applied sciences

MDPI

Article

Building and Evaluating an Annotated Corpus for Automated Recognition of Chat-Based Social Engineering Attacks

Nikolaos Tsinganos and Ioannis Mavridis *

Department of Applied Informatics, University of Macedonia, 156 Egnatia Str., 54636 Thessaloniki, Greece; tsinik@uom.edu.gr

* Correspondence: mavridis@uom.edu.gr

Featured Application: Methods and techniques demonstrated in this work can be used to increase chat-based social engineering attack detection algorithms.

Abstract: Chat-based Social Engineering (CSE) is widely recognized as a key factor to successful cyber-attacks, especially in small and medium-sized enterprise (SME) environments. Despite the interest in preventing CSE attacks, few studies have considered the specific features of the language used by the attackers. This work contributes to the area of early-stage automated CSE attack recognition by proposing an approach for building and annotating a specific-purpose corpus and presenting its application in the CSE domain. The resulting CSE corpus is then evaluated by training a bi-directional long short-term memory (bi-LSTM) neural network for the purpose of named entity recognition (NER). The results of this study emphasize the importance of adding a plethora of metadata to a dataset to provide critical in-context features and produce a corpus that broadens our understanding of the tactics used by social engineers. The outcomes can be applied to dedicated cyber-defence mechanisms utilized to protect SME employees using Electronic Medium Communication (EMC) software.

Keywords: cybersecurity; sensitive data; social engineering; corpus; annotation; chat-based attack; named entity recognition

Citation: Tsinganos, N.; Mavridis, I. Building and Evaluating an Annotated Corpus for Automated Recognition of Chat-Based Social Engineering Attacks. *Appl. Sci.* **2021**, *11*, 10871. https://doi.org/10.3390/app112210871

Academic Editor: Paula Fraga-Lamas

Received: 17 October 2021
Accepted: 14 November 2021
Published: 17 November 2021

1. Introduction

SMEs remain one of the most popular targets for cyber-attacks. It is also a fact that employees nowadays are extensively using Information and Communication Technologies (ICT) and especially Electronic Medium Communication (EMC) software for almost every aspect of their daily activities. Although EMC software is an advantageous tool for the employees, it also constitutes a large and vulnerable attack surface. The vulnerabilities lay in human personality characteristics, and the way adversaries exploit them to extort valuable information for their benefit. As stated by ENISA [1], the term Social Engineering "refers to all techniques aimed at talking a target into revealing specific information or performing a specific action for illegitimate reasons".

Researchers performed several successful classifications of the social engineering attacks [2–4] based on criteria such as the entity involved, the medium used to unleash the attack or the number of steps an attack can take. All these classifications end up in fine-grained taxonomies that include different methods of social engineering attacks. Verizon in [5] reports that in 2020 social engineering attacks were at 21%, the first step of every cyber-attack that led to a major data breach. Furthermore, the same report mentions that social engineering attackers' methods of choice are phishing and pretexting. The pretext method uses an invented scenario to facilitate the attacker to persuade the employee to do what the attacker wants [4].

Social engineering attacks are not only unleashed by humans; recent advances in Chatbot technology escalate even more the problem of social engineering. Chatbots [6]

are conversational agents that are communicating through artificial intelligence (AI) and natural language processing (NLP). A chatbot dedicated to accomplishing a malicious task can utilize multiple personalities in an attempt to deceive an SME employee and extract sensitive data. Chatbots are also capable of approaching all types of humans presenting different personalities based on the six principles of persuasion defined by R. Cialdini [7] (reciprocity, scarcity, authority, commitment, liking, and social proof). Moreover, due to the COVID-19 pandemic the use of conversational agents increased, and it is a rather common practice for an employee to socialize with a chatbot as part of daily routine operations. SME employees must be aware of the possible dangers and be protected from malicious questions which aim to extract sensitive data. Although chatbots are a novel tool in the hands of social engineers, the rapid pace in AI and NLP makes them an important factor due to their capabilities. Today, the majority of social engineering attacks are operated by humans, but AI-driven social engineering attacks are expected to happen in the near future through chatbots too [8].

As employees use chat-based software to offer their services, this can become dangerous if a social engineer decides to unleash a sophisticated attack using a pretext scenario during an online conversation. A chat-based social engineering (CSE) attack cannot be described using only technical terms due to the nature of the human vulnerabilities. Thus, we need a multi-factor recognition system to automatically detect a CSE attack. An ideal multi-factor recognizer should process the dialogue in real time and protect the employee by predicting potential CSE attacks in early stages, e.g., by alerting the employee if the probability of sensitive data leakage exceeds a predefined threshold.

Such a prediction can be based either on a pure statistical approach or a machine learning (ML) approach. Considering the importance of recognizing a CSE attack in real-time, the ML approach can qualify as an efficient solution. Furthermore, ML algorithms can be combined perfectly with NLP algorithms which can be used to process the language used in a natural communication setting. Such ML/NLP algorithms need labelled datasets to be trained and to be efficient in their predictions. For example, a set of processed dialogues (corpus) of realized CSE attacks can be tagged in sentence-level to discriminate the malicious sentences from the benign ones. A ML/NLP algorithm trained with this corpus will be able to efficiently recognize a malicious sentence in a future dialogue. The efficiency of the algorithm is related to the level of consistency between the content of the dataset and the researcher's domain of interest, i.e., if one asks how to detect phishing emails, then she needs a corpus of emails that gives insight into the question asked.

To the best of our knowledge, there is a lack of corpus composed of realized CSE attacks, in contrast with other types of attacks like network intrusions, because CSE dialogues are rarely going public. Furthermore, a CSE corpus should be of sufficient quantity so the algorithms can be trained well and be efficient. In case the collected dialogues are not of sufficient quantity, researchers apply resampling techniques [9,10] to enlarge the corpus, or they use annotation [11] to add useful in-context information as metadata to every dialogue. The metadata information augments the algorithms' effectiveness by providing pointers to what is important in a corpus. The ML/NLP algorithms extrapolate rules from the metadata provided in order to apply those rules later to unannotated text dialogues. An annotation task frequently employs a custom XML representation scheme depending on the context used. While there are some well-known annotation schemes, there is no universal standard for annotating dialogues. Furthermore, there are multiple ways for this metadata information to be stored, either as inline annotation (metadata are stored in same file as the dialogue), or stand-off annotation (metadata are stored in a separate file).

This work is targeting to answer specific research questions by investigating:

- The feasibility of collecting and building a CSE attack corpus to address the lack of training data composed of realized CSE attacks.
- The feasibility of annotating the linguistic characteristics of social engineers' language in order to constitute an adjuvant factor for the ML algorithms' training.

- The usage of the produced CSE corpus towards an early-stage automated CSE attack recognition by training a NER system that is able to identify critical in-context terms.

The paper is structured as follows. The current trends and work of utilizing ML/NLP algorithms used for social engineering attack recognition are described in Section 2. Section 3 presents the proposed approach comprised of a conceptual framework and the methodology followed to build and annotate a specific-purpose CSE corpus. The application of the methodology and the presentation of the results is presented in Section 4. Section 5 presents the evaluation of the annotated CSE corpus on training a Named Entity Recognition (NER) system based on a bi-LSTM neural network. The paper concludes in Section 6.

2. Related Work

In Lansley et al. [12–14], the authors use NLP methods and neural networks in an attempt to detect social engineering attacks. They describe a method where offline or online text documents are processed using NLP tools and through artificial neural networks, they discriminate whether the text is a social engineering attack or not. The text in the first stage is parsed and checked for syntactical/grammatical errors using natural language techniques while later an artificial neural network classifies possible social engineering attacks. The proposed method has presented high accuracy results during the evaluation where a real and a semi-synthetic dataset were used for the model training. Furthermore, several other classification models have been tried to compare the two datasets.

In [15] by Catak et al., a URL classifier is described based on Random Forest models and gradient boosting classifier. The URL classifier, in order to improve the algorithm's efficiency to detect malicious web sites, makes use of features related to the host and the linguistic characteristics of the URL. By using machine learning algorithms, the authors succeed to drastically reduce the detection time of a malicious URL. Thus, they offer real-time protection for harmless web browsing while at the same time saving computational resources.

Peng et al. [16] present a method to detect malicious statements using natural language processing techniques. They analyse the statements and discriminate the malicious ones that imply a phishing attack. The malicious intent of the statements is detected by analysing the statements. The proposed algorithm is evaluated using a phishing email dataset that is used as a benchmark set.

Lee et al. in [17] describe a method to represent the syntactic and semantic characteristics of the natural language by using a pre-trained BERT model. The proposed model seems to be resilient to adversarial attacks where the attackers intentionally replace the keywords with synonyms.

Lan [18] proposes a social engineering detection model based on deep neural network. The model is able to detect deception attempts and phishing attempts by analysing the text content. The chat history in the first stage is processed and analysed using natural language techniques and the context semantics are captured and mined using a bi-directional Long Short-Term Model (bi-LSTM). The integration of the user characteristics and chat content characteristics as features for the classification is done by ResNet.

In [19], the authors detect malicious documents using a framework that is based on machine learning algorithms. A Random Forests ensemble classifier is used which selects in random the features for each classification tree. The features are selected and extracted from the document's metadata and structure. This Random Forests classifier appears to be resilient against mimicry attacks and highly efficient in detection rate.

Chatbots can also be utilized for the purpose of recognizing a malicious act such as a CSE attack. In [20], the authors describe a system that protects against CSE attacks by deploying a pipeline of NLP components. The NLP components include NER, Dialogue Engineering, Stylometry and Ask and Framing Question. They use an active defence approach to detect the social engineer's intention and then waste her time and resources. In [21], the authors also present a different approach with a chatbot that is dedicated to

educate the users and raise their awareness regarding the social engineering attacks. The chatbot first performs an assessment of cybersecurity concepts knowledge of the user using a quiz, and then based on the answers it proposes specific training paths to fill the knowledge gaps. During the education, the chatbot makes use of malicious questions, in an attempt to extract sensitive data from the users to make them more aware of the possible dangers.

