performance of our approach on the testing data is good, the F_1 gap is the biggest on all the supporting patterns. Through our manual analysis, we find that the primary problem is on the match between the vague terms and the interpreted items. This is because both the identification of synonyms and the anaphora resolution are unsatisfying nowadays and still hard problems in the field of NLP, even for a big dataset. We plan to improve the results by exploring other features, such as the semantic similarity between the interpretative and the vague sentences.

Therefore, we can address the third research question according to these analyses.

Addressing RQ3:Experiments on both the training and testing privacy policies show that our *F·vague-Detector* can effectively identify the potentially false vagueness and the related three types of supporting sentences, especially for the supplementary and exemplification supporting types. For the interpretative supporting evidence, the identification result is relatively weaker and the F_1 is 59.60–72.58%.

7. Threats to Validity

The first threat comes from the manual annotations on the potential false vagueness and the patterns of their supporting sentences on both the 15 training and testing privacy policies. In order to alleviate the bias of subjectivity, we conduct the annotations according to the approach of two-cycle coding with the principle of grounded theory (seen in Section 4). This systematic procedure can improve the confidence level by independent annotation by the first two authors firstly, then joint discussion until agreement has been achieved, and finally the evaluation by the third annotator. On the other hand, in order to improve the quality of our annotation, we only focus on the manual selections of Lebanoff et al. [9] so that we can avoid lots of tedious and unnecessary analyzing work on the clear sentences within the context of the whole lengthy documents, which is also error-prone.

The second threat is from the patterns of supporting sentences. In this study, we focus on identifying the three primary types of supporting sentences, which may help clarify the vagueness of other sentences. We mean to illustrate that it is critical to consider the context

of the whole document while evaluating the vagueness of individual sentences. However, we cannot promise that we collect the full set of supporting patterns due to the limited dataset for analysis (i.e., 15 privacy policies).

The third threat is caused by the generality evaluation. We use the other 15 privacy policies, which do not occur simultaneously in the training set, for the generality evaluation. However, we need to evaluate our approach on a bigger dataset in future work.

8. Conclusions

Nowadays, information systems have spread to every corner of people's lives, so privacy matters to everyone. As a legal statement, privacy policies should obviously be clear. There is a lot of research interest on the identification of vague words or sentences in privacy policies, including work such as [7–9]. However, in any case, the words and sentences cannot be interpreted fully without the consideration of the context of the entire document, due to semantic interdependence.

In this present study, we mean to show that a considerable amount of individual sentences (about 29–39% in our case) selected as vague are potentially false vague, with the existence of at least one of the three types of supporting sentence as evidence. In particular, in order to identify this potential false vagueness from the answer sheets of the current (manual or automatic) vagueness detection approaches, we firstly defined four types of common potential false vagueness and further defined three patterns of their supporting sentences by analyzing the crowdsourcing manual annotations of the vague sentences on 100 privacy policies from [9]. We then proposed an approach called the *F·vague-Detector* in which we design algorithms to identify the evidence of *interpretative supporting sentences*, *supplementary supporting sentences*, and *exemplification supporting sentences*. Experiments show great effectiveness and generality of our approach. To be specific, our approach is capable of identifying the supporting sentences with an overall F_1 of 78.51% on the training data on the three supporting patterns and 69.24% on the testing data. In particular, for the supplementary and exemplification pattern, the F_1 reaches more than 90.0% on the training dataset and 83.33% on the testing dataset.

In the future we plan to continue this work from the following directions:

- Evaluate our *F·vague-Detector* on many more privacy documents. In this pilot study, we only use the random 30 privacy policies of the corpus of Lebanoff et al. [9].
- Evaluate the vagueness of whole documents rather than single words or sentences. We hope to assist in improving the quality of privacy policies with a metric of the document's vagueness.
- Recommend the crucial but unspecified items in a privacy policy. These kinds of items usually appear frequently and definitely hinder people's understanding of the privacy policies without clear definition.

Author Contributions: Conceptualization: X.L. (Xiaoli Lian) and D.H.; Methodology and Software: D.H., X.L. (Xuefeng Li) and Z.Z.; Validation: Z.Z., X.L. (Xiaoli Lian) and Z.F.; Formal analysis: X.L. (Xiaoli Lian) and M.L.; Resources: Z.Z.; Data curation: D.H. and X.L. (Xuefeng Li); Writing—original draft: D.H.; Writing—review and editing: X.L. (Xiaoli Lian) and Z.F.; Visualization: D.H.; Supervision: X.L. (Xiaoli Lian); Project administration: X.L. (Xiaoli Lian); Funding acquisition: X.L. (Xiaoli Lian) and M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been funded by the National Science Foundation of China Grant No. 62102014. It is also supported by Innovation Fund of Beijing Huaxing Tai Chi Information Technology Co., Ltd. No. 19010201.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available within the article.

