Université de Montréal

Titre du mémoire

Self-disclosure model for classifying & predicting text-based online disclosure

Par

Ramyasree Vedantham

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures et postdoctorales en vue de l'obtention du grade de Maître ès sciences (M.Sc.)

en Informatique, option Intelligence Artificielle

Juin 2021

Université de Montréal

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Ce mémoire intitulé

Self-disclosure model for classifying &

predicting text-based online disclosure

Présenté par

Ramyasree Vedantham

A été évalué par un jury composé des personnes suivantes

Claude Frasson
Président-rapporteur

Esma Aïmeur
Directrice de recherche

Jian-Yun Nie
Membre du jury

Résumé

Les médias sociaux et les sites de réseaux sociaux sont devenus des babillards numériques pour les internautes à cause de leur évolution accélérée. Comme ces sites encouragent les consommateurs à exposer des informations personnelles via des profils et des publications, l'utilisation accrue des médias sociaux a généré des problèmes d'invasion de la vie privée. Des chercheurs ont fait de nombreux efforts pour détecter l'auto-divulgation en utilisant des techniques d'extraction d'informations. Des recherches récentes sur l'apprentissage automatique et les méthodes de traitement du langage naturel montrent que la compréhension du sens contextuel des mots peut entraîner une meilleure précision que les méthodes d'extraction de données traditionnelles.

Comme mentionné précédemment, les utilisateurs ignorent souvent la quantité d'informations personnelles publiées dans les forums en ligne. Il est donc nécessaire de détecter les diverses divulgations en langage naturel et de leur donner le choix de tester la possibilité de divulgation avant de publier.

Pour ce faire, ce travail propose le « SD_ELECTRA », un modèle de langage spécifique au contexte. Ce type de modèle détecte les divulgations d'intérêts, de données personnelles, d'éducation et de travail, de relations, de personnalité, de résidence, de voyage et d'accueil dans les données des médias sociaux. L'objectif est de créer un modèle linguistique spécifique au contexte sur une plate-forme de médias sociaux qui fonctionne mieux que les modèles linguistiques généraux.

De plus, les récents progrès des modèles de transformateurs ont ouvert la voie à la formation de modèles de langage à partir de zéro et à des scores plus élevés. Les résultats expérimentaux montrent que SD_ELECTRA a surpassé le modèle de base dans toutes les métriques considérées pour la méthode de classification de texte standard. En outre, les résultats montrent également que l'entraînement d'un modèle de langage avec un corpus spécifique au contexte de préentraînement plus petit sur un seul GPU peut améliorer les performances.

Une application Web illustrative est conçue pour permettre aux utilisateurs de tester les possibilités de divulgation dans leurs publications sur les réseaux sociaux. En conséquence, en utilisant l'efficacité du modèle suggéré, les utilisateurs pourraient obtenir un apprentissage en temps réel sur l'auto-divulgation.

Mots-clés : Auto-divulgation, Traitement du langage naturel, Extraction d'informations, Apprentissage automatique, Réseaux de neurones, Vie privée sur les réseaux soc

Abstract

Social media and social networking sites have evolved into digital billboards for internet users due to their rapid expansion. As these sites encourage consumers to expose personal information via profiles and postings, increased use of social media has generated privacy concerns. There have been notable efforts from researchers to detect self-disclosure using Information extraction (IE) techniques. Recent research on machine learning and natural language processing methods shows that understanding the contextual meaning of the words can result in better accuracy than traditional data extraction methods.

Driven by the facts mentioned earlier, users are often ignorant of the quantity of personal information published in online forums, there is a need to detect various disclosures in natural language and give them a choice to test the possibility of disclosure before posting.

For this purpose, this work proposes "SD_ELECTRA," a context-specific language model to detect Interest, Personal, Education and Work, Relationship, Personality, Residence, Travel plan, and Hospitality disclosures in social media data. The goal is to create a context-specific language model on a social media platform that performs better than the general language models.

Moreover, recent advancements in transformer models paved the way to train language models from scratch and achieve higher scores. Experimental results show that SD_ELECTRA has outperformed the base model in all considered metrics for the standard text classification method. In addition, the results also show that training a language model with a smaller pre-training context-specific corpus on a single GPU can improve its performance.

An illustrative web application designed allows users to test the disclosure possibilities in their social media posts. As a result, by utilizing the efficiency of the suggested model, users would be able to get real-time learning on self-disclosure.

Keywords: Self-disclosure, Natural Language Processing, Information extraction, Transformers, Privacy on social media, User interface.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

API	Application Programming Interface
ARPANET	Advanced Research Projects Agency Network
ATS	Automatic Text Summarization
BERT	Bidirectional Encoder Representations from Transformers
BI-LSTM	Bidirectional Long Short-Term Memory
BOW	Bag-of-words
CNN	Convolution Neural Network
CPU	Central Processing Unit
CR	Coreference Resolution
DRER	Disclosed Related Entity Recognizer
ELECTRA	Efficiently Learning an Encoder that classifies Token Replacement Accurately
Glove	Global Vectors for Word Representation
GLUE	The General Language Understanding Evaluation
GPU	Graphics Processing Unit
IC	Information content
IE	Information Extraction
IP	Internet Protocol
IR	Information Retrieval
LIWC	Linguistic Inquiry and Word Count
LSTM	Long Short-Term Memory
MLM	Masked Language Modeling
NEL	Named Entity linking
NER	Named entity recognition

NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NP	Noun Phrase
OCR	Optical character recognition
POS	Part-of-speech
RAM	Random Access Memory
RE	Relation Extraction
RNN	Recurrent Neural Network
RTD	Replaced Token Detection
SD_ELECTRA	Self-disclosure ELECTRA
SDTM	Self-disclosure topic model SDTM
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
WWW	Word Wide Web
XL-NET	Transformer-XL

To my parents, Sravya, Sajjad, for their endless love, support, and encouragement

and brothers Harish and Swamy, who have been a source of inspiration....

Acknowledgments

I want to thank my incredible supervisor, Professor Esma Aïmeur, for her unlimited moral support and continuous guidance. I feel very fortunate to be supervised by such a caring, helpful and inspirational mentor. Without your guidance, constant support, and belief in me, it would have been impossible for me to accomplish this thesis on detecting Self-disclosure.

I would like to thank the jury members, Professor Claude Frasson and Professor Jian-Yun Nie, for sparing their precious time to review and evaluate my thesis.

I would particularly like to thank my fellow lab mate Rim for teaching me several tips while writing my research work and surpass my limits. I want to thank Yishu, my brother and classmate, for being there for me whenever I needed.

Finally, I would like to thank my parents, siblings, and love for all the support they gave me to achieve my goals.

Chapter 1

Introduction

In this chapter, we discuss the evolution of social media, the potential threats to privacy on disclosing personal information on social media, the self-disclosure, and the motivation behind this thesis' work. We also identify the research objectives, the contribution, and organization of our thesis dissertation. In Chapter 2, we discuss the background and related work.

1.1 Evolution of social media

Over the decades, interacting and communicating with people at long distances has always been a challenge. The inventions such as the telephone and computers paved the way for a new form of communication. In the 1950s, computer science was an area of research interest for many years [1]. During the 1960s, the researchers found that communication between different users can be achieved using wide area networks [2]. In the early 1970s, Kleinrock (2010) published about the prototype of Transmission Control Protocol (TCP) and Internet Protocol (IP) [2] and in 1980s, supercomputing networks are developed [3]. During 1989-1990, Switzerland's Computer scientist [4] launches the first concept of using the *World Wide Web* (WWW). The World Wide Web paves the way to link hypertext documents into an information system that can be accessed by any nodes connected to the same network [5]. Since 1990, the internet has started a revolution in the ways people can communicate. It introduced electronic mails, telephonic conversations over the internet, video chats, and text messages. The internet has taken over the world since, and by 2007, 97% of the world's communication happened to be on the internet [6]. The internet continued to grow, paving the way for e-commerce, entertainment, and other social networking services.

Social Networking sites or *Social Media* are online platforms where people socialize, communicate, share and exchange personal information such as career plans, likes, dislikes, favorites, and interests. The discovery of smartphones in 1992 has enabled people to use social networking sites and social media applications in abundance [7]. People use social media applications to communicate using desktops, tablets, and smartphones. The concept

of social networking existed among humans from ancient years where people in the same town or village used to gather and socialize [8]. This age-old habit of people has made social media platforms grow, as people started using them to connect with others of different time zones and locations across the world.

Web-based sites such as LinkedIn and Myspace have been in a boom during the early 2000s, whereas entertainment sites such as YouTube came in 2005 bought an immense change in the way entertainment is viewed [9]. According to Pew Research Center (2014), by 2006, Facebook and Twitter expanded and became more popular. According to Iqbal (2021), the social networking application Facebook has a massive 2.80 billion active monthly users by 2021 [10]. When joining these Social Media platforms, users create public profiles to share personal information with the networking site providers [11]. As for Facebook, once a public profile is created, users can share a lot of personal data (photos, contact details, and others), location and can express their personal views on their "walls," leading to the discussion on user's information privacy. On another social networking site Airbnb, users can share contact information, location details, preferences, travel plans, and much more personal data publicly visible.

1.2 Potential threats to privacy on social media

With the significant expansion of social networking services, large amounts of personal information are stored in the site provider's database as well as cloud database. Many causes contribute to the invasion of privacy over social networking sites. The business of social media runs on the content shared between users publicly [12]. Though this is not considered harmful, people should be aware of each application’s privacy concerns and privacy settings. The primary issue found is that there are no laws for protecting the information shared over social media, and hence any breach of privacy cannot be penalized.

In 2010, Mark Zuckerberg, founder of Facebook, bought new privacy changes to Facebook settings [13] and it brought back a lot of discussions on data privacy at that time. Facebook has been caught on sharing enormous information with third-party companies where users' privacy is kept at stake. This incident brought a transformation in the system where the social networking sites' ethics have been questioned.

It is a well-known reality that the data cannot be hidden or deleted once posted on the internet. User awareness about privacy has rapidly increased from 2010, where much research is done on the data privacy area. By 2013, 60% of the teenage users of Facebook have made their profiles private, which means that they only share details with their friends and family members, avoiding any stranger viewing their profiles [14]. Discussions are still ongoing regarding privacy concerns as social networking sites such as LinkedIn allow employers to review users' profiles before offering them jobs at their organization. The ability to obtain the right balance on information privacy is still a question as the media dynamics are changing.

Information Privacy is mainly related to the users' personal information stored in computer systems [15]. The relationship between collecting data and protecting them according to the user's preferences is also called data privacy or data protection [16]. There is a need to protect the personal information as it can be linked to medical records, financial transactions, and business-related information. Information privacy is restricted to the users and the privacy of organizations and institutions and on how much data they can share or communicate with others [17]. Information privacy can be applied to data in many ways where the information is masked, encrypted, decrypted, or authenticated. There are different types of information privacy, such as online privacy, financial privacy, and medical privacy, explained in Technopedia (2021). In online privacy, the service provider sites would give privacy policies stating the intent behind collecting online and offline data. The data is more sensitive during financial privacy as there is much fraud revolving around the financial details, and the privacy policies are set accordingly. Medical privacy has exceptionally stringent laws as the medical records of users are confidential. The authentications and strong regulations are defined in the health organizations to access their medical information. Though social networking sites and organizations define the information privacy laws, privacy concerns still exist.

Privacy concerns have begun to gain attention during the early 2000s where users are concerned about how much of their information can be protected [18]. There is growing concern over the years because the individual perceptions of storage of data have changed. Such concerns relate to how the service provider misuses the data and whether any third-party companies use them without authorization from the user [19].

The Information privacy concern is not limited to the individual perceptions. However, it has been moved further to the management level where the stakeholders, business leaders, scholars, activists, and government bodies are involved. A poll conducted in America on the information privacy part revealed that 72% of people were concerned about the online behaviors on their profiles [20]. The study conducted by Smith, Dinev, & Xu (2011) has shown that large companies like Google, Facebook and Amazon have collected and shared the data with the stakeholder companies for different purposes.

The Privacy concerns over social media spiked over recent years. The data breach Incidents such as what happened with Yahoo in 2014 and other similar incidents have alarmed the users' data security [21]. Yahoo has been facing lawsuits for two significant data breaches considered as one of its kind in history. Thielman (2016) has published that about 500 million users were affected because of the breach. There was also a breach of users' email addresses, telephone numbers, security questions and answers, date of birth, and other encrypted data. And with this information, the hackers can easily hack the victims' linked online accounts [22]. The computer security teams worldwide have warned the users that the incident has a more significant impact than seen. It can have a massive impact on the banking and other financial accounts linked to the same email addresses. McMillan & Knutson (2017) also discovered that the information was sold to affiliated companies of yahoo during the data breach even though the users' profiles have been deleted.