Recent trends in social engineering attack recognition utilize AI, ML and NLP tools and techniques to empower their efficiency. However, the lack of a dedicated CSE training dataset is an inhibitory factor. There is a need for a methodologically built and annotated CSE corpus that will be able to train a multitude of algorithms for the purpose of CSE attack recognition. Such a corpus that will focus on the syntactic and semantic characteristics of the adversaries' language is expected to broaden our understanding of the malicious patterns used.

3. The Proposed Approach

The proposed approach contains a conceptual framework for studying CSE attacks and a methodology of building and annotating a CSE corpus.

3.1. Conceptual Framework

According to ENISA [22], the ISO/IEC 15408-1:2009(E) standard [23] is commonly used as a resource for evaluating the security of IT products and systems, and it is also used for procurement decisions with regard to such products by providing an abstract cybersecurity concept map. This concept map is focused on the protection of assets from threats. Threats are related to malicious or other human activities that seek to abuse the assets. In this high-level concept map, the cybersecurity concepts are interconnected through relationships depicting the strong interrelation between them.

After a thorough text analysis of the collected raw dialogues that is presented in Section 4, we concluded that social engineers were trying to extract information that can be discriminated using the following three categories of sensitive data:

- Personal: Details related to the employee, e.g., first or last name, telephone number etc.
- IT: Details related to the Information Technology ecosystem, e.g., Network-share names, hardware/software characteristics, etc.
- Enterprise: Details related to the SME ecosystem, e.g., Department names, Office numbers, etc.

Fitting the CSE attack concept in the ENISA's concept map by adding the aforementioned findings, and at the same time by narrowing down our focus in SME environments, we enriched the concept map with valuable in-context details related to our goal. We introduced (see Figure 1) seven new concepts that are necessary to describe the CSE attacks domain. Using this concept map we can see that the main concern for *SMEs* is to safeguard their *assets* from *threats*. *Threat agents*, such as *Social Engineers*, give rise to these threats by exploiting *vulnerabilities* leading to greater *risk* for the assets. It is the responsibility of the SMEs to safeguard the assets where, in the case of CSE attack the most critical asset is the *Sensitive Data* that can leak. In addition, sensitive data can include *Personal*, *IT* or *Enterprise* details.

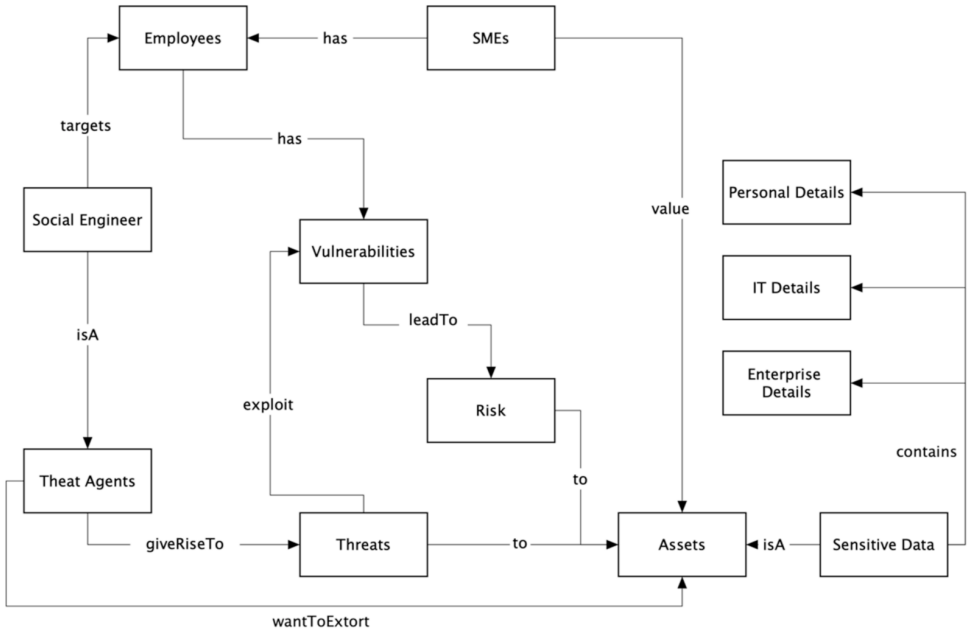

Figure 1. The concept map of CSE Attack.

3.2. Methodology

The proposed methodology for building and annotating a CSE corpus contains two phases, each comprising of several steps, as described below:

- PHASE 1: CSE Cor*pus Building*
  - STEP1-*Sources Selection*: the CSE attack dialogue sources are identified.
  - STEP2-*Dialogues Collection:* the CSE attack dialogues are collected through manual and automated web scraping methods.
  - STEP3-*Dialogues Enrichment*: Dialogues enrichment techniques are selected and applied.
  - STEP4-*Linguistic Analysis:* Dialogues are analyzed from a linguistic perspective.
  - STEP5-*Dialogues Processing*: the dialogues are processed using NLP techniques to form the CSE corpus
- PHASE 2: CSE Cor*pus Annotation*, based on the steps described in [24].
  - STEP1-*Goal Declaration:* the goal of the annotation task is declared.
  - STEP2-*Model and CSE Ontology Creation*: the phenomenon in study is represented in abstract terms and an ontology is created to be used as the base for the annotation task.
  - STEP3-*Specifications Definition:* a concrete representation of the model is created based on the CSE ontology.
  - STEP4-*Annotator Guidelines:* a blueprint is produced to help annotators in element identification and element association with the appropriate tag.
  - STEP5-*Annotation Task:* the annotation process is performed. In case of changes in the CSE ontology the process returns recursively to STEP2.
  - STEP6-*Inter-Annotator Agreement:* the inter-annotator agreement is validated.
  - STEP7-*Adjudication Task:* the final version (gold standard) of the annotated CSE corpus is finally formed.

4. Application and Results

The application of the proposed methodology is presented in the first two sub-sections followed by the presentation of the resulting CSE corpus.

4.1. CSE Corpus Building

4.1.1. Source Selection

To create the CSE corpus, we first had to locate the sources to retrieve CSE attack dialogues. For that purpose, we initially started by gathering synonyms and similar search terms for the word *dialogue*. Using the *word embedding* [25] technique, similar terms were identified and selected based on a ranking of the most used words in context. The words that were selected were: *dialog*, *dialogue*, *chat*, *conversation*, and *discourse.* Additionally, the following terms in the CSE context were used: *discourse analysis*, *data leakage*, *sensitive data exposure*, *corpus*, *dataset*, *online chat*, *instant messenger*, *excerpts*, *conversation transcripts*. Using combinations of the aforementioned search terms, we located several websites, books, logs, and forums with relevant content. The discovered sources belong to one of the following categories:

- Social engineering dark websites (tutorials, guides, and others)
- Social engineering dark forums (text dialogues)
- Social engineering books
- Instant Messaging logs
- Telephone conversations

The following table (Table 1) presents the top twenty web sites, forums and books used to acquire dialogues useful in producing the CSE corpus.

Table 1. Top twenty sources for collecting CSE attack dialogues.

No#	Web Sites/Forums
1	'What is the bloody point?' https://www.whatsthebloodypoint.com/ (accessed on 15 November 2021)
2	'Social Engineering', Nulled. https://www.nulled.to/forum/69-social-engineering/ (accessed on 15 November 2021)
3	'Sinisterly-Social Engineering' https://sinister.ly/Forum-Social-Engineering (accessed on 15 November 2021)
4	'Ripoff Scams Report | Consumer Complaints & Reviews | Ripandscam.com'https://www.ripandscam.com/ (accessed on 15 November 2021)
5	MPGH-MultiPlayer Game Hacking & Cheats https://www.mpgh.net/forum/forumdisplay.php?f=670 (accessed on 15 November 2021)
6	'Home', BlackHatWorld. https://www.blackhatworld.com/ (accessed on 15 November 2021)
7	'Hack Forums' https://hackforums.net/index.php (accessed on 15 November 2021)
8	'Demonforums.net', demonforums.net. https://demonforums.net/ (accessed on 15 November 2021)
9	419 Eater-The largest scambaiting community on the planet! https://www.419eater.com/ (accessed on 15 November 2021)
10	Socialengineered Forum https://web.archive.org/web/20200119025819/https://socialengineered.net/ (accessed on 15 November 2021)
11	SE Forums https://web.archive.org/web/20180401044855/https://seforums.se/ (accessed on 15 November 2021)
12	Leak Forums Net https://web.archive.org/web/20190123070424/https://leakforums.net/ (accessed on 15 November 2021)

Table 1. *Cont.*

No#	Web Sites/Forums
	Books
13	Advanced Persistent Threat Hacking: The Art and Science of Hacking-Tyler Wrightson
14	G. Watson, A. Mason, and R. Ackroyd, Social Engineering Penetration Testing: Executing Social Engineering Pen Tests, Assessments and Defense. Syngress, 2014
15	C. Hadnagy, Unmasking the Social Engineer: The Human Element of Security. John Wiley & Sons, 2014
16	M. I. Mann, Hacking the Human: Social Engineering Techniques and Security Countermeasures. Gower Publishing, Ltd., 2012.
17	K. D. Mitnick and W. L. Simon, The Art of Deception: Controlling the Human Element of Security. John Wiley & Sons, 2011.
18	K. Mitnick, Ghost in the Wires: My Adventures as the World's Most Wanted Hacker. Hachette UK, 2011.
19	J. Long, No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing. Syngress, 2011.
20	C. Hadnagy, Social Engineering: The Art of Human Hacking. John Wiley & Sons, 2010.