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

References

1. Torre, D.; Abualhaija, S.; Sabetzadeh, M.; Briand, L.C.; Baetens, K.; Goes, P.; Forastier, S. An AI-assisted Approach for Checking the Completeness of Privacy Policies Against GDPR. In Proceedings of the 28th IEEE International Requirements Engineering Conference, RE 2020, Zurich, Switzerland, 31 August–4 September 2020; Breaux, T.D., Zisman, A., Fricker, S., Glinz, M., Eds.; IEEE: Piscataway, NJ, USA, 2020; pp. 136–146.
2. Hosseini, M.B.; Breaux, T.D.; Slavin, R.; Niu, J.; Wang, X. Analyzing privacy policies through syntax-driven semantic analysis of information types. *Inf. Softw. Technol.* **2021**, *138*, 106608. [CrossRef]
3. Caramujo, J.; da Silva, A.R.; Monfared, S.; Ribeiro, A.; Calado, P.; Breaux, T.D. RSL-IL4Privacy: A domain-specific language for the rigorous specification of privacy policies. *Requir. Eng.* **2019**, *24*, 1–26. [CrossRef]
4. Breaux, T.D.; Hibshi, H.; Rao, A. Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements. *Requir. Eng.* **2014**, *19*, 281–307. [CrossRef]
5. Bhatia, J.; Evans, M.C.; Breaux, T.D. Identifying incompleteness in privacy policy goals using semantic frames. *Requir. Eng.* **2019**, *24*, 291–313. [CrossRef]
6. Massey, A.K.; Eisenstein, J.; Anton, A.I.; Swire, P.P. Automated text mining for requirements analysis of policy documents. In Proceedings of the 2013 21st IEEE International Requirements Engineering Conference (RE), Rio de Janeiro, Brazil, 15–19 July 2013; pp. 4–13. [CrossRef]
7. Bhatia, J.; Breaux, T.D.; Reidenberg, J.R.; Norton, T.B. A theory of vagueness and privacy risk perception. In Proceedings of the 2016 IEEE 24th International Requirements Engineering Conference (RE), Beijing, China, 12–16 September 2016; pp. 26–35.
8. Liu, F.; Fella, N.L.; Liao, K. Modeling language vagueness in privacy policies using deep neural networks. In Proceedings of the 2016 AAAI Fall Symposium Series, Arlington, VA, USA, 17–19 November 2016.
9. Lebanoff, L.; Liu, F. Automatic detection of vague words and sentences in privacy policies. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 3508–3517.
10. Liu, F.; Ramanath, R.; Sadeh, N.; Smith, N.A. A step towards usable privacy policy: Automatic alignment of privacy statements. In Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 884–894.
11. Martin, P.Y.; Turner, B.A. Grounded Theory and Organizational Research. *J. Appl. Behav. Sci.* **1986**, *22*, 141–157.
12. Van Deemter, K. *Not Exactly: In Praise of Vagueness*; Oxford University Press: Oxford, UK, 2012.
13. Keefe, R. *Theories of Vagueness*; Cambridge University Press: Cambridge, MA, USA, 2000.
14. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
15. Kempson, R.M. *Semantic Theory*; Cambridge University Press: Cambridge, MA, USA, 1977.
16. Cranor, L. *Web Privacy with P3P*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2002.
17. Cranor, L.F.; Guduru, P.; Arjula, M. User interfaces for privacy agents. *ACM Trans.-Comput.-Hum. Interact. (TOCHI)* **2006**, *13*, 135–178. [CrossRef]
18. P3P Implementations. Available online: http:www.w3.org/P3P/implementations (accessed on 28 October 2019).
19. Galle, M.; Christofi, A.; Elsahar, H. The Case for a GDPR-specific Annotated Dataset of Privacy Policies. In Proceedings of the AAAI Workshop, Honolulu, HI, USA, 27–28 January 2019.
20. Sadeh, N.; Acquisti, A.; Breaux, T.D.; Cranor, L.F.; McDonald, A.M.; Reidenberg, J.R.; Smith, N.A.; Liu, F.; Russell, N.C.; Schaub, F.; et al. *The Usable Privacy Policy Project*; Technical Report, CMU-ISR-13-119; Institute for Software Research School of Computer Science, Carnegie Mellon University: Pittsburgh, PA, USA, 2013.
21. Ammar, W.; Wilson, S.; Sadeh-Koniecpol, N.; A Smith, N. *Automatic Categorization of Privacy Policies: A Pilot Study*; Technical Report CMU-LTI-12-019; School of Computer Science, Language Technology Institute: Pittsburgh, PA, USA, 2012.
22. Wilson, S.; Schaub, F.; Dara, A.A.; Liu, F.; Cherivirala, S.; Leon, P.G.; Andersen, M.S.; Zimmeck, S.; Sathyendra, K.M.; Russell, N.C.; et al. The creation and analysis of a website privacy policy corpus. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 1330–1340.
23. Wilson, S.; Schaub, F.; Ramanath, R.; Sadeh, N.; Liu, F.; Smith, N.A.; Liu, F. Crowdsourcing Annotations for Websites' Privacy Policies: Can It Really Work? In Proceedings of the 25th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, Montreal, QC, Canada, 11–15 April 2016; pp. 133–143.
24. Sathyendra, K.M.; Wilson, S.; Schaub, F.; Zimmeck, S.; Sadeh, N. Identifying the provision of choices in privacy policy text. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; pp. 2774–2779.
25. Boyd S, S.; Zowghi, D.; Farroukh, A. Measuring the expressiveness of a constrained natural language: An empirical study. In Proceedings of the 13th IEEE international conference on Requirements Engineering (RE'05), Paris, France, 29 August–2 September 2005; pp. 339–352.
26. Yang, H.; de Roeck, A.; Gervasi, V.; Willis, A.; Nuseibeh, B. Analysing anaphoric ambiguity in natural language requirements. *Requir. Eng.* **2011**, *16*, 163–189. [CrossRef]
27. Cruz, B.D.; Jayaraman, B.; Dwarakanath, A.; McMillan, C. Detecting Vague Words & Phrases in Requirements Documents in a Multilingual Environment. In Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference (RE), Lisbon, Portugal, 4–8 September 2017; pp. 233–242.