According to the Cyber Security teams, the Facebook- Cambridge Analytica data scandal in 2016, discovered later in 2018, is one of the major security breach incidents recorded worldwide [23]. About 267 million users' data were leaked and collected by a British Consulting firm named Cambridge Analytica [24]. Confessore (2018) published that the users' data were used to provide analytical assistance to the US presidential campaigns in 2016. After the scandal, significant awareness was raised among the people to protect their personal information. Later in 2017, the Facebook application found bugs, and about 50 million users' profiles have been exposed to the data breach [25].

In January 2021, the popular text messaging application WhatsApp has announced a new change in the privacy policies and mandates users to accept the terms and conditions of the policies [26]. This announcement has bought quite an outrage among the people regarding WhatsApp's information since it was acquired by Facebook. The new policy in the European

region explicitly states that users' profiles are shared among Facebook and WhatsApp [27]. Though the CEO of WhatsApp rubbished the claims of sharing users' personal information, such as location and encrypted text messages, there is still a debate on the users' major privacy concerns [28]. Other encrypted data such as IP address, mobile networks, language, and time zone can be shared in the future. And there are critics that there is an intent of earning money behind the policy. Though Facebook has been in the limelight for breaching the security policies, it is still a discussion topic on the latest policy of WhatsApp. This incident brought a wave of awareness among the users. There has been a significant hike in the users' profiles in other secured applications such as Signal and Telegram [29].

In the past decade, many criminal activities have been recorded where users' sensitive information were stolen, and their accounts were hacked without consent. In addition, social media threats such as data mining, phishing, identity theft, malware, botnet attacks, stalking, employment scams, and sexual predators have been increasing over the years [30]. In 2009 researchers at Carnegie Mellon University showed that it is possible to predict each person's social security numbers using the personal information available online and, in the database, [31].

Personal information is the information used to identify a living individual. It must be either about the user's personality or identity. The user's personal information on social media sites consists of birth dates, identity, social security numbers, and other details. Though the provider sites have security policies to secure this information, there is still a significant threat to the data in privacy attacks [32].

In 2016, Aljohani et al. conducted an online survey and found that disclosing personal information depends on both the individual who uses it and the collectors [33]. The survey covered various factors, such as the purpose and the intention behind collecting the information and the control over the data shared by users [33]. The study shows that about 74% of the people are aware of the privacy attacks yet still share personal information if they benefit from the sharing. About 54% of the people never mind what people are reading about them on social media. Aljohani et al. (2016) reveal in their survey that the same type of information could be considered sensitive and non-sensitive based on whether the user voluntarily revealed the information or it taken without his knowledge.

Thus, personal information is considered sensitive based on the context of self-disclosure instead of the whole information itself [34].

1.3 Self-disclosure and motivation

The concept of *self-disclosure* is defined as personal information a person discloses to other individuals and organizations [35]. Self-disclosure plays a vital role in maintaining relationships, and recent research found a strong link between disclosure and liking [36]. The most recent study on self-disclosure notices a significant impact on people's mental health sharing and communicating online [37]. The impact can be defined in both positive and negative ways. For example, when people started expressing their views and opinions on social media, they tend to feel confident and expressive in their lives [38]. On the other hand, the disclosed information on the internet has its drawbacks.

Most of the early studies focus on using social media and sharing information between family, friends, and colleagues. Simultaneously, recent studies on people observe the extent of self-disclosure on social networking sites [39]. In Acquisti & Gross (2006) study on Facebook, results show that 75% of the users revealed their full names in their profiles, 24% have given their postal addresses, and 84% revealed their date of births on their walls. As years passed, people became more aware, and only 10% of people are willing to reveal their postal address in public [40]. Thus, from the studies, it is understood that disclosing information depends on the difference of opinions of individuals. Multiple aspects of location, age, career, gender, and others are involved during self-disclosure [41].

Furthermore, past research discloses that specified costs and benefits are significantly involved during self-disclosure. In the context of online disclosure, the benefits are maintaining social relationships, building connections, and developing a social network for oneself, entertainment, and business-related capital [42]. As mentioned in the previous sections, the costs related to privacy have adverse effects on the individual in both the physical and the digital world [43].

According to the research in 2014, *privacy* in online disclosure can be categorized as informational, social, and psychological privacy [44]. Informational privacy describes to what extent a user has control over personal information. Social privacy explains regulating the

proximity and distance with others. At the same time, Dienlin & Trepte (2014) stated that psychological privacy regulates the intimacy of the information and emotional inputs and outputs. In recent years, studies confirm that though individuals are aware of privacy concerns, they make merely the slightest changes to their social media profiles. This type of privacy-compromising behavior is called privacy-paradox [45].

The privacy paradox can be specified as a measure of individual privacy behavior that does not match with the concern about his privacy. However, in the analysis, Norberg & Horne (2007) have also pointed out that human decision-making behavior is not always rational; heuristics and external biases also influence it.

However, users underrate the dark side of online disclosure. It is still observed by studies and surveys that people are always open to platforms where they find freedom of expression by being anonymous without involving in face-to-face conversations [46]. According to a Harvard study in 2013, 80% of the information posted online by individuals is about themselves [14]

The reason for such an over-whelming self-disclosure ratio notices is that users want to maintain social capital, seek feedback about their works, and communicate with others [47]. Moreover, some theories address the trust users have in networking sites, and the time they devote to these networks [48]. At the same time, other researchers expose that personality traits and age also affect self-disclosure [49]. The authors also explain evidence of social influence, peer influence catalyzing to disclose more information. It is evident that though users and organizations are aware of the privacy concerns, other factors outweigh these concerns. The need to be a part of online communities and share about themselves has taken over the privacy fears [50].

There are immense risks involved while disclosing as there is a loss of control of information once it is shared online. Growing privacy threats are identity theft, harassment, security threat, Cybercrimes, Phishing attacks, information theft, abuse, and other shady practices. For example, geo-tagging information such as location and time on Facebook can potentially be an unwanted security threat. This example illustrates that disclosing personal information online can cause danger if the data is misused. Hence, privacy concern is the most significant underlying risk involved in self-disclosure.

Though there are debates about the negative relationship between privacy concerns and self-disclosure, self-disclosure still plays an essential role in social media platforms and is often characterized as the 'privacy setting' of one's user profile [51]. A practical model of privacy protection is an urgent need in the age of social networks, which result in activities such as online publishing, chatting, text messaging, blogging, and playing online games, among others. As a result, a lack of adequate disclosure control can be the source of numerous privacy issues, and the adverse effects of disclosing information might be enormous. In other words, the detection of self-disclosure to reduce the privacy risk behind online disclosure is necessary. Addressing the need, it is critical to build algorithms and tools that assist users in detecting privacy-related self-disclosure in a social media text. In this thesis, we identify a potential approach to detecting various disclosures and propose an end-user solution that identifies disclosure when users enter data.

1.4 Research objectives and contribution

The primary purpose of this thesis is to detect self-disclosure from a text and identify the different categories of self-disclosure. *Natural language processing* has efficiently understood human-generated languages and extracted information from digital texts by using several NLP tools. By identifying the self-disclosure, we aim to warn the users about the privacy risk behind revealing their personal information. This kind of work would provide awareness for users to protect their privacy while enjoying social media. In this thesis, our objectives are:

- Proposing a method to detect self-disclosure from a user-generated text on social media, using natural language processing.
- Using current natural language processing techniques, designing an effective text classification model SD_ELECTRA (Self-disclosure ELECTRA).
- Identifying the disclosure category, including: Interest disclosure, Personal disclosure, Education and Work disclosure, Relationship disclosure, Personality disclosure, Residence and Travel plan disclosure, and Hospitality disclosure.
- Experimenting and evaluating the designed methodology on different metrics and improve performance compared with the base models.
- Analyzing the potential risks of disclosing information online and discussing risks associated with each disclosure in detail.

- Designing an illustrative user interface platform where the SD_ELECTRA is combined with a front-end web application to test the method's feasibility on real-time social media posts.
- Evaluating user interface on correctness to identify disclosures and enlighten users about the possible disclosures in their social media posts.

This work consists of pre-training a language model from scratch, SD_ELECTRA, by adopting standard techniques from ELECTRA-small published by Clark et al. in 2020 [52].

We propose two different models SD_ELECTRA_V1 and SD_ELECTRA_V2, trained on two different GPUs using two different tokenizers. In addition, experiments were performed on the Airbnb host profiles dataset specially labeled for multiple disclosure categories [53]. Our contribution in this work also includes designing an illustrative web application using an open-source framework [54], enabling the user to examine their real-time social media posts and get the possibility of disclosures predicted by SD_ELECTRA.

Furthermore, our contribution enhances the privacy perspective of social media data. Finally, it throws light on achieving higher performance by context-specific methodology using significantly less computational resources than bigger language models.

1.5 Thesis organization

The thesis is organized into five chapters, including the first chapter, which is the introduction. Chapter 2 provides insights on the recent research work on NLP techniques for information extraction and self-disclosure, along with the limitations of each approach. Chapter 3 presents details about the proposed approach, including the architecture and algorithm used to categorize disclosure. Chapter 4 includes experimental setup and results achieved on different datasets. Finally, we conclude the thesis in Chapter 5 and discuss the prospects of this research work.

Chapter 2

Related Work

In this chapter, we discuss natural language processing (NLP), Information extraction (IE), identifying self-disclosure using NLP, and the recent research work done in the areas related to this thesis.

2.1 Natural Language Processing

Natural languages are the languages spoken by humans. Natural language processing is a field of artificial intelligence and linguistics that involves understanding these languages by a computer and generating natural languages [55]. In the 1950s, the field of NLP began as the intersection of artificial intelligence and linguistics [56]. The authors Chopra, Prashar, & Sain (2013) did a detailed study on the evolution of natural language processing in their article. According to the authors, by the 1980s, concepts like semantics, rule-based parsing, morphology, and other areas of natural language understanding started to grow swiftly. Thus, the field of NLP has grown significantly in the previous ten years. This research has benefited from the rise of social media and the internet. The ready availability of digital data and the hardware capabilities such as speed and memory of the computers also increased the efficiency of NLP algorithms [57].There are multiple levels of language approach explaining the Natural Language Processing system. The levels are phonology, morphology, Lexical, Syntactic, Semantic, Discourse, and Pragmatic Analysis as demonstrated in [55] [56].

Morphology deals with the componential nature of the words, which are called morphemes [58]. For example, a word like ‘unsuccessful’ has three morphemes where the prefix is ‘un,’ the root is ‘success,’ and the suffix is ‘full. At this level, the NLP system understands the word by recognizing the meaning of each morpheme.

In *Phonology*, the system analyses sound waves and interpret the word from the digital signal. It uses several phonological rules to interpret speech sound across words [59].

At the level of *Lexical analysis*, the NLP system breaks the texts into paragraphs, sentences, and words and interprets the meaning of each word which is called word-level understanding [60].

At the *Syntactic analysis level*, the grammatical structure of the sentences and words are depicted [61]. At this level, their meaning is conveyed by syntax by analyzing the order and the dependency sentence. For example, the sentences 'The cat chased the rat' and 'The rat chased the cat' have different meanings that differ only in syntax.

At the *Semantic level*, the possible meaning of the sentences is determined by focusing on word-level meanings of the sentence [62]. For example, the word "bank" can be used as a noun to define a riverside, or the other meaning connects it to money. In this case, we require to disambiguate words at a semantic level instead of a lexical level.

At the *Discourse level*, the NLP system identifies the sentence's meaning by the preceding or the following one, contributing to the meaning of the multi-level sentence texts [63]. For example, when pronouns are used such as "she liked the food", the previous discourse context has an impact on the meaning of the sentence.

The *Pragmatic level* is mainly concerned about language usage in different situations and understanding the context based on the utilization and situation [64]. For example, the word "complete the task" should be interpreted as a request instead of an order. The above-discussed levels in the hierarchy are shown in Fig. 1.

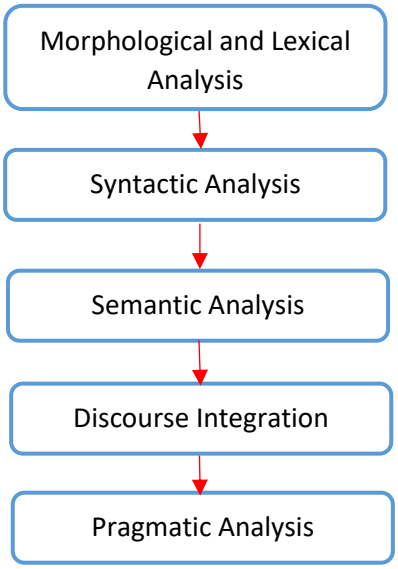

Fig. 1. Levels of Natural Language Processing [65].