4.1.2. Dialogues Collection

To perform the collection of dialogues [26] a cloud-based infrastructure was established to host all the required software services and custom scripts. More specifically, the infrastructure was deployed in a Cloudstack environment by provisioning virtual machines with discrete roles. A dissection of the infrastructure based on functionality is illustrated in Figure 2. The *Corpus Building* section includes the *n Web Scrapers* used, which host custom scripts that scan and collect text from different web sources. The Web Scrapers are the only members of the infrastructure that communicates with the Internet to locate dialogues. This section also hosts every server that stores information at all stages of the project. The *Raw Content* server contains the scraped content before any preprocessing, while the *Database* server and the *Corpus* server contain selected information based on specific criteria and the final corpus, respectively. The *Processor* server applies the custom scripts in different processing stages and stores the results in the appropriate target. Finally, the *Corpus Annotation* section includes the workstations of the cybersecurity experts acting as annotators.

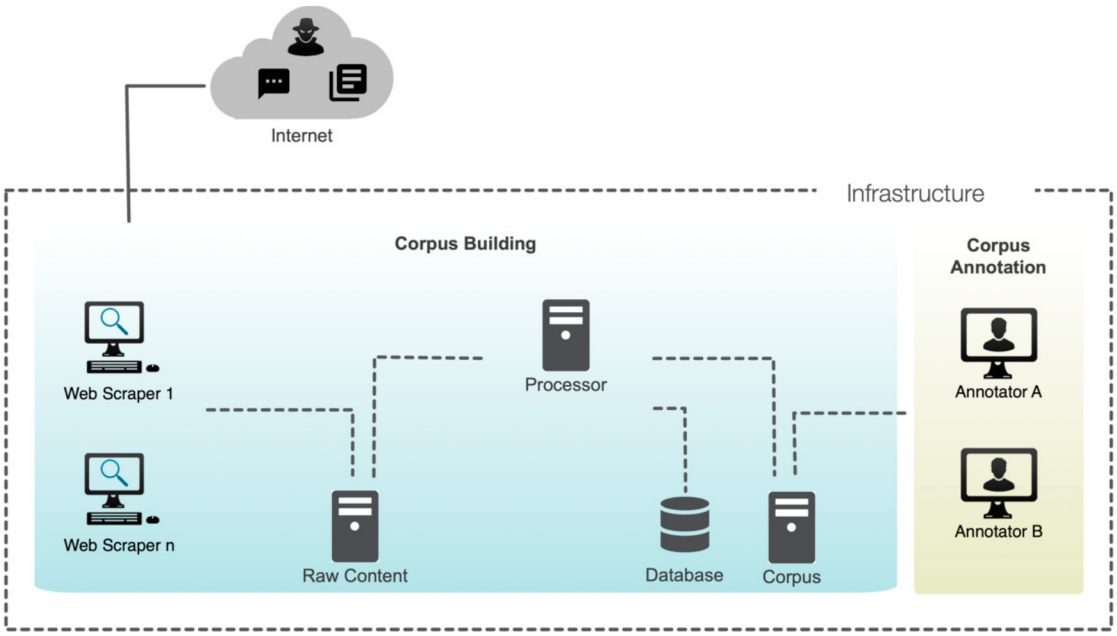

Figure 2. Infrastructure Setup.

4.1.3. Dialogues Enrichment

As mentioned earlier, corpus can be small in size, and this can have a negative influence on ML/NLP algorithms' efficiency. To enrich the collected dialogues, we parsed and created a list of 200 critical nouns belonging to one of the three sensitive data categories. Afterwards, for each noun, ten new sentences were added where the noun identified as critical was substituted by a similar word. Thus, the dialogues were enriched by 2000 new sentences. The discovery of similar words was based on the Word Embeddings technique [25], that is used to capture the meaning of the words using a dense vector representation. Each point in the embedding space represents a word and, based on the surrounding words of the target word, these points are learned and moved around. We used a pre-trained word2vec model to discover the synonyms based on the *distributional hypothesis* that linguistic items, e.g., words with similar distributions in a specific context (dialogue/document), have a similar meaning. This way, a segregation of the different domain words is being created using their vector values. Words with similar meanings were grouped due to their similar distribution on the dialogues. We created a vector space where each unique word in a dialogue is assigned a corresponding vector. Hence, the vector space is a vector representation of the words in the collected dialogues.

The individual dimensions of these vectors have no inherent meaning. However, it is the overall patterns of location and distance between vectors that ML/NLP algorithms take advantage of. For example, the application of word embedding technique in order to locate the ten most similar words to the word 'password' gave us the list illustrated in Figure 3:

```
['password'] [
    ('passwords', 0.769),
    ('passphrase', 0.669),
    ('login', 0.644),
    ('Password', 0.637),
    ('passcodes', 0.637),
    ('logon', 0.632),
    ('username_password', 0.619),
    ('Passwords', 0.615),
    ('logon_credentials', 0.609),
    ('passphrases', 0.602)
]
```

Figure 3. Ten most similar words of 'password' term.

One can notice that the word 'password' and the word 'logon have a Euclidean distance of 0.632 while the word 'password' and the word 'passphrases' have a smaller Euclidean distance of 0.602.

4.1.4. Linguistic Analysis

The linguistic analysis was performed according to the following levels of observation (see Figure 4). A sample of the CSE dialogues was analysed from a linguistic perspective and served as a baseline to ensure that the software tools and libraries were able to achieve the desired level of quality. Initially, as seen in Figure 4, a *lexical analysis* was performed by breaking down a sentence into words, phrases, or other meaningful part, a task known as *chunking*. Afterwards, a *morphological analysis* was performed where the structure and the form of each word was the main concern. Part-of-speech tagging (POS), stems and lemmas were identified, and a *syntactic analysis* followed that focused on grammar and syntax patterns. Subsequently, the meaning of the words and phrases were examined, where the semantics of words and phrases were investigated. The most crucial step, though, was the *pragmatic analysis* that took place to identify the *actual meaning* of the utterances. This is reasonable because an automation tool cannot understand the hidden intent of a speaker or a writer.

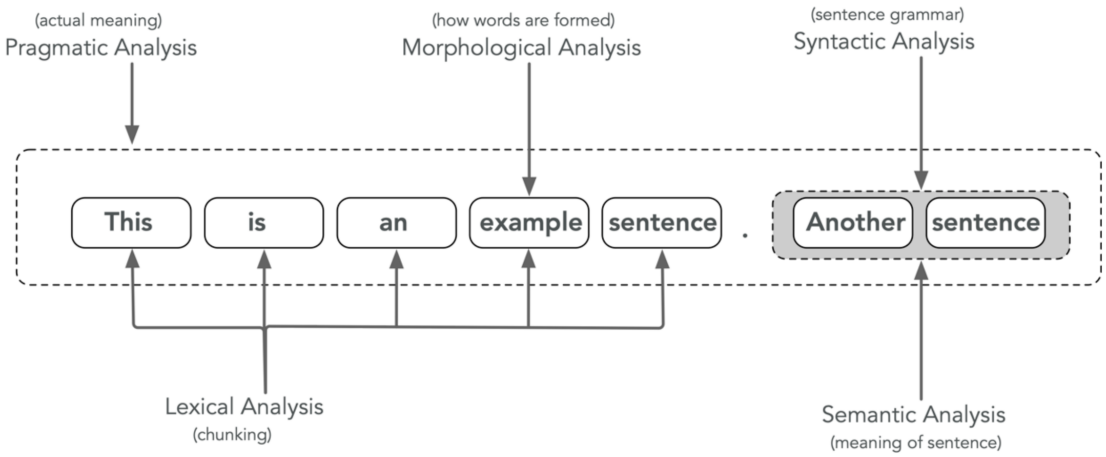

Figure 4. Example of linguistic analysis.

4.1.5. Dialogues Processing

We used the text processing workflow illustrated in Figure 5 to prepare the collected CSE dialogues. At first, all dialogues were converted to use the UTF-8 encoding scheme. Slang and abbreviations were removed or substituted with corresponding phrases using open-source slang databases Using dictionaries and the Textblob library [27] we performed spelling correction and removed any emojis. Empty lines, specific stopwords and specific punctuation marks were removed using traditional NLP libraries like NLTK [28] and spaCy [29]. Moreover, all HTML or other programming code and URLs or paths were stored in a separate database and substituted with *_code_*, *_url_*, and *_path_* placeholders in the dialogue. Any illegal characters were also stripped and all text transformed to lowercase.

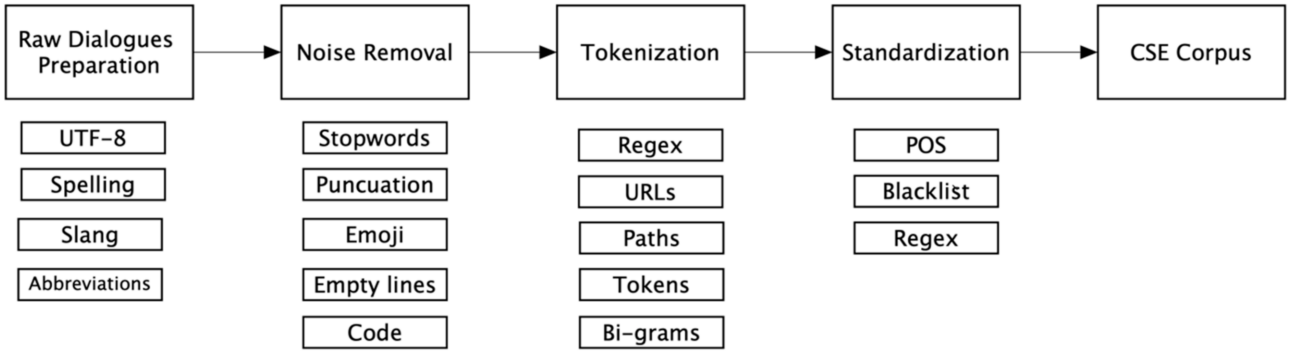

Figure 5. The CSE dialogues processing workflow.