28. Asadabadi, M.R.; Chang, E.; Sharpe, K. Requirement ambiguity and fuzziness in large-scale projects: The problem and potential solutions. *Appl. Soft Comput.* **2020**, *90*, 106148. .: 10.1016/j.asoc.2020.106148. [CrossRef]
29. Tjong, S.F.; Berry, D.M. The Design of SREE—A Prototype Potential Ambiguity Finder for Requirements Specifications and Lessons Learned. In *Requirements Engineering: Foundation for Software Quality*; Doerr, J., Opdahl, A.L., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 80–95.
30. Yang, H.; De Roeck, A.; Gervasi, V.; Willis, A.; Nuseibeh, B. Speculative requirements: Automatic detection of uncertainty in natural language requirements. In Proceedings of the 2012 20th IEEE International Requirements Engineering Conference (RE), Chicago, IL, USA, 24–28 September 2012; pp. 11–20.
31. Guélorget, P.; Icard, B.; Gadek, G.; Gahbiche, S.; Gatepaille, S.; Atemezing, G.; Égré, P. Combining vagueness detection with deep learning to identify fake news. In Proceedings of the 2021 IEEE 24th International Conference on Information Fusion (FUSION), Sun City, South Africa, 1–4 November 2021; pp. 1–8. [CrossRef]
32. Onwuegbuzie, A.J.; Frels, R.K.; Hwang, E. Mapping Saldana's Coding Methods onto the Literature Review Process. *J. Educ. Issues* **2016**, *2*, 130–150. [CrossRef]
33. Saldana, J. *The Coding Manual for Qualitative Researchers*; SAGE Publications Ltd.: Thousand Oaks, CA, USA, 2009.
34. Cohen, J. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46. [CrossRef]
35. Viera, A.J.; Garrett, J.M. Understanding interobserver agreement: The kappa statistic. *Fam. Med.* **2005**, *37*, 360–363.
36. Hazem, A.; Daille, B. Semi-compositional method for synonym extraction of multi-word terms. In Proceedings of the 9th Edition of the Language Resources and Evaluation Conference (LREC 2014), Reykjavik, Iceland, 26–31 May 2014.
37. Frantzi, K.; Ananiadou, S.; Mima, H. Automatic recognition of multi-word terms: The C-value/NC-value method. *Int. J. Digit. Libr.* **2000**, *3*, 115–130. [CrossRef]
38. Hazem, A.; Daille, B. Word Embedding Approach for Synonym Extraction of Multi-Word Terms. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018; European Language Resources Association (ELRA): Miyazaki, Japan, 2018; pp. 297–303.
39. Piao, S.; Forth, J.; Gacitua, R.; Whittle, J.; Wiggins, G. Evaluating tools for automatic concept extraction: A case study from the musicology domain. In Proceedings of the Digital Economy All Hands Meeting-Digital Futures 2010, Nottingham, UK, 11–12 October 2010. Available online: https://core.ac.uk/download/pdf/1557928.pdf#:~:text=This%20paper%20reports%20on%20an%20evaluation%20of%20five,most%20suitable%20for%20the%20task%20of%20concept%20extraction (accessed on 4 April 2023).
40. Frantzi, K.; Ananiadou, S. The C-value/NC-value domain independent method for multi-word term extraction. *J. Nat. Lang. Process.* **1999**, *6*, 20–27. [CrossRef] [PubMed]
41. Lossio-Ventura, J.A.; Jonquet, C.; Roche, M.; Teisseire, M. Combining C-value and Keyword Extraction Methods for Biomedical Terms Extraction. In Proceedings of the International Symposium on Languages in Biology and Medicine (LBM'2013), Tokyo, Japan, 12–13 December 2013; pp. 45–49.

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.