Thus, based on the requirement of the research, different levels can be implemented by NLP systems. In our thesis, we are implementing low-level language processing such as morphemes, words, sentences, and high-level language processing, which understands the contextual meaning of the sentences.

NLP algorithms mainly use statistical machine learning techniques [66]. As stated by the Khan et al. previously, *rule-based algorithms* were used in language processing, such as decision trees in which no learning occurs [66]. By introducing machine learning in NLP tasks, the algorithms automatically learn rules by analyzing similar datasets or real-world applications.

Automatic learning methods have been using statistical and probabilistic methods that produce robust models which can also analyze unknown datasets [67]. The combination of deep learning and natural language processing has increased the efficiency of NLP systems in performing various tasks.

The Natural language tasks include Machine translation, Discourse analysis, Named entity recognition (NER), Natural language understanding, Morphological segmentation, Optical character recognition (OCR), Information Retrieval (IR), Automatic Text Summarization (ATS), part of speech tagging, Information extraction (IE), Question-Answering system, sentiment analysis, and Parsing [68].

This thesis concerns the Information extraction task while detecting the personal information disclosure in the texts and using levels of NLP predicting the categories of self-disclosure.

2.2 Information Extraction using NLP

Information extraction (IE) refers to using computational methods to detect relevant information in a document and convert it into computer-based representation for processing, retrieval, and storage purposes [69]. Detecting sensitive information in raw text or user-generated real-time data has been a challenging problem. However, there are various efficient techniques and methods in detecting sensitive information in emails, social security numbers, medical data, and other domain-specific data [70].

In 2012, Sanchez et al. presented a general-purpose model to detect the sensitive information in the document in a domain-independent way [71]. Information content (IC) is measured using Information theory concepts, and terms with higher IC are considered as sensitive. The

Information Content (IC) of a term is a measure of how much information it provides when it appears in a context. The inverse of the probability of encountering the term t in a corpus (p(t)) is used to calculate the IC of t. Equation (1) explains the formula to calculate IC.

$$IC(t) = -log_2 p(t) \qquad (1)$$

The terms t are extracted using noun phrases (NPs) which reveal too much information in document d. These NPs are detected using NLP tools such as sentence detection, tokenization, part-of-speech tagging, and syntactic parsing. As a final step, Sanchez et al. proposed that sensitive terms are those which has IC greater than or equal to the detection threshold β. Equation (2) computes the value of sensitive Noun phrases in document d.

$$NP_{sensitive} = NP \in d \mid IC(NP) >= \beta \qquad (2)$$

Though the research successfully identified the sensitive information in the noun phrases independent of the domain, the method has some limitations. In this method, the words considered potentially sensitive are removed based on the information content, and hence the document utility is lost, and words are left out in a meaningless order.

The other techniques of information privacy disclosures are rule-based approaches. The study conducted in 2008 by Tang et al. reveals that in this method, systematic rules are designed to extract information from text instead of dictionaries [72]. The two main rule-based approaches are the bottom-up approach, which learns rules from special situations to general ones, and the top-down method, which learns rules from general cases to special ones, which are the two basic rule learning algorithms used.

Some of the rules discussed in the bottom-up approach are tagging rules, contextual rules, and correction rules.

Tagging rule

A tagging rule consists of two parts: a left side that contains a pattern of conditions on a connected series of words, and a right side that is an operation that inserts a tag into the texts. In the given example of Table 1. the tag <speaker> is inserted in the sentence.

Table 1. Example of initial tagging rule [72].

Pattern					Action
Word	**POS**	**Kind**	**Lookup**	**Name Entity**	**Tag**
Rita	NNP	Word	Person's Name	Person	<speaker>
is	VBZ	Word			
assistant	NN	Word	Job title		
professor	NN	Word			
.	.	Punctuation			

Contextual rule

Contextual rules are the selected rules where the non-best rules are applied to tag the sentence. For example, consider the rule that places a <speaker> tag between a capitalized and a lowercase word. Because of its low precision on the corpus, this rule does not belong in the best rule pool, however, it is dependable when used solely when closing a tag <speaker>. As a result, it will only be used if the best rules have previously identified an open tag <speaker> but not the associated closing tag </speaker>.

Correction rule

Correction rules are used to correct the wrong tags provided by tagging rules. Table 2. shows an example of a correction rule on one such tagging mistake where "pm" should be part of time expression.

Table 2. Example of correction rule [72].

	Pattern	Action
Word	**Wrong tag**	**Correct tag**
at		
4	</stime>	
pm		</stime>

In this approach, the sensitive information is detected first using general rules, and then the information is removed from the documents. Fig. 2 shows a sample flow of the rule based NLP model . This method fails when the patient's personal information is necessary to give him proper treatment and the information is removed, leaving the patient unidentified. This

method also fails to detect the entities related to the removed sensitive information such as phone numbers and names in the medical records.

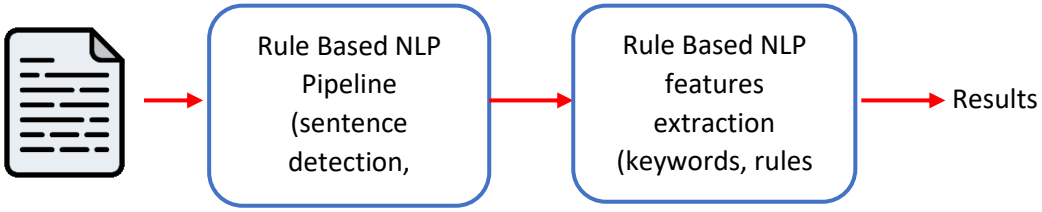

Fig. 2. Rule-based NLP Model [72].

By Using the information extraction technologies and NLP, new research sub-fields of information extraction tasks such as *Named entity recognition* (NER), *Named Entity Linking* (NEL), *Coreference Resolution* (CR), *Relation Extraction* (RE), *Knowledge based Construction* (KBC), and reasoning are emerged. As per Tang, Hong, Zhang, & Li (2008), the low-level NLP tasks such as Part-of-Speech tagging, parsing, NER are building blocks of complex NLP tasks. Hence the low-level tasks such as tokenizer, stemmer, parser, part-of-speech tagger are very efficient nowadays. It is also important to look at how the different information extraction tools would affect the performance of the NLP tasks.

2.2.1 NLP methods for feature extraction

The data collected from social media platforms are generally unstructured. Many NLP tools are required to make meaningful inferences from the text and extract the features which are sent as input for machine learning and deep learning classifiers. The NLP tools applicable in our thesis, are discussed below.

Tokenization

There is a sense of ambiguity in the natural language, and sometimes it is tough to define whether a group of characters form a word or whether they are part of multiple words [73]. For example, words like “t-shirt” should be considered a single word instead of separating “t” and “shirt” whereas some words such as “shouldn’t” can be considered as either a single word or two words. This type of ambiguities has no particular answer in NLP, and word tokenization in NLP can be one of approaches to solve the issue [74]. Sometimes words with symbols, emoticons are important, which is why tokenization is used to break the words into strings.

NLTK libraries (Natural Language Toolkit) usually implement the tokenizer and the words are delimited by matching the suffixes, slangs, emoticons, and abbreviations. At first, the prefix is removed, and the words are matched with the corpus. If no match is found, the suffix is removed, and the matching word is searched [75]. At the end of the tokenization process, a list of tokens is provided, and these tokens are sent to other processes for extracting the meaning from the words.

Stemming

The goal of *stemming* and *Lemmatizing* is to dimensionally reduce the words to their root. The difference between both processes is that the suffix of the words are removed, which might not be the actual word in the dictionary form [76]. *Porter stemmer* is the algorithm commonly used in NLP downloaded from NLTK library which take the token of words and removes the suffix and generated the corresponding stemmed root words [77].

Word Vectorization

In the data extraction process, the text in a document is normally represented in the form of a matrix where the document rows consist of every single text instance and the column represents the tokens. This type of representation is called feature vectors [78].The values can either be Boolean values or determined from a method called *term frequency-inverse document frequency* (TF-IDF). These word vector representations are used in the thesis, which acts as the inputs to the language models.

2.2.2 Information Extraction task- Named Entity Recognition

Named Entity Recognition (NER) is a word-level tagging process where each word in a sentence is mapped to a named entity. The features from data are fed into the classifiers and the output is in the form of labels of person, organization, location, etc [79]. In NER, three kinds of information are extracted; entity name, time, and quantitative expression [80]. The implementation of this algorithm can be used in text mining, predicting future decisions, extract the content to track the data circulation, and other fields.

Si, Zhou, & Gai (2020) have extracted data from unstructured Chinese text and applied a rule-based approach, including extracting feature words, values, and units present in the data. The authors have divided the data extraction process into parts. The first one is word

segmentation, and the second one is rule matching algorithm where a feature word list is created, consisting of trending words, fuzzy words, conjunctions and ending words. It is observed that the authors were successful in extracting data with an accuracy of more than 90%. However, though the rule-based approach has been efficient, it has limitations such as inability to automatically add new rules and new feature words as the learning process does not occur.

Recently, researchers have started using semi-supervised learning models for NER tasks. One such methods is bootstrapping, where a small amount of supervision is needed to provide a seed set for the learning process [81]. In 2015, Siencnik has proposed to use Word2Vec features in the NER tasks [82]. It is also observed that implementation of bi-directional Long Short-Term Memory (BI-LSTM) and LSTM based models for NER tasks has outperformed other methods with an F1 score of 90.94 [83]. Many researchers used the NER tools in detecting self-disclosure in the different types of datasets, which is discussed in the later sections of this chapter.

2.2.3 Information extraction task- Sentiment Analysis

Sentiment Analysis is one of the widely used methods in NLP. This method is most useful when users express their opinions and thoughts in social media comments, surveys, blogs, and other reviews. Usually, sentiment analysis is used to classify whether the comment or post is negative, positive, or neutral. A polarity score is calculated by evaluating all the factors contributing to the sentence's meaning [84].

The Sentiment Analysis is performed by using machine learning techniques such as supervised and unsupervised models [85]. The supervised models used by sentiment analysis include Naïve Bayes, Random Forest, and gradient boosting models. In the unsupervised methods of sentiment analysis, lexicon-based methods are used where the polarities of the words are applied to calculate the sentiment score.

A group of researchers in 2005 proposed the method to use machine learning methods such as Support Vector Machine (SVM) to extract the sentiment expressions from the text by classifying them into a sentimental tag and by attaching a sentimental weight to each sentence [84]. Based on the sentimental tags, the genre of each sentence is calculated, and classifiers are used to predict the corresponding sentiment label of the given text. Though the

authors achieved good performance in extracting sentiment information, the growth in machine learning models has paved the way to perform sentiment analysis on bigger datasets that have a larger amount of real-time data.

2.2.4 Information Extraction task- Relation Extraction

Relation Extraction is divided into two steps: the first step is to detect the relation utterance in the texts. The second one is classifying the detected relation into different classes [86]. The RE is used in different applications such as knowledge base population, Question-Answering, and other applications.

According to Zhang, Chen, & Liu (2017), the text is analyzed by part of speech tagging and dependency parsing techniques to extract relations. The relation extraction task can be performed by different methods such as:

- Hand-built patterns
- Bootstrapping methods
- Supervised methods
- Distant supervision
- Unsupervised methods

The *hand-built patterns* are one of the earlier methods for relation extraction [87]. These patterns are hard to write and hard to maintain, and they are domain dependant as it is required to build patterns by hand for each relation.

Later the relation extraction systems started using *bootstrapping methods* where different seed sets need to be constructed for each data set, and every relation should have seeds [88]. The biggest problem with the bootstrapping method is that the algorithms are sensitive to original seed sets, and there are a lot of parameters involved in tuning the models. Therefore, to get higher precision and overcome the challenges of handwritten rules and bootstrapping, supervised machine learning techniques were used [89].

The *supervised machine learning methods* have been efficient in giving higher accuracy, and the sample flow of the supervised machine learning model is given in Fig. 3. But it has significant limitations since a very large amount of human annotated data is required. The

cost of labeling more relations has been expensive. Hence, to overcome the problem of labeled datasets, a distance supervised method is introduced [90].

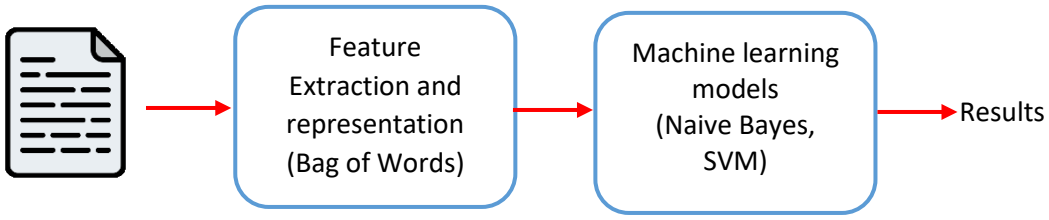

Fig. 3. Supervised machine learning Model [89].