The standard Penn Treebank [30] tokenisation rules were utilised for sentence tokenisation, and finally, standardisation processes using regular expressions and lookup tables were applied to tune the CSE dialogues. Figure 5 depicts the dialogues processing workflow where each stage, along with the individual tasks below, is shown.

At the end of this step, a corpus composed of CSE dialogues was formed. Table 2 presents the summary of the produced corpus, named *CSE corpus*.

The CSE corpus is composed of realized and fictional CSE attack dialogues. The existence of fictional attack dialogues does not affect its quality and capability because, similarly, a social engineer does not always act spontaneously but frequently prepares a fictional scenario (pretext attack) to guide the conversation and unleash his attack. The same applies to the CSE corpus, which incorporates a combination of confirmed and fictional CSE attacks.

Table 2. CSE corpus summary.

Corpus Name	CSE corpus
Collection methods	Web scraping, pattern-matching text extraction
Corpus size	(N) 56 text dialogues/3380 sentences
Vocabulary size	(V) 4500 terms
Content	chat-based dialogues
Collection date	June 2018–December 2020
Creation date	June 2021

4.2. CSE Corpus Annotation

Following the methodology presented in Section 3.2, we initially made a plan and set the goal that we wanted to achieve by annotating the CSE corpus. Afterwards, a simple model was developed to represent the annotation task in abstract terms and a CSE ontology was created to represent the in-context entities and their interconnection. Next, a specifications file was created where the entities and interconnections were described in a formal way and become tags. This file, along with the corpus guidelines, was given to the annotators to perform the annotation task. After the annotators finish their job, an inter-annotator agreement assessment took place using Cohen's Kappa metric, and the gold standard version of the CSE corpus is finally produced.

4.2.1. Goal Declaration

The annotation goal was to create the appropriate semantic target to facilitate the CSE attacks recognition by assigning the correct tag to in-context words in a sentence. It was crucial to label all related words or sequences of words or text-spans in CSE attack context so we can later perform efficient NER or text classification. Each word or text span was labelled with a type identifier (tag) drawn from a vocabulary created based on the CSE ontology and indicated *what* various terms denote in the context of a CSE attack and *how* they interconnect between them.

As mentioned before, SME employees are vulnerable to sensitive data leakage of type *Personal*, *IT*, and *Enterprise* details. Moreover, ML/NLP algorithms are more efficient to recognise terms that belong to abstract classes than terms that belong to more specific subclasses. Thus, our aim during annotation was to identify and assign the correct ontology tag to words or text based on CSE ontology.

Further refinement of the goal resulted in defining the following objectives:

- Identify keywords, syntax, and semantic characteristics to detect *Personal* data leakage. If found, label with appropriate tag.
- Use keywords, syntax, and semantic characteristics to detect *IT* data leakage. If found, label with appropriate tag.
- Identify keywords, syntax, and semantic characteristics to detect *Enterprise* data leakage. If found, label with appropriate tag.
- Identify noun-verb combinations and verb repetitions based on blacklists.

4.2.2. Model and CSE Ontology Creation

An abstract model that practically represented the aforementioned goal was defined. A three-category classification (Personal, IT, Enterprise) was introduced to be the basis of this abstract model for identifying CSE-related terms in a dialogue. Our model M consists of a vocabulary of terms T, the relations between these terms R, and their interpretation I. The triple in Formula (1) represents our model

$$M = < T, R, I > \quad (1)$$

where:

T = {CSE_Ontology_term, Personal, Enterprise, IT}
R = {CSE_Ontology_term:: = Personal | Enterprise | IT}
I = {Personal= "list of personal terms in vocabulary", IT = "list of Information Technology terms in vocabulary", Enterprise = "list of enterprise/corporate terms in vocabulary"}

The CSE ontology was based on the concept map presented in Section 3.1 and is asset-oriented as an *asset* is defined in cybersecurity ontologies [31]. This ontology will support our work by grouping similar in-context concepts and relations so that we can later use these categorised hierarchies for our benefit. This CSE ontology, in order to be useful must meet the following requirements:

- The ontology should focus on assets (sensitive data) that could leak from an SME employee.
- The ontology should include only the necessary concepts in-context.
- The ontology should not exceed three levels of depth because that would lead to difficulties for algorithms and annotators to recognise the concepts.

The ontology was created in a semi-automated manner using a custom Information Extraction System (IES) to acquire structured knowledge about the related concepts. Several documents (Corporate IT Policies, IT professionals' CVs, ICT Manuals, and others) were used as input to the IES. Subsequently, regular expressions rules were used to extract related concepts and relations. Figure 6 illustrates the process of the IES workflow.

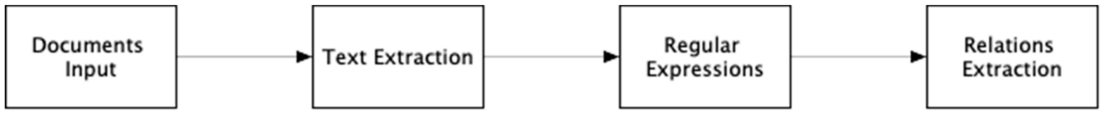

Figure 6. The IES workflow.

The proposed CSE ontology extends the ontology presented in [32], which connects the social engineering concepts with cybersecurity concepts. The reason is that we are interested in protecting sensitive data leakage in *personal*, *IT* and *enterprise* context. The implemented ontology was finalised using Protégé [33] software.

An excerpt of the CSE ontology created in Protégé, is illustrated in Figure 7 along with the arc types.

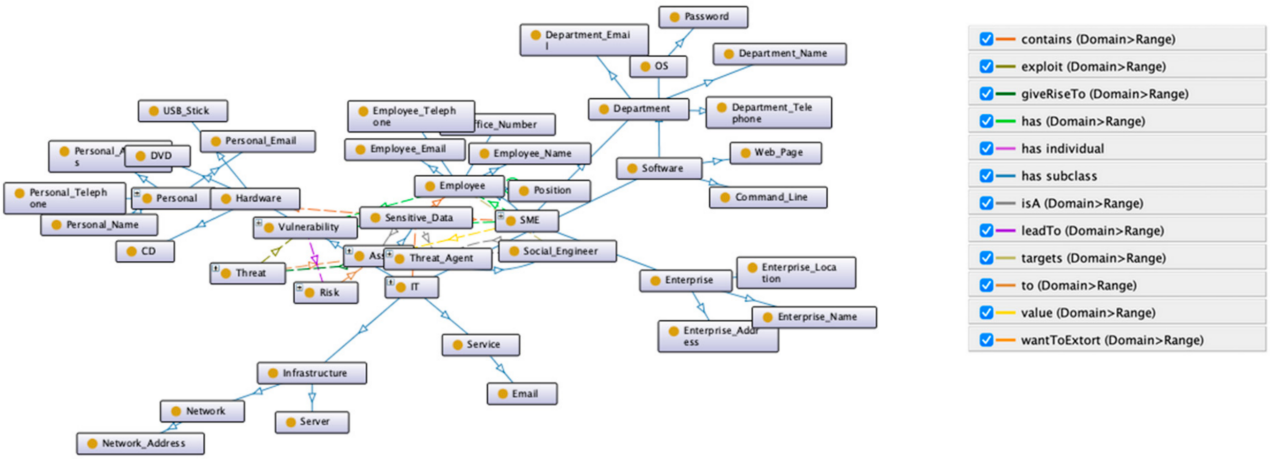

Figure 7. Excerpt of the proposed CSE ontology.

In the following Figure 8, the core concepts of the proposed CSE ontology are presented.

```
Sensitive Data ⊑ Asset                    Personal Address ⊑ Personal
Hardware ⊑ IT                             Personal Name ⊑ Personal
Infrastructure ⊑ IT                       Personal Telephone ⊑ Personal
Service ⊑ IT                              Risk
Software ⊑ IT                             Department ⊑ SME
Personal Email ⊑ Personal                 Employee ⊑ SME
Threat                                    Enterprise ⊑ SME
Vulnerability                             Social Engineer ⊑ Threat Agent
```

Figure 8. Core concepts.

Figure 9 presents an excerpt of the CSE ontology's object properties.

```
contains                                  isA
∃ contains Thing ⊑ Information            ∃ isA Thing ⊑ Social Engineer
⊤ ⊑ ∀ contains SME                        ∃ isA Thing ⊑ Information
⊤ ⊑ ∀ contains Employee                   ⊤ ⊑ ∀ isA Threat Agent
⊤ ⊑ ∀ contains Personal
⊤ ⊑ ∀ contains IT                         ⊤ ⊑ ∀ isA Asset

exploit                                   leadTo
∃ exploit Thing ⊑ Threat                  ∃ leadTo Thing ⊑ Vulnerability
⊤ ⊑ ∀ exploit Vulnerability               ⊤ ⊑ ∀ leadTo Risk

giveRiseTo                                targets
∃ giveRiseTo Thing ⊑ Threat Agent         ∃ targets Thing ⊑ Social Engineer
⊤ ⊑ ∀ giveRiseTo Threat                   ⊤ ⊑ ∀ targets Employee

has                                       to
∃ has Thing ⊑ Employee                    ∃ to Thing ⊑ Threat
∃ has Thing ⊑ SME                         ∃ to Thing ⊑ Risk
⊤ ⊑ ∀ has Employee                        ⊤ ⊑ ∀ to Asset
⊤ ⊑ ∀ has Vulnerability
                                          value
wantToExtort                              ∃ value Thing ⊑ SME
∃ wantToExtort Thing ⊑ Threat Agent       ⊤ ⊑ ∀ value Asset
⊤ ⊑ ∀ wantToExtort Asset
```

Figure 9. Excerpt of Object properties.

4.2.3. Specifications Definition

The produced specification file holds the annotation schema and uses XML DTD [34] representation. It describes the model and the CSE ontology by turning the abstract ideas and entities/relations into tags and attributes. Following a clear and simple approach for the specifications file helps to build better and more accurate prediction models at a later stage. An excerpt of the produced specification file is depicted in Figure 10:

```
<!ENTITY name "CSE-Corpus_v1.0">
<!ELEMENT actor (#PCDATA)>
<!ATTLIST actor id ID prefix="ACTOR_" #REQUIRED>
<!ATTLIST actor role (attacker|victim)>
<!ELEMENT personal (#PCDATA)>
<!ATTLIST personal id ID prefix="PERSONAL_" #IMPLIED>
<!ATTLIST personal type (Personal_Address|Personal_Email|Personal_Name|Personal_Telephone)>
<!ELEMENT enterprise (#PCDATA)>
<!ATTLIST enterprise id ID prefix="ENTERPRISE_" #IMPLIED >
<!ATTLIST corporate type (Enterprise_Address Enterprise_Location|Enterprise_Name )>
<!ELEMENT it (#PCDATA)>
<!ATTLIST it id ID prefix="IT_" #IMPLIED>
<!ATTLIST it type (Hardware| Infrastructure |Service |Software)>
```

Figure 10. Excerpt of the specifications file.

4.2.4. Annotator Guidelines

Two cybersecurity experts with heavy knowledge of the SMEs ecosystem were selected as the annotators and assigned the task to perform annotation at the sub-sentence level. The annotators were also members of the annotation schema development team. They performed a stand-off annotation by character location because, in this case, the metadata type tags are stored separately from the original chat text. This way, we avoided changing the original text format, and reading the original text could be done more efficiently. Most of all, we will be able to deal with overlapping tags if different annotation schemes are to be merged in the future.

Over twenty different annotation tools (open source and commercial) were tested during the annotation project. We decided to use GATE [35] and Prodigy [36] due to their completeness and capability of building complete natural language pipelines of NLP tools.

Annotators were given guidelines to direct them about tagging the important terms in a compliant way based on our schema. Several examples were also given to help resolve ambiguous term cases. First, the different tags were explained, and prioritisation guidelines were given for the case of an ambiguous term that could belong in more than one category. For example, if an outsider asks for the (First or Last) Name of a department's supervisor, the answer can be tagged by Personal and Enterprise tag simultaneously. Annotators were advised to follow this order:

1. prefer IT category over (Enterprise category or Personal category)
2. prefer Enterprise category over Personal category

This priority order ensures that IT details have greater importance over all other categories, and Enterprise details have greater importance over Personal details.

More specifically, an excerpt from the annotators guidelines follows:

1. Each text span may be tagged with *Personal*, *IT* or *Enterprise* main tags
2. If a sentence has entities that can fit in two or more tags, then follow the following priority IT > Enterprise > Personal
3. Prefer individual words to word combinations
4. The tag *Personal* is assigned to whatever information is related to a person. e.g., first name, Last Name. Consult CSE ontology
5. The tag *IT* is assigned to whatever information is related to Information Technology, e.g., USB stick, computer, software and others. Consult Ontology
6. The tag *Enterprise* is assigned to whatever information is related to enterprises and enterprise environment, e.g., Department, office, positions, resumes and others. Consult CSE ontology
7. A sentence can have no important words. This sentence is call Neutral
8. Noun-verb combinations and verb repetitions will be identified automatically based on blacklists

Table 3 lists a sample of the examples that were given to the annotators where the important terms are in italics, and the tags assigned to each one of them is on the right column.

4.2.5. Annotation Task

The annotation task performed had the target to label the words of CSE corpus based on their semantic and syntactic characteristics. The two cybersecurity experts were responsible of labelling the words based on their semantic characteristics, and thus performed semantic annotation. By annotating the semantic characteristics of the words, the background information in each dialogue was linked with the CSE ontology. The syntactic characteristics of the words were labelled by the use of the specific-purpose annotation software we developed to perform syntactic annotation by extracting statistical information regarding the words used by social engineers. Table 4 presents the two different types of annotation and the specific answers that we were looking for.

Table 3. Important terms and tags.

No#	Sentence	Tag
1	You will find contact numbers on the *Intranet*	IT
2	Let me get that name again and give me your *employee number*	Enterprise
3	Our *project leader* is Jerry Mendel	Enterprise
4	I can cut some corners and save some time, but I'll need your *username* and *password*	IT
5	Okay, can you tell me again your *employee ID number*.	Enterprise
6	I'll just *hit reset* and the old one will be wiped out	IT
7	For *account identification* may you please provide your *account number*	IT
8	Tom, Its Eddie . . . go ahead and try your *network connection*	IT
9	May I have your *full name* and *email address* please	Personal
10	Rest assured that they will be able to read the chat transcript as well as the documentation in your case	-

Table 4. Questions per annotation type.

Syntactic	Semantic
What Part-of Speech (POS) type is it?	Is it a CSE ontology entity?
Is it a bi-gram?	What role does it play in the CSE ontology?

The syntactic characteristics that are annotated are the POS type, and the bi-grams which denote the syntactic relationship between pairs of words. By tagging the POS type and bigrams we were able to perform subsequent phrase structure analysis and extract the dependency trees of the corresponding phrases. This extra information can be used by parsers for Named Entity Recognition, n-grams identification, and Bag-of-Words representation.

The labelled words or text spans were further processed in order to extract statistical information valuable for the task of CSE attack recognition. Furthermore, for each sentence, a plethora of counters was also attached as metadata. This metadata information included vocabulary words frequency, words sequence pattern, words context relativity, urgency indicators, number of URL or paths appeared, blacklisted verb and noun combinations, backlisted bi-grams (e.g., tech support, USB stick and others). Moreover, the sentences similarity was measured to be used as attacker persistence metric; i.e., if the attacker uses different sentences but the sentences are highly similar, this means that the attacker persists on his initial intent; this information counts also as metadata information for the annotation task.

GATE and Prodigy handled all of the annotation steps along with WordNet [37] and LIWC [38] software. Figure 11 illustrates a tree-structured taxonomy of the syntactic and semantic annotation targets.

The preferred encoding scheme to tag the existing chunks of text (word, or text spans) was based on the IOB format [39]. In the IOB encoding scheme, the "*I-*" prefix of a tag indicates that the tag is inside a chunk, "*O*" indicates that the tag does not belong to a chunk and the "*B-*" prefix of a tag indicates that a token is the beginning of a chunk.

For example, the sentence '*Mr. Robinson is the Head of Financial Department, and his office is on the second floor'* using the IOB format is presented in Table 5.

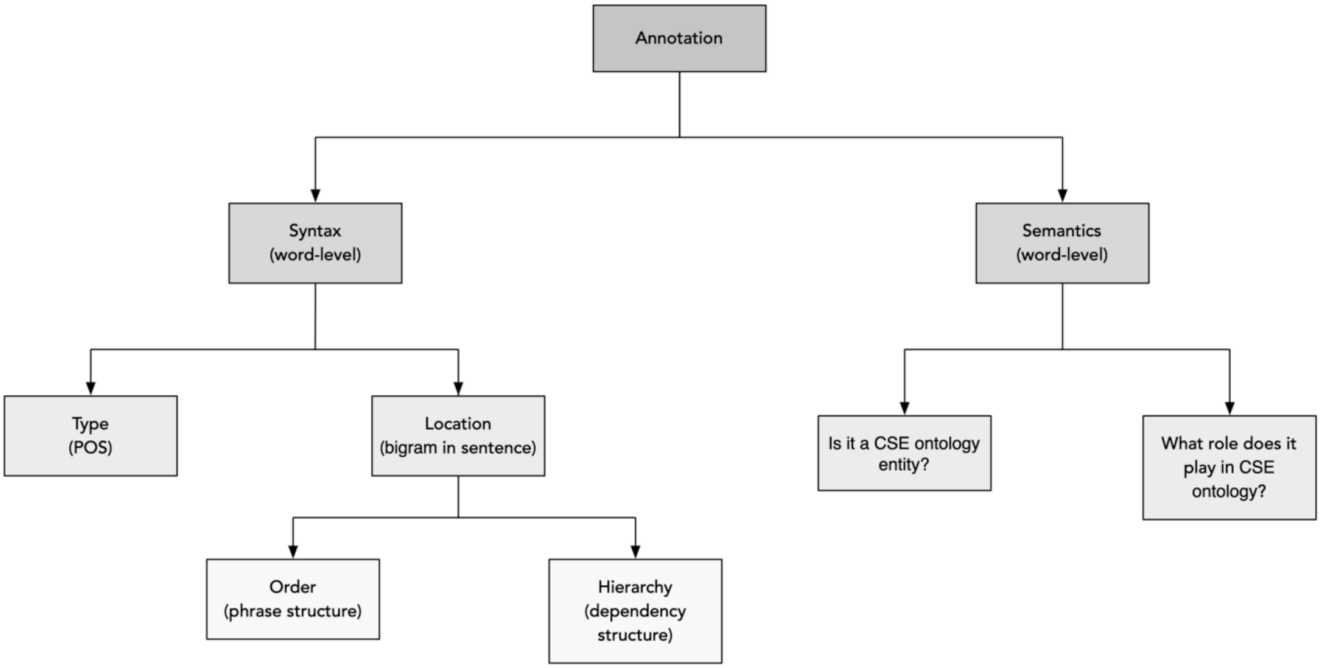

Figure 11. Annotation targets.

Table 5. IOB encoding example.