In the *distant supervision* methods knowledge base, such as Wikipedia, is used as a text corpus for automatically training and tagging labels [91]. This method used a knowledge base to find whether the entities are related, and it assumes that if entities are present in a single sentence, there is a relation between them. Unfortunately, this method leads to the false-positive patterns.

Supervised learning models are time-consuming, and labeling input and output variables always require expertise. To overcome these challenges, *unsupervised deep learning methods* have been introduced toward the information extraction process. In supervised learning, classifying large amounts of data might be difficult, unsupervised learning, on the other hand, can handle massive amounts of data in real-time. The example of an unsupervised deep learning model is shown in Fig. 4.

In the recent trends of relation extraction, some of the deep learning techniques such as Convolution Neural Networks (CNN) [92], Recurrent Neural Networks (RNN) [93], long short-term memory (LSTM) [94], and LSTM-RNN are popularly used. In recent years, deep learning models have started achieving higher efficiency than the conventional relation extraction models.

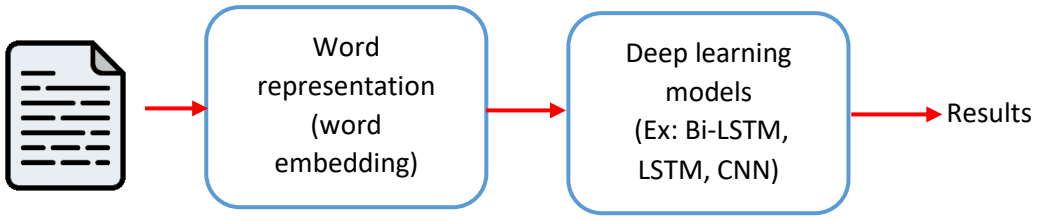

Fig. 4. Unsupervised deep learning Model [92].

BERT

Recently, a study on the representation of words using contextual word representations has paved the way for new deep neural language models *BERT* [95] and *XLNET* [96]. Bidirectional Encoder Representations from Transformer (BERT) is a transformer-based machine learning technique is used for NLP. BERT is pre-trained and developed by Google employees Devlin, Chang, Lee, & Toutanova in 2018.

As stated by the authors, BERT is an attention mechanism that learns from contextual relations between words and sub words. The Transformer is considered bidirectional because it reads the sequence of words at once. The architecture of BERT model is presented in Fig. 5. This characteristic of BERT allows the model to learn the context of a particular word from the surrounding words.

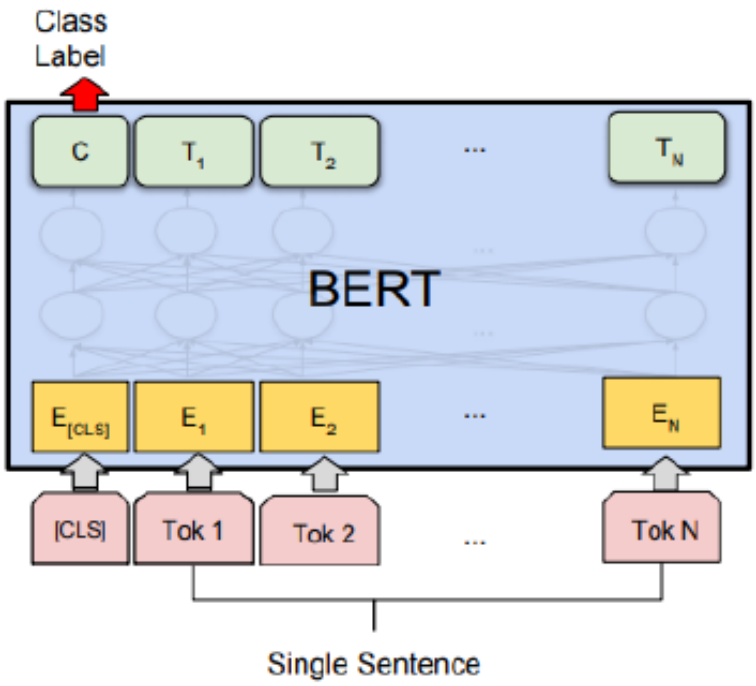

Fig. 5. BERT Model Architecture [95].

The BERT model can be used during sentiment Analysis, Question Answering tasks, and Named Entity Recognition (NER) and Relation extraction [97]. In Named Entity Recognition, the BERT model is required to mark entities such as a person, name, date, phone number, etc. In Relation extraction, BERT model is used as a feature extractor and loads pre-trained parameters for fine-tuning [98]. The BERT model integrates external semantic knowledge with entity relation knowledge to improve the efficiency of the classifier.

Two factors that contributed to BERT's success are its transformer-encoder design (enabling bidirectional language understanding) and its pre-training process. Looking at the latter in-depth, including Masked Language Modelling (MLM), we can quickly see several areas to improve.

1. 15% of tokens are replaced by MLM with the [MASK] token, then BERT is trained to reproduce the original sequence using the remaining unmasked context. The token efficiency, or the amount of language understanding obtained per token during the pre-training phase, is substantially hampered by this approach as it only predicts the 15% masked tokens limiting the learning from each sentence. This inefficiency, however, cannot be solved by merely raising the mask-token ratio.
2. BERT requires a lot of computational resources to train, fine-tune, and make inferences. BERT also relies on huge training data where about 2.8 billion words of Wikipedia data with 800 million words of book corpus data are used during pre-training BERT.

To overcome these limitations, a completely new method of pre-training would be required. In this thesis addressing all the limitations listed above, we propose a new model, SD_ELECTRA, which uses a different pre-training approach and trains on a single GPU while using 45x less compute. The approach is discussed in detail in chapter 3.

The following sections discuss different approaches used by other researchers to identify self-disclosure using NLP techniques.

2.3 Self-disclosure detection using NER

A study from a group of researchers from Pennsylvania State University in 2019 has proposed detecting self-disclosure in comments on news articles from four major English News websites [99]. The researchers considered data set from ABC News, CNBC, The Huffing Post, and TechCrunch when the news was published from March to August 2015. A total of about 60000 user comments from 22132 users are collected and categorized.

For the process of categorization, the researchers Umar, Squicciarini, & Rajtmajer (2019) used rule-based matching. Self-disclosure in the text was signaled by a specific category-related verb and relevant named entities. For Example, in the sentence "I live in Pennsylvania" , the

matched verb " live" with location entity "Pennsylvania" attributes for location disclosure. Hence the approach identifies self-disclosure categories in the sentence with the knowledge of verb and named entities. However, sometimes these method gives false positives. A proximity window is used for either side of the verb, which detects which category the sentence belongs to. To give an example, in the sentence "I have countless arguments with seemingly educated people in many countries on why Singapore works", the matched verb "have" with location entity "Singapore" attributes for location disclosure. However, the proximity check prohibits it from being classified as a location category. The researchers considered objective categories such as Birthday, Age, Race, Sexual Orientation, Location, Affiliation, Money, Relationships, Experience, whereas Subjective categories as Interests, Feelings, Opinions.

To detect Self-disclosure, the authors used opinion extraction techniques to extract opinions [100]. They also used techniques such as first-person pronouns as prominent features detected as linguistic markers, which are used to detect the disclosures in online contents [101]. The overall process of detection, according to the authors, takes place in four phases 1) construction of dictionary 2) subject-verb-object triplet detection 3) Named Entity Recognition (NER), and 4) rule-based matching.

In the first phase of the developing of a dictionary, a vocabulary is constructed using the verbs that disclose different self-disclosure categories using as a reference the Airbnb labeled dataset [53]. In the second phase of subject-verb-object triplet detection, the authors detect a basic pattern referring to the subject, verb and object extraction. Finally, in the third phase of NER, the authors used the OneNote corpus to detect different entities.

According to the predictions from linguistic indicators, the study revealed that anonymous users tend to disclose more personal information than regular users. In this study, the authors used unsupervised categorization methods to identify self-disclosure language patterns and classify them into relevant categories. They also discussed the percentage of people who use anonymity to disclose information, and this information can be used to identify them and disclose their identity.

The automatic detection process used in the paper is limited to category-specific dictionaries, and the taxonomy used is based on the reference to Airbnb dictionaries. In contrast, there

other categories can be considered. Also, a more intelligent method can be used to detect the disclosure in sentences by considering the sentences' contextual meaning.

2.4 Self-disclosure detection using Supervised Machine Learning models

The self-disclosure is not limited to human interactions but also during human-machine interactions, such as real-world users and Amazon Alexa where significant, patterns of self-disclosure are widely identified.

A research study in 2018 has determined that the conversations with dialog systems have specific self-disclosure patterns [102]. The information such as personal beliefs, thoughts, likes, dislikes, aspirations, and preferences is considered disclosure. In the experiments conducted by Ravichander et al. the amount of self-disclosure information is proportional to the amount of information added to the conversation [102]. The real-world data is collected from the users who interacted with Amazon Alexa [103], and then the data is inputted into a machine learning model. The collected information results in a combination of Bag-of-words features, Linguistic style Features, and LIWC features.

The *Bag-of-words* (BOW) model is a tool for feature generation where a text is represented as a bag of words, irrespective of the grammar and word order. The term frequency-inverse document frequency (TF-IDF) is a measure to identify whether a word is important in a document or not [104]. The Linguistic style Features refer to a person's speech pattern. For example, it includes part of speech tagging, word choices, pauses, jokes, negation, and other elements. The style of the utterances can also indicate self-disclosure in the texts [105].

The *Linguistic Inquiry and Word Count* (LIWC) features for self-disclosure are taken as words with strong emotions and family relations [106]. These features are combined and passed as input to a support vector machine (SVM) with a linear kernel. The classifier has achieved a precision of about 72.7% over 134 test data [103] in identifying the self-disclosure available in the user collected utterances. Though the work successfully identified the self-disclosure in the chats with the dialog systems, the research limitation is that self-disclosure is considered binary in the study. Hence, the depth of the disclosure is not considered along with classifying the levels of self-disclosure.

2.5 Self-disclosure detection using Semi-supervised Machine Learning models

Group of authors Bak et al. researched a vast dataset of Twitter conversations proposed a semi-supervised machine learning method for categorizing self-disclosure [107]. In this approach self-disclosure was classified into three levels; G (general) for no disclosure, M for medium disclosure, and H for high disclosure [108]. The data set is collected from Twitter from English tweeting users, which consisted of nearly 2 million conversations posted between the years 2007-2013 [107].

A semi-supervised machine learning model Self-disclosure topic model (SDTM) is created for classifying the levels of disclosure. SDTM assumes that self-disclosure behavior may be predicted by combining simple linguistic features (for example, pronouns) with found semantic themes (i.e., topics). For example, the phrase "I am finally over this awful relationship" employs the first-person pronoun and discusses the topic of personal relationships. It is found that about 64.5% of accuracy is achieved using SDTM as compared to Bayesian model [109], supervised topic model [110], LIWC [111], Seed words and trigrams [107], a joint model for sentiment and topic in seed words [112] and First-person pronouns [113].

In the research by it is found that there is a relation between initial conversation frequency and self-disclosure. It is also observed that self-disclosure leads to more frequent conversations between individuals. The research paper has succeeded in proving the social psychology results that self-disclosure is proportional to the high initial conversation frequency on large-scale datasets. However, though the research has achieved good accuracy, it can still be improved using better models. Also, self-disclosure features only include nouns and topics, whereas the patterns, lexical semantics, and non-parametric topic models can be included for better accuracy.

2.6 Self-disclosure detection using combining NLP techniques and machine learning approaches

A research study published in 2019 has used Natural Language Processing techniques and machine learning approaches to reveal depression-related disclosures in Reddit posts [114]. In this paper, Reddit users' dataset consists of different features such as depressed and non-

depressed [115]. The depression-related posts are posts by users who seek support from the online community Reddit. Example of words that indicate depression in the posts are “alone, break, depressed, unhappy, wrong, loneliness”, etc. The approach applied to detecting depression is NLP techniques along with text classification.

In the feature extraction stage, to explore the linguistic usage in the posts, they employed N-gram features [116], Latent Dirichlet Allocation topics Model [117], Linguistic Inquiry, and Word Count Dictionary (LIWC) [118]. For the word frequency, they used unigrams and bigrams by using the TF-IDF method. And in the Text classification techniques, classifiers such as Support Vector Machine, Random Forest classifier, Logistic Regression, Adaptive Boosting, and Multilayer Perceptron (MLP) are used to estimate depression.

In the feature analysis and predictive power of *N-gram* features, the top 100 unigrams and top 100 bigrams of each post are studied. The results show that feelings, negative emotions, suicidal thoughts, anger, hostility, negation words, and hopelessness are depression indicative words. In the LIWC technique by Pennebaker, Booth, Boyd, & Francis(2015), all the words, both depressed and non-depressed, are converted into psychological, linguistic features resulting in the correlation presented in the feature extraction. It is observed that the highest correlation is present in psychological processes, linguistic dimensions, and personal concerns.