Chunk	Encoding
Mr	B-EMP-NAME
Robinson	I-EMP-NAME
is	O
the	O
head	I-POS
of	O
Financial	B-DEPT-NAME
Department	I-DEPT-NAME
and	O
his	O
office	I-DEPT
is	O
on	O
the	O
second	I-OFF-NUM
floor	I-OFF-NUM

During the annotation task, the words were labelled based on their semantic and syntactic characteristics. This way, relationships were extracted between words or text spans belonging to different branches of the ontology. Moreover, hidden patterns that attackers use in a conversation were discovered and valuable information about how different CSE concepts and ontology entities interact was extracted. For example, in Figure 12, where the semantic categories are labelled with tags in brackets, we can discover that an *Employee_Name* has an "IS" relation with *Position* and an "OF" relation with *Department_Name.*

```
Mr Robinson [Employee_Name] is the Head [Position] of Financial Department [Department_Name]
 and his office [Department] is on the second floor [Office_Number]
```

Figure 12. Discovery example.

4.2.6. Inter-Annotator Agreement

The inter-annotator agreement (IAA) was validated using Cohen's Kappa, which is one of the most popular statistics to measure the agreement between two annotators of N terms on m categories [40]. The formula to calculate Cohen's Kappa for two annotators is:

$$k = \frac{p_0 - p_e}{1 - p_e} = 1 - \frac{1 - p_0}{1 - p_e}$$

where p_0 is the relative observed agreement among annotators and p_e is the hypothetical probability of chance agreement.

The produced CSE corpus has N = 4500 terms and m = 3 categories, and both annotators (A and B) agreed for the *personal* category 1665 times, for the *enterprise* category 1442 times and for the *IT* category 1194 times. Table 6 contains a contingency matrix where each x_{ij} represents the multitude of terms that annotator A classified in category i, but Annotator B is classified in category j, with i, j = 1, 2, 3. The proportions on the diagonal, x_{ii}, represent the proportion of terms in each category for which the two annotators agreed on the assignment.

Table 6. Contingency matrix.

Annotator		**B**	**B**	**B**	**Total**
	Category	**personal**	**enterprise**	**IT**	
A	**personal**	1665	63	13	1741
A	**enterprise**	59	1442	31	1532
A	**IT**	14	19	1194	1227
Total		1738	1524	1238	4500

The observed agreement is calculated as: $P(o) = \frac{1665+1442+1194}{4500} = 0.956$ (95.6%).

The expected change agreement, thus the proportion of terms which would be expected to agree by chance is:

$$P(e) = \frac{\frac{(1741 \times 1738)}{4500} + \frac{(1532 \times 1524)}{4500} + \frac{(1227 \times 1238)}{4500}}{4500} = 0.339$$

so, we conclude that the Cohen's Kappa metric is, $k = \frac{p_0 - p_e}{1 - p_e} = \frac{0.617}{0.661} = 0.933$.

Interpreting the Cohen's kappa value of 0.933 [41], we can safely conclude that the level of agreement for the CSE corpus annotation task was almost perfect.

4.2.7. Adjudication Task

After the two annotators finished the annotation task, having succeeded an almost perfect level of inter-annotator agreement, the gold standard corpus was created by performing adjudication over the data. The gold standard corpus was the final annotated version of the CSE corpus. No annotator was part of the adjudication process. As expected, we did not encounter any difficulties due to the high level of annotators' agreement. Moreover, it is important to state that there were not any cases that annotators agreed and we, as adjudicators, had different opinion over the annotation tag. Thus, after the adjudication, the final *CSE corpus* was produced with the annotation information stored in a different file accompanying each CSE dialogue.

4.3. CSE Corpus Presentation

The annotated CSE corpus is composed of 56 dialogues, 3380 sentences. The words that are in context and can be tagged by the CSE ontology are 4500. Table 7 presents ten random utterances from the CSE corpus.

Table 7. Ten random utterances from CSE corpus.

No #	Utterance
1	which computer servers does your group use
2	do you sign in under the username Rosemary
3	ok i am trying to logon now
4	trace it back to where its plugged
5	it could save us both big headaches the next time this network problem happens
6	did you turn your computer off
7	thanks a lot please wait for my email
8	take care and bye for now
9	my flex 2 is charged but wont turn on is it normal
10	are you experiencing any computer issues presently

As depicted in Figure 13a, the distribution of the sentence categories based on the aforementioned tags shows that the produced CSE corpus is well balanced. Valuable information can also be extracted by observing the average word count per category, as seen in Figure 13b.

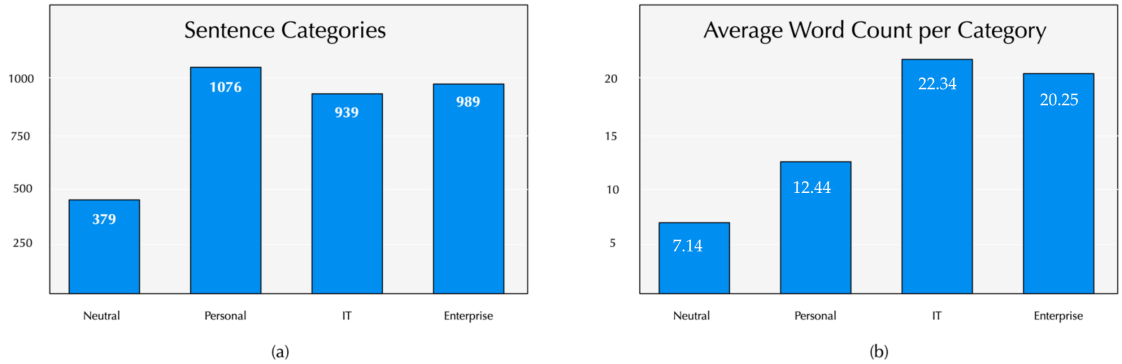

Figure 13. (**a**) Distribution of sentence categories, (**b**) Average word count per category.

Several interesting statistics were extracted that gave us information regarding the content of the sentences. Indicatively, we illustrate in Figure 14 the five most frequent words per category and in Figure 15 the five most frequent bi-grams per category.

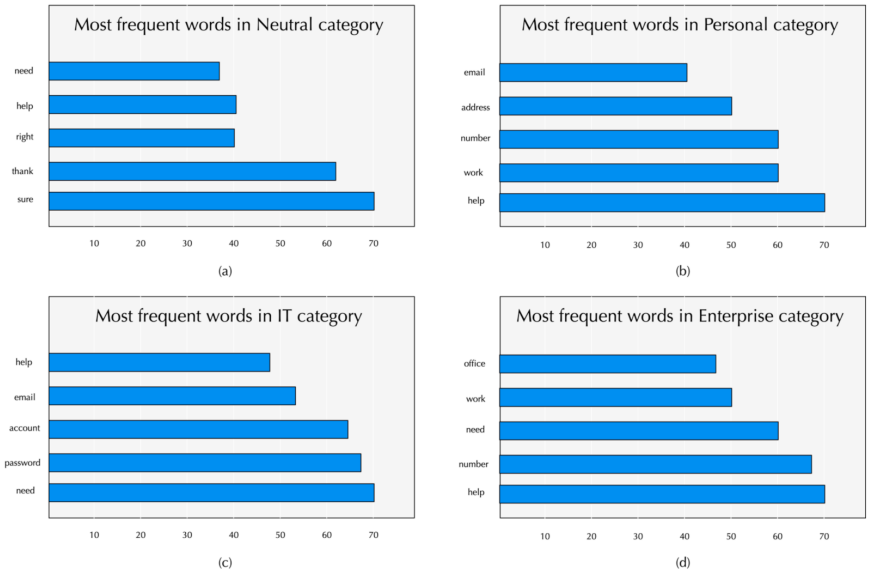

Figure 14. Five most frequent words per category. (**a**) Neutral, (**b**) Personal, (**c**) IT, (**d**) Enterprise.

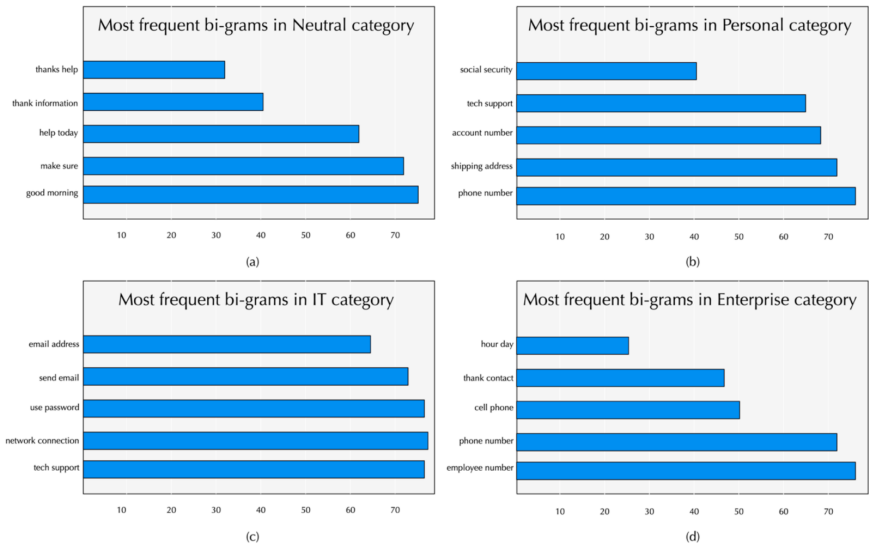

Figure 15. Five most frequent bi-grams per category. (**a**) Neutral, (**b**) Personal, (**c**) IT, (**d**) Enterprise.

5. CSE Corpus Evaluation

Named entity recognition (NER) [42] is a natural language processing method to extract in-context information from unstructured text data such as e-mails, web articles or chat-based dialogues. Using NER we can identify entities, like people, organizations, places etc., which exist in text. There are several well-known libraries like spaCy, Apache OpenNLP and TensorFlow with pretrained NER models that can be used to build a NER system that identifies the aforementioned entities. Furthermore, these pretrained models can be modified to recognize additional entities by training them with an annotated corpus that contains the additional tags. Such a corpus is the already presented CSE corpus, where personal, IT and enterprise information has been identified and tagged.