In the LDA topic model proposed by Resnik, Garron, & Resnik (2013), many hidden topics are extracted from the posts. The topic model generates multiple topics, such as job, health, depression, friends, money, and others, which have depressed, and heartbroken words written in the posts. The machine learning classification results found that a model combination of LIWC+LDA+bigram+MLP neural network gives the higher accuracy outweighing SVM and other classifiers.

The results show that higher predictive performance is hidden in proper feature selections and feature combinations. The authors successfully identified depression-related words and disclosures in the posts using NLP and machine learning techniques. The online user content can be confused, and sometimes, the context which mentions depression might not be depressed. For example, "I did a study about depression and unhappiness" can be quoted as a depressed post using the NLP techniques. But in general, it is a post showing awareness

about depression. Though the methodologies are reasonably good, the method is limited to depressed and non-depressed posts.

2.7 Self-disclosure detection using NLP and Deep learning methods

Though the machine learning methods have successfully identified the patterns, the natural language processing models and deep learning methods have been significantly efficient over text data.

A study done by researchers at Boise State University in 2019 has successfully designed a model to help the users to gain control of the personal information shared by detecting the sensitive information in the sentence level [97].The authors have used natural language processing tools such *as Disclosed Related Entity Recognizer* (DRER), which is developed from Named Entity Recognizer (NER) [97]. These tools are used to derive linguistic features like part of speech, syntactic dependencies, and entity relations from the document. A multichannel *convolution neural network* (CNN) is trained with the derived features, and the classifier detects whether the feature is disclosure or non-disclosure. The authors successfully achieved high accuracy using neural networks and notifying users about the piece of text where disclosure occurs. The approach can be improved by making the model more intelligent by understanding the lexical meaning of the sentence and testing the approach with unstructured data sets.

A study in 2019 has experimented on using various methods of identifying bullying related tweets through NLP and Machine learning [119]. A labeled dataset from the University of Wisconsin is used, which has labels like "Bullying Traces," "Types," "Form," "Teasing," "Author Role," and "Emotions." The label focused on this part of the research is "Bullying Traces." The data set is processed using various NLP features such as TF-IDF, stop words, and word-vectors. Different machine learning algorithms such as multinomial Naïve Bayes, Recurrent Neural Network, and Convolutional Neural Network are used to identify the tweets that indicate bullying.

The TF-IDF method considers each word's frequency in the tweets and takes the inverse frequency. By doing so, the stop words such as “the” and “a” weighted less heavily than words, which help in better differentiating the bullying-related tweets. The output from TF-

IDF is then fed to the multinomial Naïve Bayes Classifier. Word embedding maps each word to a "Word2Vec". The significant property of Word2Vec is they not only take syntactic properties they also consider semantic relationships between words. This data from word embedding is given as input to the RNN model containing an LSTM layer and a fully connected layer. A CNN model 2-D convolutional kernels and 2-D max-pooling kernel, a flattening layer, and a fully connected layer are used. It was observed that CNN and Multinomial Naïve Bayes were giving higher precision scores than the RNN. Using deep learning and NLP together to identify bullying-related disclosures in a tweet is helpful in cyberbullying. But the drawbacks of this paper are considering a data set without emoticons and dataset built with proper grammar. The model couldn't tell if a post was bullying or not since it needed to know the context in which it was written. Following section discusses BERT which is bi-directional, and this characteristic of the model allows it to learn the context of a word based on all of its surroundings.

2.8 Self-disclosure detection using NLP and BERT

Research done by Pennsylvania State University in 2020 has proposed using an ensemble method of both BERT and CNN (Convolution Neural Network) to detect self-disclosure [120]. Akiti, Rajtmajer, & Squicciarini (2020) developed a multi-modal approach for classifying self-disclosure. The self-disclosure their paper is classified as Emotional disclosure, Information disclosure, Support, General Support, Information Support, and Emotional support. The authors have proposed two different models in their study where the BERT model is used to fine-tune the word representations and classification using sentence representations. The second model uses a CNN as a classifier whose embedding layer is replaced by a pre-trained BERT model. The BERT model achieved a mean F1-score of 0.525 with mean precision of 0.45 and mean recall of 0.655, whereas the CNN model provides a mean F1-score of 0.485 with a precision of 0.417 and recall of 0.592. It is observed from the results that the BERT model predicts and classifies the text better as compared to the BERT embedded CNN model. Though the authors have successfully implemented the BERT model for privacy detection in disclosures, the fine-tuning and data cleansing process would have achieved higher prediction scores.

The same research group has tested models such as *BiLSTM* (Bidirectional Long short-term memory) and BERT on user-generated Reddit data [121]. An in-depth semantic analysis is

performed using the combination of word embedding and the BERT model. Self-disclosure detection performance is increased in sentences, and meaningful semantic information about the disclosures is furnished. The authors used Semantic Role Labeling to identify the lexical units and their semantic roles, which detects self-disclosure. Semantic Role Labeling assigns semantic roles consistent with the frame semantics predefined in FrameNet [122]. They approach the problem of detecting disclosure by learning semantic-role based labels that are common to disclosure. The idea behind Semantic Role Labeling is to assign semantic roles that are consistent with FrameNet's predefined frame semantics. For Example, the semantic role labeled sentence "I am worried" contains emotion disclosure. The predicate "worried" is assigned a frame Emotion_Active and the lexical unit "I" is assigned a semantic role of Experiencer. Thus, Emotion_Active with an Experiencer as "I" leads to emotional self-disclosure. The dataset consists of emotional disclosure and information disclosure data collected from user-generated Reddit content. The data is tested for emotional disclosure classification on BERT, BiLSTM, and an ensemble method of Global Vectors for Word Representation (Glove) embedding with BERT. The results are observed that the ensemble method of Glove+BERT has the highest F1 score of 0.64 and Recall of 0.69. This approach predicts information about disclosure behavior levels, whether it is high, medium, or low, in addition to the improved performance. This approach also shows some studies on the peer influence factor in the disclosure. This research study seems to be promising and has covered details about the in-depth semantic analysis.

In 2020, Dadu et al. [123] published an ensemble BERT-based models for detecting social media disclosure in text. The authors have ensembled RoBERTa [124] and ALBERT [125] in their model and fine-tuned the ensembled model to achieving an F1 score gain of 3% compared with ROBERTa and ALBERT individually. However, these models are harder to fine-tune and ship as a single model. These models also have a larger prediction time since predictions of possibly hundreds of models are needed. A new model that outperforms BERT in computational power, using smaller datasets and less training time, is discussed in the next section.

2.9 Transformers for Information Extraction – ELECTRA

It is known that from the above-discussed sections that transformers have been dominating Information Extraction in Natural language processing since 2017. The BERT model [95] has improved to different models such as *ALBERT* [125], *RoBERTa* [124], *DistilBERT* [126], and *XLNET* [96], relying on larger datasets and having high-cost computation time. In March 2020, a new approach to training the language model with significantly less computation power had been introduced by Clark et al. from Stanford University and Google [52].

Masked language models (MLMs), such as BERT, RoBERTa, and ALBERT, predict the identities of a small number of masked words from the input. Instead of predicting every input token, MLM models predict a tiny portion, the 15% that was masked out, limiting the amount learned from each sentence. Unlike BERT, which uses *Masked Language Modeling* (MLM) during pre-training, ELECTRA (Efficiently Learning an Encoder that classifies Token Replacement Accurately) uses *replaced token detection* (RTD) in pre-training. Rather than masking a random selection of input tokens, this method employs a second neural network that seeks to deceive the model by substituting fake tokens for random tokens.

The technique is similar to that used by GANs. We pit two networks (the generator and discriminator) against each other when training GANs. The generator is programmed to 'trick' the discriminator by providing increasingly fake data. The discriminator is then left to determine whether the data generated by the generator model is true and which is false. A similar strategy is used by ELECTRA. The discriminator in this approach is in charge of determining which tokens are true and which are fake.

The architecture of ELECTRA is not the same as GAN-type architecture because the generator is not optimized to increase the discriminator's loss. Instead, it is trained as a conventional MLM model, guessing the [MASK] values. According to the authors Clark et al. this pre-training method is more efficient than MLM because the model must consider every token in every sample it encounters. In contrast, MLM only requires the model to concentrate on [MASK] tokens. [52].

After the pre-training is completed, the generator can be discarded, and the remaining ELECTRA model can be used for downstream tasks such as Classification and Question

Answering. The Clark et al. used the new ELECTRA model on GLUE performance and predicted that the ELECTRA model is on par with the BERT-small model, which uses 12 times lesser computation than BERT-small [52]. The authors also showed significant improvements in comparing ELECTRA with Roberta and XLNET models, where ELECTRA uses 4times less computation time.

When ELECTRA is compared to various state-of-the-art NLP models, it is discovered that given the same compute budget, it outperforms earlier techniques, performing similarly to RoBERTa and XLNet while utilizing less computation power. The Fig. 6. shows the amount of computational power utilized to train the model (measured in FLOPs) is shown on the x-axis, while the dev GLUE (General Language Understanding Evaluation) score is shown on the y-axis. ELECTRA learns substantially faster than previous NLP models that have been pre-trained.

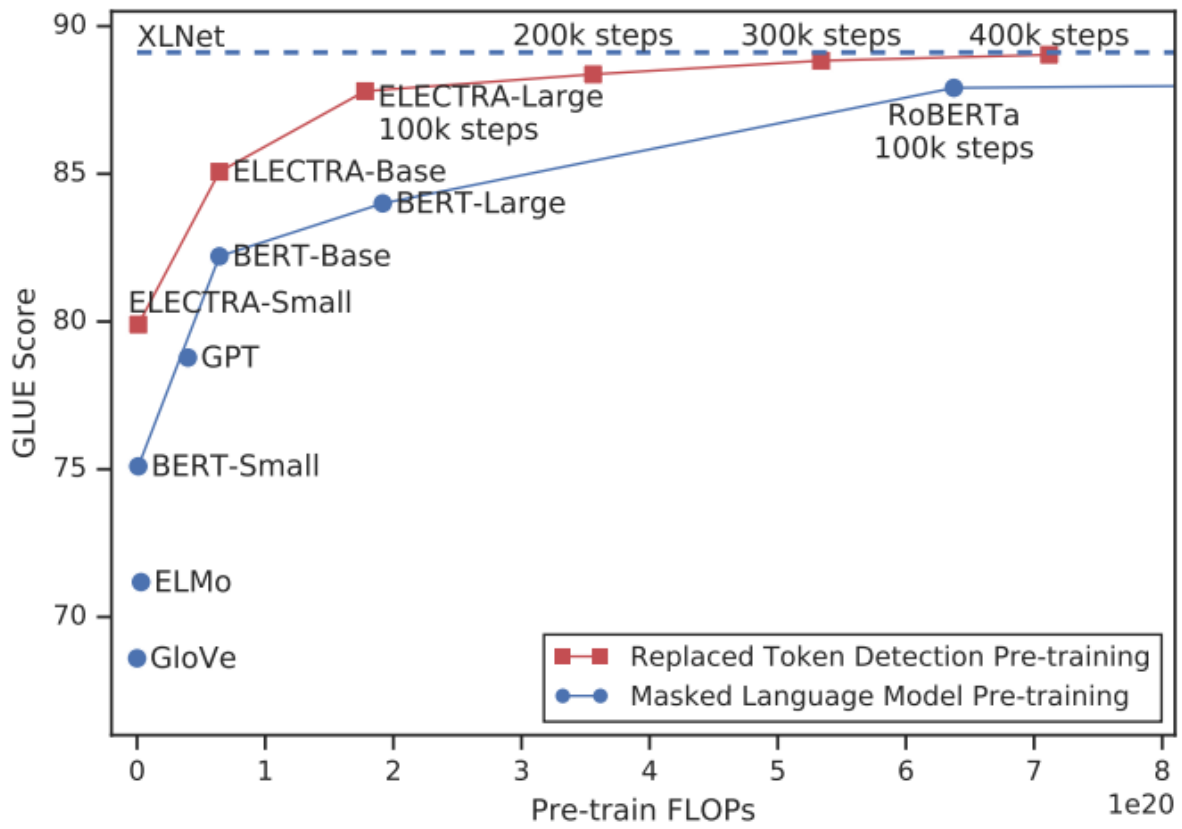

Fig. 6. Comparison of ELECTRA for computational power utilized to train the model with other models on dev GLUE score [52].

In the original paper of ELECTRA, the authors Clark et al. further compared BERT to ELECTRA for various model sizes. They found that as the models get smaller, the gains from ELECTRA get larger. The tiny models are fully trained to convergence, as shown in Fig. 7. demonstrating that fully trained ELECTRA outperforms BERT in downstream score.