We evaluated the CSE corpus by building and testing a NER system trained with it. This NER system takes as input preprocessed CSE dialogue sentences and identifies, in each sentence, the words or text spans that can be labelled by a tag that is associated with an entity of the CSE ontology. This kind of task is a sequence-labelling task where a token is classified as belonging to one or none of the annotation classes. We used Keras [43] and TensorFlow [44] to build, train and evaluate a standard bi-directional Long Short-Term Memory neural network (bi-LSTM) [45] for CSE attack named entity recognition.

LSTMs were designed to overcome Recurrent Neural Networks (RNNs) inefficiency, since they fail to learn long dependencies due to bias toward their most recent inputs. RNNs are a family of neural networks that operate on sequential data; they take as input a sequence of vectors $(x_1, x_2 \ldots, x_N)$ and produce as output another sequence that represents information about every step in the input sequence. LSTMs incorporate a memory cell to capture long-range dependencies. Using several gates, LSTMs control the proportion of input that is given to the memory cell, as well as the proportion from the previous state that should be forgotten.

To build the NER system the following steps were taken:

1. *Preprocessing:* The CSE corpus has already been annotated in IOB format and thus each line of the corpus can easily be read and stored as a list of token-tag pairs. Then, each token was represented by a word embedding using a pre-trained English language model of the spaCy NLP library.
2. *Building:* Using Keras, a bi-LSTM model was constructed, which is comprised of two compound layers:
    - *Bi-directional LSTM layer*, where the forward and backward pass were encapsulated and the input and output sequences returned were stacked.

- *Classification layer*, where classification was performed to every position of the sequences in the stack. The SoftMax activation function was used to scale the output and obtain sequences of probability distributions.

3. *Training:* To train the model in Keras, a loss function for the model was specified to measure the distance between prediction and truth, and a batch-wise gradient descent algorithm was specified for optimization.
4. *Assessing:* The performance assessment of the model was conducted by applying the model to the preprocessed validation data. For each sample dialogue sentence and each token in a sentence, a tensor was obtained that contained a predicted probability distribution over the tags. The tag with highest probability was chosen, and for each sentence and each token the true and predicted tags were returned.

More specifically, inside the bi-LSTM neural network the following actions occurred in sequence:

1. Each dialogue sentence was split into a sequence of token-tag pairs.
2. Each token-tag pair was represented by a word embedding vector.
3. The word embeddings were provided by spaCy NLP library and pretrained to encode semantic information. This approach is known as Transfer Learning [46].
4. Going forward the bi-LSTM model at each step:
   a. the input vector was read and combined with the internal memory
   b. the output vector was produced and the internal memory was updated
5. This sequence of actions was performed by the bi-LSTM model to the input sequence and produced an output sequence of the same length.
6. Going backwards, the model read the input sequence again and produced another output sequence.
7. At each position, the outputs of steps 4 and 5 were combined and fed into a classifier which outputted the probability for the input word, at this position, that should be annotated with the Personal, IT or Enterprise tag.

Figure 16a,b below illustrate the training history that contains the loss and the accuracy achieved on training and validation after each epoch.

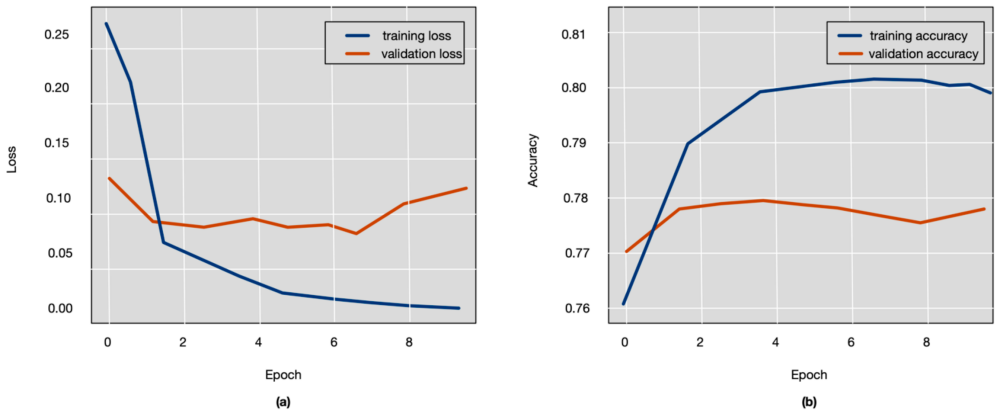

Figure 16. (**a**) Loss metrics, (**b**) Accuracy metrics.

Finally, accuracy, precision, recall and f1-score on the level of tag categories were calculated using the *scikit-learn* library. The following measures were used to assess the performance:

- *True Positives (TP)*: CSE ontology entities where the true tag is positive and whose category is correctly predicted to be positive.
- *False Positives (FP)*: CSE ontology entities where the true tag is negative and whose category is incorrectly predicted to be positive.
- *True Negatives (TN)*: CSE ontology entities where the true tag is negative and whose category is correctly predicted to be negative.

- *False Negatives (FN)*: CSE ontology entities where the true tag is positive and whose class is incorrectly predicted to be negative.

Using the above measures, we can calculate:

- *Accuracy*, which is the number of CSE ontology entities correctly identified as either truly positive or truly negative out of the total number of entities

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

- *Recall*, which identifies how many tags from the positive class are correctly detected

$$Recall = \frac{TP}{TP + FN}$$

- *Precision*, which identifies the frequency with which a model was correct when predicting the positive class.

$$Precision = \frac{TP}{TP + FP}$$

- *F1 Score*, which is the harmonic average of the precision and recall, and measures the effectiveness of identification when just as much importance is given to recall as to precision

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

Training a model for 10 epochs, we reached the scores presented in Table 8:

Table 8. bi-LSTM Performance Metrics.

Tag	F1	Precision	Recall	Support
PERSONAL	0.67	0.63	0.51	65
IT	0.86	0.93	0.64	122
ENTERPRISE	0.71	0.84	0.56	89

The results seem to be satisfactory, taking into account the relatively small size of the CSE corpus. A performance improvement at CSE attack NER can be achieved if we tune the model by:

- Enriching the CSE Corpus, i.e., adding sentences where more critical nouns belonging to one of the three sensitive data categories are substituted by similar words.
- Replacing the last feed-forward layer by a conditional random field (CRF) model,
- Reducing the imbalance of the tag distribution, e.g., by using a different loss function,
- Providing more input by tagging newcoming CSE dialogues and thus training more data.

6. Conclusions

In this paper, we have presented a methodology to build and annotate a CSE corpus. The application and results of the methodology have been demonstrated by producing the annotated CSE corpus composed of realized and fictional dialogues of CSE attacks. We have obtained satisfactory results after evaluating the CSE corpus by applying Named Entity Recognition. NER was performed using a bi-LSTM neural network that was trained by the CSE corpus.

The outcomes of this work broaden our understanding about the language used during a CSE attack and offer a tool that can be used to build defence mechanisms against social engineers. The CSE corpus has attached a plethora of in-context metadata and can be used for NER and various other types of text classification tasks. A lexicon-based approach

using a vocabulary and a weighting function can be combined with algorithms to enhance prediction capabilities even further.

Due to the difficulty of finding CSE attack dialogues, it was equally difficult to efficiently train ML algorithms to recognize CSE attacks. The CSE corpus is a corpus annotated with in-context terms that can be used to train not only NER systems but also many other ML algorithms for the purpose of detecting specific patterns based on interest (e.g., persuasion attempts, personality characteristics, etc).

There are, certainly, dimensions of the CSE attack phenomenon that are not well captured, and probably dependencies among terms in the data that are not identified simply because we currently cannot recognize them. Furthermore, due to the nature of the CSE attack, there are questions that remain unanswered, such as how the linguistic characteristics of the language used by social engineers can facilitate the automated recognition of specific behaviours that lead to a successful CSE attack. Further research is needed to investigate the capability of the CSE corpus to train additional text classification algorithms. In [47] persuasion [48,49], influence [50], human personality [51] and speech-act [52] have been identified as enablers of a successful CSE attack. Research into training recognizers of the aforementioned enablers is already underway.

Author Contributions: Conceptualization, N.T. and I.M.; methodology, N.T. and I.M.; software, N.T.; validation, N.T. and I.M.; writing—original draft preparation, N.T. and I.M.; writing—review and editing, N.T. and I.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository. The data presented in this study are openly available in GitHub at doi:10.5281/zenodo.5635627, reference number [26].