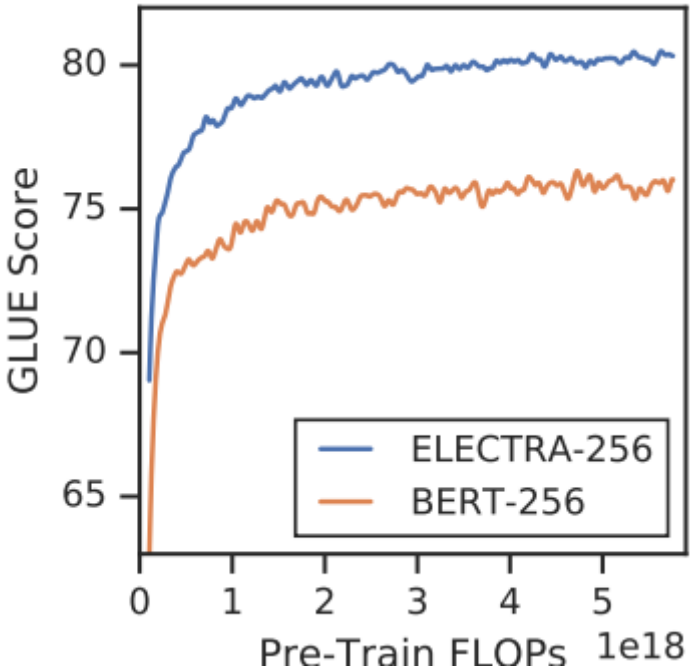

Fig. 7. ELECTRA vs BERT on GLUE score [52].

From the above discussion, it is understood that to be effective, most existing pre-training approaches require a lot of computing power, which raises questions regarding cost and accessibility. Inspired by the research studies by Clark et al. we propose a compute efficient and an effective downstream performance model to detect self-disclosure in this thesis. We have considered ELECTRA-small proposed by authors as the base model for our thesis, as observed from the Fig. 7. ELECTRA-Small outperforms a comparably small BERT model by 5 points on GLUE score. The solution proposed in this research work consists of building a language model SD_ELECTRA from scratch with pre-training and fine-tuning the SD_ELECTRA model to identify the self-disclosure in user-generated text data with greater accuracy and classify them based on different types of disclosures as explained in Chapter 3.

Chapter 3

SD_ELECTRA: Domain specific language model ELECTRA built to detect Self-disclosure in social media

As discussed in the paper “ELECTRA: Pre-training Text Encoders as Discriminators rather Than Generators” [52], the state of art method outperforms other language models with excellent efficiency even at a small scale. This method made it to build task-specified language models on a single GPU that can be trained in a few days.

The proposed work targets privacy-related issues in social media posts and reviews, thus detecting self-disclosure in social media with a Context-specific language model SD_ELECTRA. This model is pre-trained on the Airbnb dataset from scratch, fine-tuned, and other disclosures in the Airbnb host profiles data set are predicted.

This chapter discusses the architecture of the proposed method, the pre-training and fine-tuning process of language models, data preparation and pre-processing methods, Feature Extraction, and implementations of the algorithms proposed.

3.1 Model Architecture

This section introduces the Architecture of the proposed model, which consists of six steps, as shown in Fig. 8. In the pre-training step, a large dataset of around 6GB is collected from a non-commercial website, “Inside Airbnb,” which hosts Airbnb data for analysis and research every month. This large corpus is then cleaned using preprocessing methods and then used as pre-training data for the ELECTRA model. The Existing google published ELECTRA model is taken as a reference to construct SD_ELECTRA.

In the fine-tuning step, the pre-trained model is fine-tuned using the task-specific Airbnb host profile data specially designed for identifying the self-disclosures present in the profile introduction and reviews by the hosts. This data is split into 80% training dataset and 20% test dataset. The training dataset is used for fine-tuning the model, and the test data set is passed to the final fine-tuned model to predict the classes of disclosure.

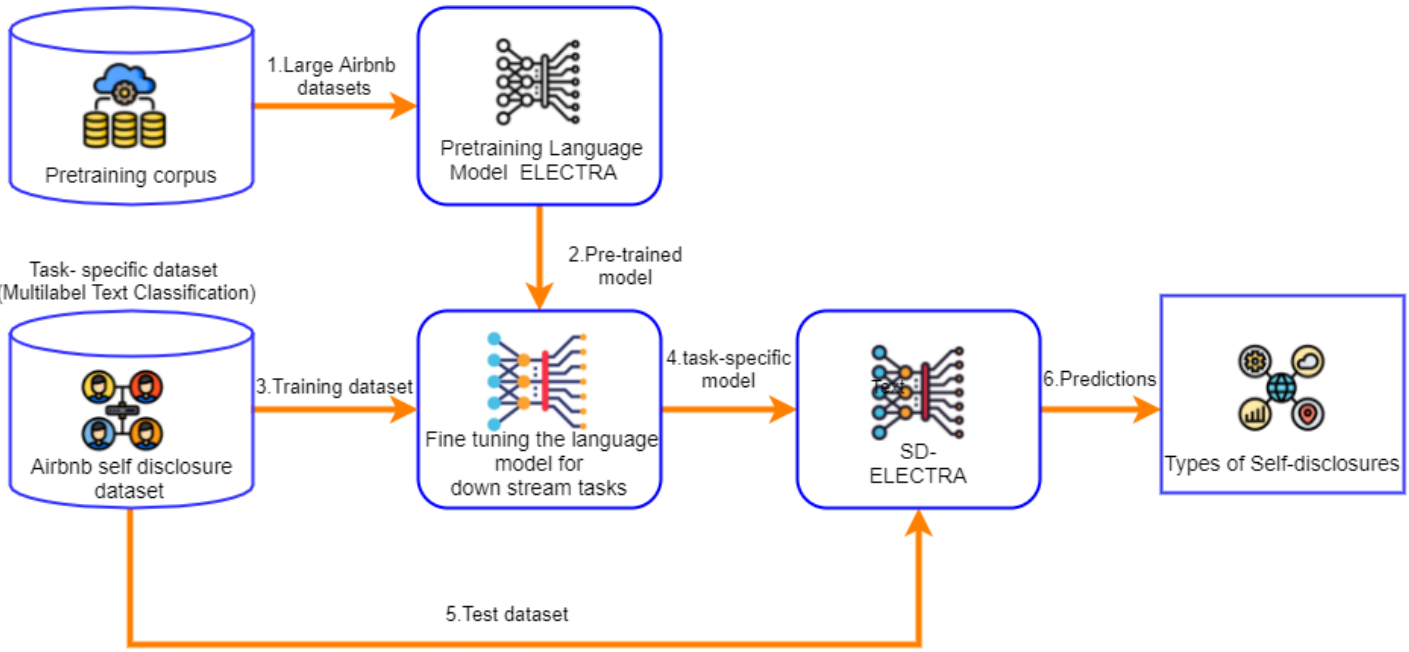

Fig. 8. Proposed Architecture for Detecting Self-disclosure in social media data.

The following are the steps of architecture;

- Collecting pre-training corpus
- Pre-train SD_ELECTRA
- Collecting fine-tuning training dataset
- Fine-tuning task specific SD_ELECTRA
- Evaluating SD_ELECTRA on test dataset
- Predicting the self-disclosure categories.

The next sections include a detailed discussion on the concepts of the proposed architecture.

3.2 Pre-Training Corpus

For the Construction of the pre-training corpus, a large corpus is collected from the "Inside Airbnb" website [127]. This website publishes Airbnb data independently for non-commercial purposes such as analysis and research on the data. These datasets consist of data from all

regions worldwide, and these datasets are updated every month based on the cities, states, and countries. For experiments, the data is collected from countries like the United States, Canada, Australia, Greece, Netherlands, and England in January 2021, February 2021, and March 2021.

These datasets are combined to form a significantly large unlabeled corpus consists of around 6 GB of data. The dataset is cleaned for emoticons, special characters, punctuations, stop word removal, and converted to the same case letters. This dataset does not consist of any labels; it is an unlabeled dataset prepared from scratch for the domain-specific model we proposed. The Airbnb data is chosen for the corpus to train SD_ELECTRA, and the training dataset used is also related to Airbnb. The proposed work hypothesizes that Domain-specific language models would predict results with more efficiency and less computational time. Since the disclosure labeled data used for fine-tuning on text classification task is the Airbnb host profile dataset [128], the pre-training corpora considered is a large amount of unlabeled Airbnb data.

Table 3. shows data sample of Montreal, Canada for April 2021 with detailed listings and detailed reviews files combined for all the major cities of countries.

Table 3. Sample data of Montreal Airbnb listings, reviews for the month of April 2021.

Montreal, Quebec, Canada

Data Compiled	Country/City	File Name	Description
17 April,2021	Montreal	listing.csv.gz	Detailed listings data for Montreal
17 April,2021	Montreal	calander.csv.gz	Detailed calendar data for listing in Montreal
17 April,2021	Montreal	review.csv.gz	Detailed Review data for listing in Montreal

3.3 Pre-Processing Methods

The pre-processing task is one of the most significant activities in natural language processing that should first be completed. It is a critical step since it will clean the dataset by minimizing its complexity and allowing the data to be prepared for the classification process. The dataset is first tokenized to break down the words into tokens, and then stemming is used to limit the tokens to a single type, usually a root word; for example, the term "images" is reduced to "image."

Although sometimes missed, one of the simplest and most effective text preparation forms is to lowercase all text data. As a result, stemming minimizes the number of superfluous words in a document. It is used to solve most text mining and NLP problems, and it is very beneficial when the dataset is not very huge. It also considerably improves predicted output consistency. The next step we did was to remove stop words. A set of frequently used terms in a language is known as stop words. Stop words in English include "a," "the," "is," "are," and other similar expressions. The idea behind stop word removal is that we may instead focus on the crucial words by deleting low-information terms from a document.

In the below sections, the above-mentioned preprocessing methods are discussed in detail, along with examples.

3.3.1 Tokenization

Tokenization is the process of separating sentences into different tokens, which are words, phrases, symbols, or other meaningful items.

These tokens are extremely important for pattern recognition and are used as a starting point for stemming and lemmatization. Tokenization is used to replace sensitive data pieces with non-sensitive ones. For experiments in this thesis, Auto tokenizer from Hugging face libraries is used. These Hugging face libraries are an AI community with a collection of already existing models, or newly created models can also be published here. The distil-Bert uncased tokenizer is used to create the vocabulary in this work. Fig. 9. shows a sample of Auto tokenizer and how they split words after using the tokenizer.

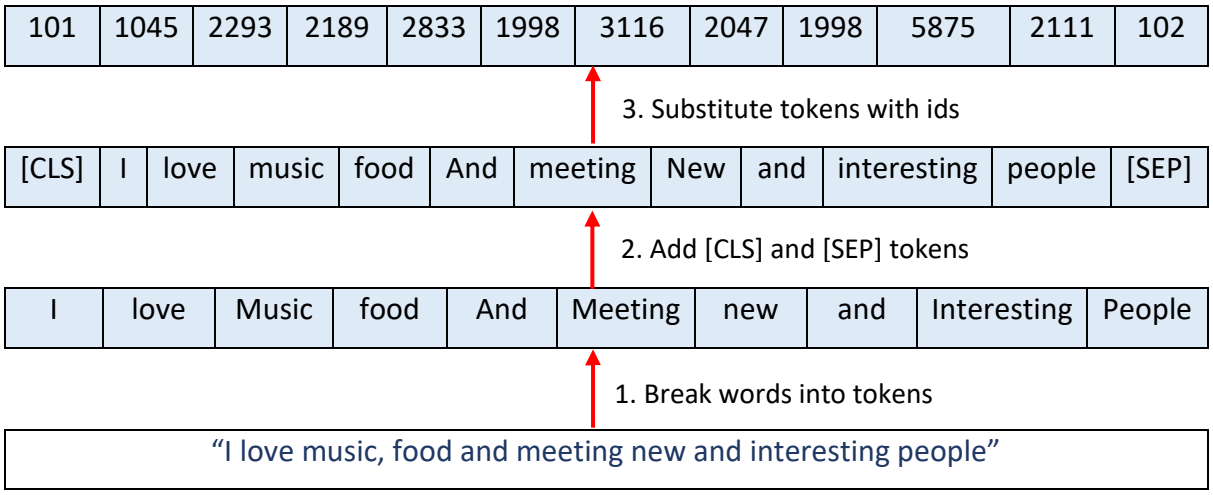

Fig. 9. Example of distil-BERT tokenizer attributes before normalization.