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

References

1. What Is "Social Engineering"? Available online: https://www.enisa.europa.eu/topics/csirts-in-europe/glossary/what-is-social-engineering (accessed on 29 October 2021).
2. Kumar, A.; Chaudhary, M.; Kumar, N. Social Engineering Threats and Awareness: A Survey. *Eur. J. Adv. Eng. Technol.* **2015**, *2*, 15–19.
3. Heartfield, R.; Loukas, G. A Taxonomy of Attacks and a Survey of Defence Mechanisms for Semantic Social Engineering Attacks. *ACM Comput. Surv.* **2016**, *48*, 1–39. [CrossRef]
4. Salahdine, F.; Kaabouch, N. Social Engineering Attacks: A Survey. *Future Internet* **2019**, *11*, 89. [CrossRef]
5. 2020 Data Breach Investigations Report. Available online: https://enterprise.verizon.com/resources/reports/dbir/ (accessed on 12 April 2021).
6. Adamopoulou, E.; Moussiades, L. An Overview of Chatbot Technology. In *Artificial Intelligence Applications and Innovations*; Maglogiannis, I., Iliadis, L., Pimenidis, E., Eds.; IFIP Advances in Information and Communication Technology; Springer International Publishing: Cham, Switzerland, 2020; Volume 584, pp. 373–383, ISBN 978-3-030-49185-7.
7. Cialdini, R.B.; Cialdini, R.B. *Influence: The Psychology of Persuasion*; Collins: New York, NY, USA, 2007; Volume 55.
8. Pogrebna, G.; Skilton, M. *Navigating New Cyber Risks: How Businesses Can Plan, Build and Manage Safe Spaces in the Digital Age*; Springer International Publishing: Cham, Switzerland, 2019, ISBN 978-3-030-13526-3.
9. Li, D.-C.; Lin, W.-K.; Chen, C.-C.; Chen, H.-Y.; Lin, L.-S. Rebuilding sample distributions for small dataset learning. *Decis. Support Syst.* **2018**, *105*, 66–76. [CrossRef]
10. Lateh, M.A.; Muda, A.K.; Yusof, Z.I.M.; Muda, N.A.; Azmi, M.S. Handling a small dataset problem in prediction model by employ artificial data generation approach: A review. In *Proceedings of the Journal of Physics: Conference Series*; IOP Publishing: Langkawi, Malaysia, 2017; Volume 892, p. 012016.
11. Wilcock, G. Introduction to Linguistic Annotation and Text Analytics. *Synth. Lect. Hum. Lang. Technol.* **2009**, *2*, 1–159. [CrossRef]
12. Lansley, M.; Polatidis, N.; Kapetanakis, S. SEADer: A Social Engineering Attack Detection Method Based on Natural Language Processing and Artificial Neural Networks. In *Computational Collective Intelligence*; Nguyen, N.T., Chbeir, R., Exposito, E., Aniorté, P., Trawiński, B., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2019; Volume 11683, pp. 686–696, ISBN 978-3-030-28376-6.

13. Lansley, M.; Mouton, F.; Kapetanakis, S.; Polatidis, N. SEADer++: Social engineering attack detection in online environments using machine learning. *J. Inf. Telecommun.* **2020**, *4*, 346–362. [CrossRef]
14. Lansley, M.; Kapetanakis, S.; Polatidis, N. SEADer++ v2: Detecting Social Engineering Attacks using Natural Language Processing and Machine Learning. In *Proceedings of the 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*; IEEE: Novi Sad, Serbia, 2020; pp. 1–6.
15. Catak, F.O.; Sahinbas, K.; Dörtkardeş, V. Malicious URL Detection Using Machine Learning. Available online: www.igi-global.com/chapter/malicious-url-detection-using-machine-learning/266138 (accessed on 12 April 2021).
16. Peng, T.; Harris, I.; Sawa, Y. Detecting Phishing Attacks Using Natural Language Processing and Machine Learning. In Proceedings of the 2018 IEEE 12th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 31 January–2 February 2018; pp. 300–301.
17. Lee, Y.; Saxe, J.; Harang, R. CATBERT: Context-Aware Tiny BERT for Detecting Social Engineering Emails. *arXiv* **2020**, arXiv:2010.03484.
18. Lan, Y. Chat-Oriented Social Engineering Attack Detection Using Attention-based Bi-LSTM and CNN. In Proceedings of the 2021 2nd International Conference on Computing and Data Science (CDS), Stanford, CA, USA, 28–29 January 2021; pp. 483–487.
19. Smutz, C.; Stavrou, A. Malicious PDF detection using metadata and structural features. In *Proceedings of the 28th Annual Computer Security Applications Conference*; Association for Computing Machinery: New York, NY, USA, 2012; pp. 239–248.
20. Dalton, A.; Aghaei, E.; Al-Shaer, E.; Bhatia, A.; Castillo, E.; Cheng, Z.; Dhaduvai, S.; Duan, Q.; Hebenstreit, B.; Islam, M.; et al. Active Defense Against Social Engineering: The Case for Human Language Technology. In Proceedings of the First International Workshop on Social Threats in Online Conversations, Understanding and Management, Marseille, France, 11–16 May 2020.
21. Saleilles, J.; Aïmeur, E. SecuBot, a Teacher in Appearance: How Social Chatbots Can Influence People. In *Proceedings of the 1st Workshop on Adverse Impacts and Collateral Effects of Artificial Intelligence Technologies—AIofAI 2021*; Aïmeur, E., Ferreyra, N.E.D., Hage, H., Eds.; CEUR: Montréal, Canada, 2021; Volume 2942, pp. 31–49.
22. ISO/IEC Standard 15408. Available online: https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/laws-regulation/rm-ra-standards/iso-iec-standard-15408 (accessed on 13 July 2021).
23. Evaluation criteria for IT Security—Part 1: Introduction and General Model—ISO/IEC 15408-1:2009. Available online: https://standards.iso.org/ittf/PubliclyAvailableStandards/c050341_ISO_IEC_15408-1_2009.zip (accessed on 13 July 2021).
24. Pustejovsky, J.; Stubbs, A. *Natural Language Annotation for Machine Learning*; O'Reilly Media: Sebastopol, CA, USA, 2013, ISBN 978-1-4493-0666-3.
25. Goldberg, Y.; Levy, O. word2vec Explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv* **2014**, arXiv:1402.3722.
26. Tsinganos, N. Tsinik/Applsci-1445637: Applsci-1445637 v1.0.0. Zenodo. 2021. Available online: http://doi.org/10.5281/zenodo.5635627 (accessed on 15 November 2021).
27. TextBlob: Simplified Text Processing—TextBlob 0.16.0 Documentation. Available online: https://textblob.readthedocs.io/en/dev/index.html (accessed on 11 May 2021).
28. NLTK: Natural Language Toolkit. Available online: https://www.nltk.org/ (accessed on 13 October 2021).
29. spaCy Industrial-strength Natural Language Processing in Python. Available online: https://spacy.io/ (accessed on 13 October 2021).
30. Taylor, A.; Marcus, M.; Santorini, B. The Penn Treebank: An Overview. In *Treebanks: Building and Using Parsed Corpora*; Abeillé, A., Ed.; Text, Speech and Language Technology; Springer: Dordrecht, The Netherlands, 2003; pp. 5–22, ISBN 978-94-010-0201-1.
31. Souag, A.; Salinesi, C.; Comyn-Wattiau, I. Ontologies for Security Requirements: A Literature Survey and Classification. In *Proceedings of the International Conference on Advanced Information Systems Engineering*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 61–69.
32. Li, T.; Ni, Y. Paving Ontological Foundation for Social Engineering Analysis. In *Proceedings of the Advanced Information Systems Engineering*; Giorgini, P., Weber, B., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 246–260.
33. Protégé. Available online: https://protege.stanford.edu/ (accessed on 11 May 2021).
34. Murata, M.; Laurent, S.S.; Kohn, D. XML Media Types. Available online: https://www.rfc-editor.org/rfc/rfc3023.html (accessed on 16 April 2021).
35. GATE.Ac.Uk—Index.Html. Available online: https://gate.ac.uk/ (accessed on 11 March 2021).
36. Prodigy An Annotation Tool for AI, Machine Learning & NLP. Available online: https://prodi.gy (accessed on 23 September 2021).
37. WordNet | A Lexical Database for English. Available online: https://wordnet.princeton.edu/ (accessed on 14 October 2021).
38. LIWC—WP Engine. Available online: https://liwc.wpengine.com/ (accessed on 15 November 2021).
39. Ramshaw, L.A.; Marcus, M.P. Text chunking using transformation-based learning. In *Natural Language Processing Using Very Large Corpora*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 157–176.
40. Craig, R.T. Generalization of Scott's index of intercoder agreement. *Public Opin. Q.* **1981**, *45*, 260–264. [CrossRef]
41. McHugh, M.L. Interrater reliability: The kappa statistic. *Biochem. Med.* **2012**, *22*, 276–282. [CrossRef]
42. Shelar, H.; Kaur, G.; Heda, N.; Agrawal, P. Named Entity Recognition Approaches and Their Comparison for Custom NER Model. *Sci. Technol. Libr.* **2020**, *39*, 324–337. [CrossRef]
43. Keras: The Python deep learning API. Available online: https://keras.io/ (accessed on 9 October 2021).

44. TensorFlow. Available online: https://www.tensorflow.org/ (accessed on 9 October 2021).
45. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural architectures for named entity recognition. *arXiv* **2016**, arXiv:1603.01360.
46. Lu, J.; Behbood, V.; Hao, P.; Zuo, H.; Xue, S.; Zhang, G. Transfer learning using computational intelligence: A survey. *Knowl.-Based Syst.* **2015**, *80*, 14–23. [CrossRef]
47. Tsinganos, N.; Sakellariou, G.; Fouliras, P.; Mavridis, I. Towards an Automated Recognition System for Chat-based Social Engineering Attacks in Enterprise Environments. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*; ACM: Hamburg, Germany, 2018; pp. 1–10.
48. Iyer, R.R.; Sycara, K.; Li, Y. Detecting Type of Persuasion: Is there Structure in Persuasion Tactics? In Proceedings of the CMNA@ICAIL, London, UK, 16 July 2017.
49. Edwards, M.J.; Peersman, C.; Rashid, A. Scamming the Scammers: Towards Automatic Detection of Persuasion in Advance Fee Frauds. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; pp. 1291–1299. [CrossRef]
50. Hidey, C.; McKeown, K. Persuasive influence detection: The role of argument sequencing. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
51. Stajner, S.; Yenikent, S. A Survey of Automatic Personality Detection from Texts. In Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain, 13–18 September 2020; pp. 6284–6295.
52. Kang, S.; Ko, Y.; Seo, J. Hierarchical speech-act classification for discourse analysis. *Pattern Recognit. Lett.* **2013**, *34*, 1119–1124. [CrossRef]