In the Fig. 8. shown, the first step is to divide the words of the sentence "I love music, food and meeting new and interesting people" into separate tokens using the tokenizer. Then, in the second step, the special tokens required for sentence categorization are then added (these are [CLS] at the first position and [SEP] at the end of the sentence). The [CLS] token stands for classification, and it refers to sentence-level classification. [CLS] is appended to BERT's beginning of each phrase (this is essentially like a start-of-sentence token).
[SEP]. BERT uses the tokens [SEP] to separate sentences, which is required for the Next Sentence Prediction task. In the third step, the tokenizer replaces each token with its id from the embedding table, which is a component we acquire with the trained model. These input ids are then given input to the language model since the models expect vectors as input format. Hence, this process of tokenization aid in the understanding of the context and the development of the NLP language model.

3.3.2 Creation of Vocabulary

The original word is broken down into smaller sub words and characters. This is because BERT *Vocabulary* has a fixed size of 30,522 tokens. The broken sub words and characters are used to represent words that are not in the lexicon.

Tokenizer examines the input sentence and decides whether to keep each word as a whole word, split it into sub words (with a particular representation of the first sub word and following sub words with a ## symbol – for example ##terest for the word Interest), or breakdown it into individual characters as the last resort. As a result, a word can be expressed as a collection of its component characters, at the very least. Fig. 10. few examples of the tokens contained in the vocabulary created by our algorithm are printed. These are tokens for the vocabulary from 5000 to 5015.

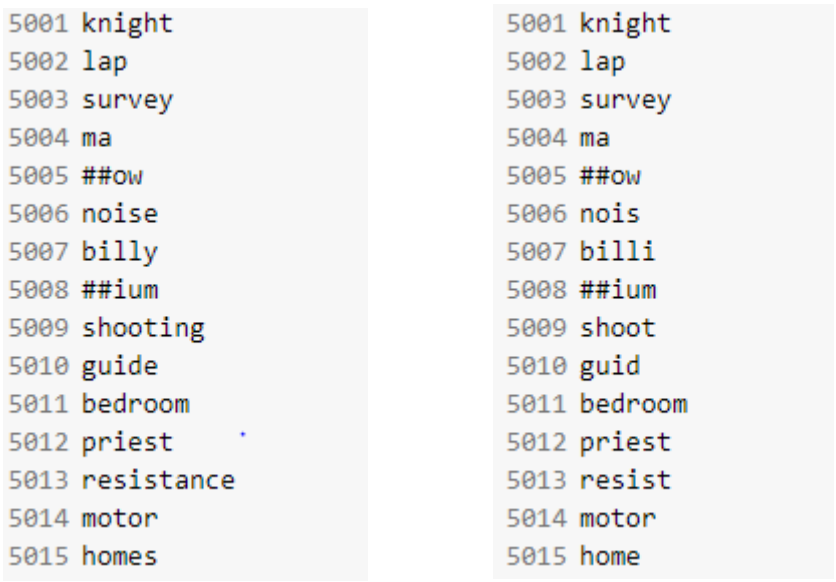

Fig. 10. Example of tokens contained in vocabulary [5000:5015] without stemming (left) and with stemming (right).

The created vocabulary is used to pre-train language models, and in this thesis, the vocabulary size considered is 30522.

3.4 Pre-Training in Language Models

Pretraining is a process where a transformer model learns to model a language. To put it another way, the Transformer will discover appropriate, context-dependent methods to represent text sequences. The model has already acquired the language features and needs to fine-tune its representations to accomplish a specific task. This obtained information may be utilized in downstream tasks, dramatically reducing the quantity of task-specific, labeled data necessary. Thus, the only data needed for pre-training is a large amount of (ideally) clean data.

Devlin et al. proposed the BERT model [95] that relies on masked language modelling (MLM) and next sentence prediction as pre-training tasks. The model is tasked with predicting whether two text sequences naturally follow each other or not in the next sentence prediction. This job improves downstream tasks like *Question Answering* and *Natural Language Inference* in the BERT study. Still, later it is proved to be unneeded in the RoBERTa [124]research, which solely employed masked language modelling. Regardless, the first technique, MLM, is what the ELECTRA [52] pre-training method attempts to improve.

3.4.1. Masked Language Modelling (MLM)

A specific percentage of the tokens are masked in Masked Language Modelling. The model is tasked with predicting the original token for the masked tokens, which might be a word or a part of a word) of an input sequence. Then the masked tokens can be replaced with an actual mask token (for example, [MASK]) or a random token from the lexicon (the set of all tokens known to the model).

On the other hand, MLM techniques only learn from the masked tokens (usually 15%) of each given example, according to the authors of the ELECTRA article [52]. As a result, the computational resources required to train a language model with MLM are extensive. Another disadvantage of MLM is that mask tokens emerge only during the pre-training step, never during fine-tuning or downstream usage. This disparity also contributes to a modest performance loss in MLM-trained models.

3.4.2 Pre-Training Strategy in ELECTRA

ELECTRA has a new pre-training strategy that seeks to meet or exceed the downstream performance of an MLM pre-trained model while utilizing much fewer computational resources. In ELECTRA, the pre-training task consists of detecting tokens that have been substituted in the input sequence. Two Transformer models, a generator, and a discriminator, are required for this arrangement.

Fig. 11. represents the original Two transformer models of the ELECTRA diagrammed by the authors of the paper [52] gives an overview of replaced token detection, a pre-training task.

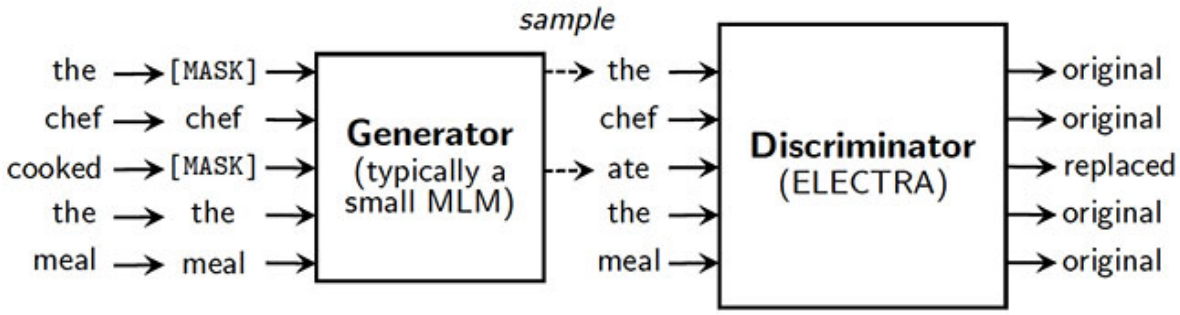

Fig. 11. Two transformer models used in the overview of replaced token detection,
A pretraining task of ELECTRA [52].

As per Clark, Luong, Le, & Manning (2020) the two neural networks Generator G and Discriminator D, are trained. Each one consists largely of an encoder such as Transformer that converts a sequence of input tokens x = [x_1,…, x_n] into a sequence of contextualized vector representations h(x) = [h_1,..., h_n]. For a specific position t, specific token x_t, token embeddings e, the generator's output for x_t with a softmax layer is given in Equation (3) [52].

$$P_G(x_t|x) = exp(e(x_t)^T h_G(x)_t) / \sum_{x'} exp(e(x')^T h_G(x)_t) \qquad (3)$$

For the same t, the predictions of the D with a sigmoid output layer is given in Equation (4) [52].

$$D(x,t) = sigmoid(w^T h_D(x)_t) \qquad (4)$$

The Generator G performs masked language modeling where it learns to predict the masked-out tokens' real identity. Whereas the Discriminator D is taught to discriminate tokens in the data substituted by generator samples. The loss functions of G and D can be represented as $L_G(x, \theta_G)$ and $L_D(x, \theta_D)$ respectively [52].

As mentioned in ELECTRA paper, the authors had significantly reduced the combined loss, where X is the large corpus as shown in equation (5) [52].

$$combinedloss = min_{\theta_G, \theta_D} \sum_{x \in X} L_G(x, \theta_G) + \lambda L_D(x, \theta_D) \qquad (5)$$

Algorithm 1 presents the process of replaced token detection in pre-training as referred to in ELECTRA paper is given below.

Algorithm 1 Replaced Token Detection in pre-training ELECTRA

Input: tokens $x = [x_1, ..., x_n]$, contextualized vector representations $h(x) = [h_1, ..., h_n]$

Output: $x^{corrupt}$ matching the original input x

Initialization:

let Generator be G with maximum likelihood and Discriminator D
For a given position t, x_t = [MASK]
G outputs $P_G(x_t|x)$, probability for generating a particular token xt
D predicts $D(x, t)$, whether x_t is real
set of positions to mask $m = [m_1, ..., m_k]$
replace positions with mask $x^{masked} = REPLACE(x, m, [MASK])$
create $x^{corrupt}$ by replacing masked tokens with generator samples
D predicts corresponding x for $x^{corrupt}$

The following are some pretraining steps followed in the ELECTRA method, and the generator is thrown out, whereas the discriminator is fine-tuned for downstream tasks.

Pretraining process steps:

1. Replace specific tokens in each given input sequence with a **[MASK]** token at random.
2. For all masked tokens, the generator predicts the original tokens.
3. The discriminator's input sequence is constructed by replacing **[MASK]** tokens with generator predictions.
4. The discriminator predicts whether each token in the sequence is an original or replaced by the generator.

The *discriminator* model is trained to identify which tokens have been replaced given a corrupted sequence. In contrast, the *generator* model is trained to predict the original tokens for masked-out tokens. As it conducts prediction on each token, the discriminator loss may be calculated over all input tokens. Only the masked tokens are used to calculate the model loss when using MLM is demonstrated to be a significant difference between ELECTRA and BERT's two models, and the fundamental cause for ELECTRA's superior efficiency.

This arrangement is comparable to a *GAN* (Generative Adversarial Network) training configuration; only the generator is not trained to trick the discriminator (so it is not adversarial per se). In addition, if the generator adequately predicts the original token of a masked token, the token is regarded as an original token (since it has not been damaged or modified).

After pre-training, the discriminator model is employed for downstream tasks, while the generator is discarded. One of the ELECTRA pre-training method's biggest advantages is that we can train our own Domain-specific language models on a single GPU, and this motivated us to propose our model for self-disclosure. In the thesis experiments, the model is trained using Tesla V100 16GB GPU for about four days for 1 million steps, and this pre-trained model SD_ELECTRA is used for the next steps of fine-tuning and classification. In chapter 4, the experimental setup and results are discussed in detail.

3.5 Airbnb dataset

Airbnb is a peer-to-peer internet accommodation marketplace that facilitates monetary and social exchange between individuals [129]. Hosts can offer places for visitors to rent on Airbnb, such as rooms, flats, mansions, and even boats and castles. The guest is frequently a transient tourist who is unfamiliar with the host outside of Airbnb. Airbnb has 5.6 million listings and 900 million guests on the website as of this writing [130].

The hosts typically put their personal information on their profiles to attract the guests and make them comfortable. Given the importance of these profiles and the self-disclosure done by the hosts in the websites, we are using an Airbnb host profile dataset to address the self-disclosure issue in social media. This labeled data set consists of about nine types of disclosure specially labeled for the disclosure classes. Ma et al. studied various disclosures such as Interest, Personal Values, Education and Work, Relationships, Personality, Residence, Travel Plan, Hospitality, and other disclosures that do not fall in the disclosures mentioned earlier. In 2017, they published the dataset [128]. The data is labeled using binary labels '1' and '0,' which consists of about 1200 Airbnb host profiles and about 6000 sentences approximately. Fig. 12. Represents various disclosures plots, as mentioned by the Ma et al.

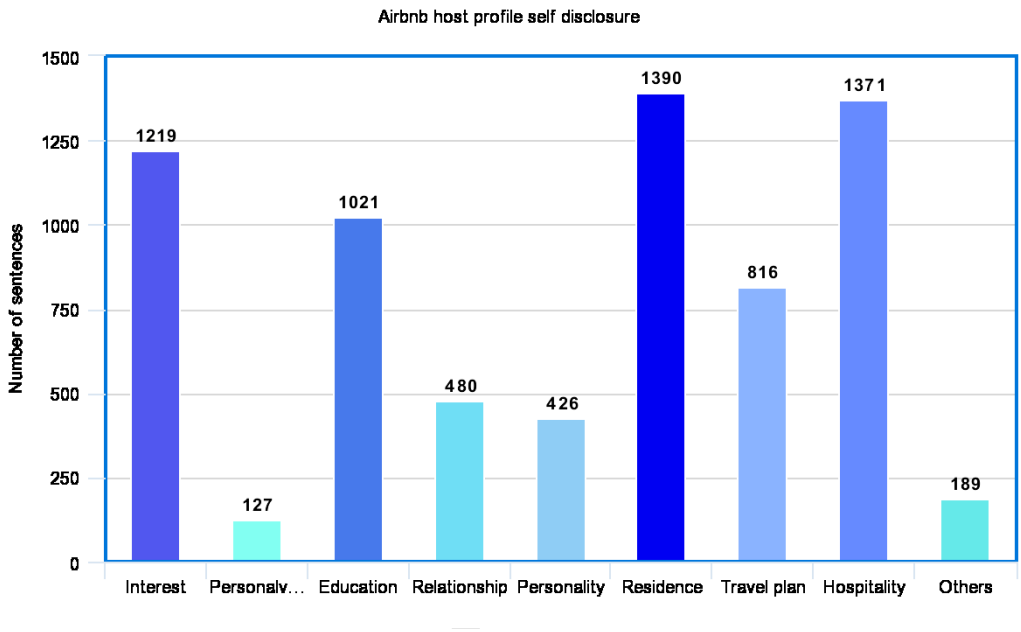

Fig. 12. Various disclosures plotted in the unbalanced Airbnb host profile dataset.

As discussed in Chapter2, Related work, the authors who previously published papers on self-disclosure using NLP [121] [123] worked on the OffMyChest data set. Nanyang Technological University from Singapore published this dataset, which has two types of disclosures: information disclosure and emotional disclosure. These authors proposed using the BERT model to predict two labels. This research interests to predict different types of disclosures as found in the earlier mentioned data set, which is one of its kind, which discusses multiple disclosures.

From Fig. 12, we can conclude that the percentage of Residence, Hospitality, and Interest disclosures is more than the other disclosures. In addition, hosts are interested in talking about relationships and education in their profiles which is distracted from the purpose of the Airbnb websites.

In this thesis, the Airbnb host profile dataset is used as a training and test dataset. When studying the dataset, it is found that the dataset is unbalanced, with various classes being distributed randomly. Also, the data set is a multi-label dataset as it is discovered that there are sentences that belong to more than one disclosure. Table 4. outlines the sample dataset representation of the Airbnb host profiles. Table 5. represents a sample of Interest disclosure in our considered dataset where Input text is used to predict output interest.

Table 4. Sample dataset representation of the Airbnb host profiles.

Interest	Personal values	Education & work	Relationship	Personality	Residence	Travel plan	Hospitality	Others
1	0	1	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0
0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0

Table 5. Example for Interest disclosure in Airbnb host profiles.

Input text	Interest
Artist dog lover!	1
I'm a grad student of Arts Management in Chicago	0
I'm from Shanghai China and hope to visit more places in the States	0
I'm always finding out what exactly I like, and I love to try new things	1
I enjoy playing and watching sports and listening to music…all types and all sorts!	1
Hello!	0
I am a well travelled guy who has been all over the world	0

The dataset mentioned above is divided into training and test data sets where the training dataset consists of 80% of the dataset with labels for the experimental purpose, and similarly, the test dataset consists of the remaining 20% of the dataset, which is evaluated without sending labels. The labels are predicted to find the scores of the model's performances. The next step is to fine-tune the pre-trained SD_ELECTRA using the above discussed datasets and evaluate the results. Once the model is pre-trained on a huge amount of unlabeled data, the pre-trained model can be fine-tuned with labeled data for other NLP tasks. This is how transfer learning works in transformer-based models such as ELECTRA, and it is discussed in detail in the next section.

3.6 Transfer Learning

Transfer learning is the process of extracting knowledge from one situation and applying it to another. Transfer learning in the form of pre-trained language models has become prevalent in NLP in less than a year and has contributed to state of art on a wide range of problems

[131]. Ruder, et al. in (2019) stated that different types of transfer learning could be classified into Transductive transfer learning and Inductive transfer learning, which can be further classified into Domain adaptation, Cross-lingual learning, multi-task learning, and Sequential transfer learning. Table 6. gives an overview of the transfer learning methods.

Table 6. Over-view of the transfer learning methods.

Methods	***Over-view***
Domain adaption	Different source Domains
Cross-lingual learning	Different languages
Multi-task learning	Tasks learned simultaneously
Sequential transfer learning	Tasks learned sequentially

So far, *sequential transfer learning* has resulted in the most significant gains. The Fig. 13. represents the general procedure followed in the sequential transfer learning.

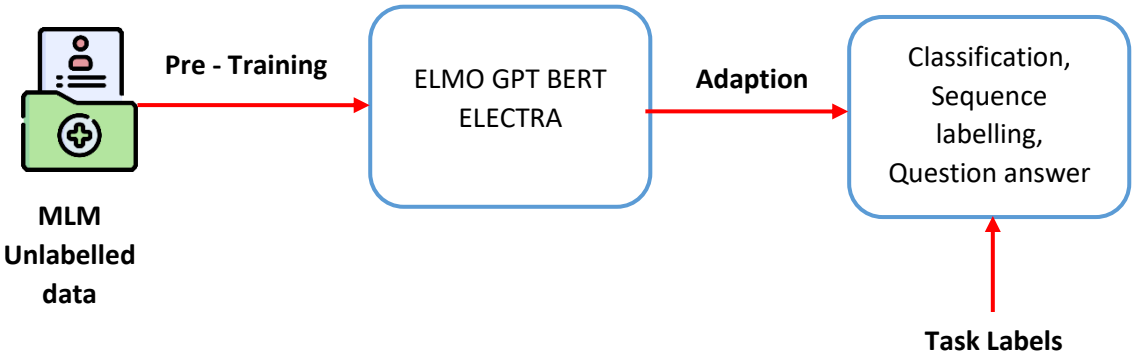

Fig. 13. procedure of sequential transfer learning in general [131].

Sequential transfer learning is the process of transferring knowledge through a series of steps, where the source and target tasks are not always the same. Sequential transfer learning consists of two stages. The model is trained on source data in the first phase of pre-training, and the source model is trained for the target task in the second step of adaptation. In the pre-training task, all models are pre-trained on a significantly large source data set that is, in the best-case scenario, highly similar to the target task. The adoption of the target task is the second step. The fundamental difference is whether the pre-trained model weights are retained (feature extraction) or changed to the target task (fine-tuning). The pre-trained models can be fine-tuned to different target tasks such as classification, sequence labeling,

question answer, and NER. As seen in this section, this standard procedure is to pre-train representations on a large unlabeled text corpus is our preferred method in this thesis, then adapt these representations to a supervised target task using labeled data.

3.6.1 Adaptation

There are various orthogonal directions we can decide on when adapting a pre-trained model to a target task: architectural adjustments, optimization strategies, and whether to gain more signal [132]. In our proposed method, we did not perform any architectural changes that deal with modifying the internal architecture of the pre-trained model. Instead, we mainly dealt with optimizing techniques that deal with choosing the weights that need to be updated and updating these weights accordingly.

The optimization model process can be further classified into *feature extraction* and *fine-tuning* [132]. In the feature extraction, the pre-trained weights are not changed. That is, on top of the pre-trained representations, a linear classifier is trained. The best results are usually obtained by learning a linear combination of layer representations rather than just representing the top layer. Pretrained representations can also be employed in a downstream model as features. Only the adapter layers are taught when adding adapters.

The other optimization technique is fine-tuning, in which the pre-trained weights are changed by using different learning algorithms [133]. Finally, the downstream model's parameters are initialized using the pre-trained weights. During the adaptation step, the entire pre-trained architecture is then trained. In the following sections, we discuss these methods in detail.

3.7 Fine- Tuning ELECTRA

However, there is a lot of research being done from recent years into how fine-tuning should be carried out. In *Computer vision*, methods proposed are typically freezing most of the network and fine-tune only the model's top layers [134]. Because NLP models are often shallower than their computer counterparts, which train one layer at a time, that is similar to layer-wise pretraining in deep learning networks. There are three methods proposed for fine-tuning for language models, such as BERT in NLP. These methods include training the entire architecture, training some layers while freezing others, and freezing the entire architecture [135].

Train the entire architecture

In this method a dataset is used to train the complete pre-trained model and pass the obtained results to a SoftMax layer. The error is backpropagated through architecture as a whole, in this case, and the model's pre-trained weights are adjusted depending on the new dataset.

Freezing some layers and training others

As proposed by Dodge et al. another option is to train a pre-trained model partially [135]. We can maintain the weights of the model's early layers static while retraining only the upper levels. We can experiment with how many layers to freeze and how many to train.

Entire Architecture is frozen

All of the model's layers are frozen and attach a few neural network layers of own and train the new model. During model training, just the weights of the associated layers will be updated.

In the proposed methodology, the first approach of fine-tuning is used, where the entire architecture of the pre-trained model of SD_ELECTRA is trained.

3.8 SD_ELECTRA

Our hypothesis is that privacy data and self-disclosure posts rely on language-specific structures, word occurrences, and vocabulary that aren't always captured by pre-training datasets. This approach in narrow domains such as bio-medical data, financial data, and propagandistic news articles. Our goal was to expose models to disclosure related data, similar to models trained on domain-specific corpora by Beltagy et al. (2019) on scientific text [136] and Andrei et al. (2020) on propagandistic and biased news articles [137].

Previous research has shown that employing in-domain text can bring extra benefits over general-domain language models in specialized domains [136] [137]. Transfer learning is successful when the target data is scarce and the source domain is highly relevant to the target one, according to Amittai et al. research (2011) [138]. Thus, in this thesis, we investigate domain-specific pre-training and its implications for downstream tasks.

Hence, SD_ELECTRA is pre-trained on a large amount of unlabeled Airbnb data and fine-tuned with labeled data on self-disclosure for text classification tasks. During fine-tuning, an extensive search has been performed on learning rates, and it is observed that smaller models perform better for larger learning rates.

By using the architecture discussed in this chapter, the Experiment is set up, and results are generated especially classifying self-disclosure in the social media text, which contributes to this thesis. The next chapter discusses in detail the experimental setup, the model hyper parameters, learning rates, and a detailed discussion on our results.

Chapter 4

Experimental Evaluation

In this chapter, the proof-of-concept of the state-of-the-art method is implemented. We discuss an experimental setup, including model configurations, hyperparameter settings, computational sources, learning rate adjustments, and details regarding all the evaluation results on considered evaluation metrics. It is observed that SD_ELECTRA could predict different types of disclosures with greater efficiency compared to the base model confirming that domain-specific language models can be trained from scratch with significantly lower compute resources.

4.1 Experimental Setup

As discussed in chapter 3, SD_ELECTRA is proposed to detect self-disclosure in social media datasets. The fine-tuning process of SD_ELECTRA (ELECTRA -small, uncased) involves tuning on five epochs, as in the typical procedure [52]. The entire model (14 million parameters), which has 12 layers and a hidden size of 256, is fine-tuned on text classification tasks. Two models are trained SD_ELECTRA_V1 and SD_ELECTRA_V2 on Tesla p100 GPU and Tesla V100 GPU, respectively, which have 16 GB RAM. Two different tokenizers, such as bert-base-uncased and faster version of distilbert-base-uncased, are used in the training process of SD_ELECTRA_V1 and SD_ELECTRA_V2, respectively. We experiment with two different models on two different systems with two different tokenizers to analyze the difference in training time of the model.

The following are the configurations of ELECTRA -small model used to train SD_ELECTRA.

- Model: ELECTRA-small
- Layers: 12
- Hidden Size: 256
- Parameters: 14M

As discussed in the chapter 3, section 3.2. the pretraining corpus is collected, and the summary is shown below.

- Number of Sentences: 41826506
- Size of the corpus: 5.8GB

Model Pretraining Summary

SD_ELECTRA_V1

Version 1, SD_ELECTRA_V1, is trained for 1 million training steps on the data size of 6GB corpus on Tesla P100 with a memory of 16 GB. The configuration values used are shown in Table 7.

Table 7. Configurations of the SD_ELECTRA_V1.

Configurations	***Value***
Tokenizer	bert-base-uncased
num_train_steps	1000000
vocab_size	30522
hidden_size	256
generator_size	0.25
max_seq_length	128

Below are the configurations in detail:

Optimizer: ELECTRA, as similar to BERT, also uses *AdamW optimizer.* It also employs a learning rate schedule that firstly warms up from 0 and then decays to 0.

Vocab_size: Defines the number of different tokens represented by the obj:" input_ids." We have used the default vocabulary of 30522.

Hidden_size: Dimensionality of the encoder layers and the pooling layer. It is set to 256 for our model.

Maximum_position_embedding: The maximum sequence length that this model might ever be used with. In our case, we are using a maximum sequence length of 128 for our Electra-small model.

Hidden_act: This is a non-linear activation function in the encoder and pooler. In our configuration, we have used “gelu,” which is the default activation function.

Using the previously mentioned configurations and other default parameters, SD_ELECTRA_V1 is trained for 1 million steps. It took almost six and half days to train the entire model, with the pre-training loss reduced, as shown in Fig. 14.

Fig. 14. represents the training process with an X-axis of the number of training steps and a Y-axis with the loss. It is observed that after 700k steps, the loss has been around 9, and since we increased the number of training steps to 1 million, our loss ultimately converged to 8.8.

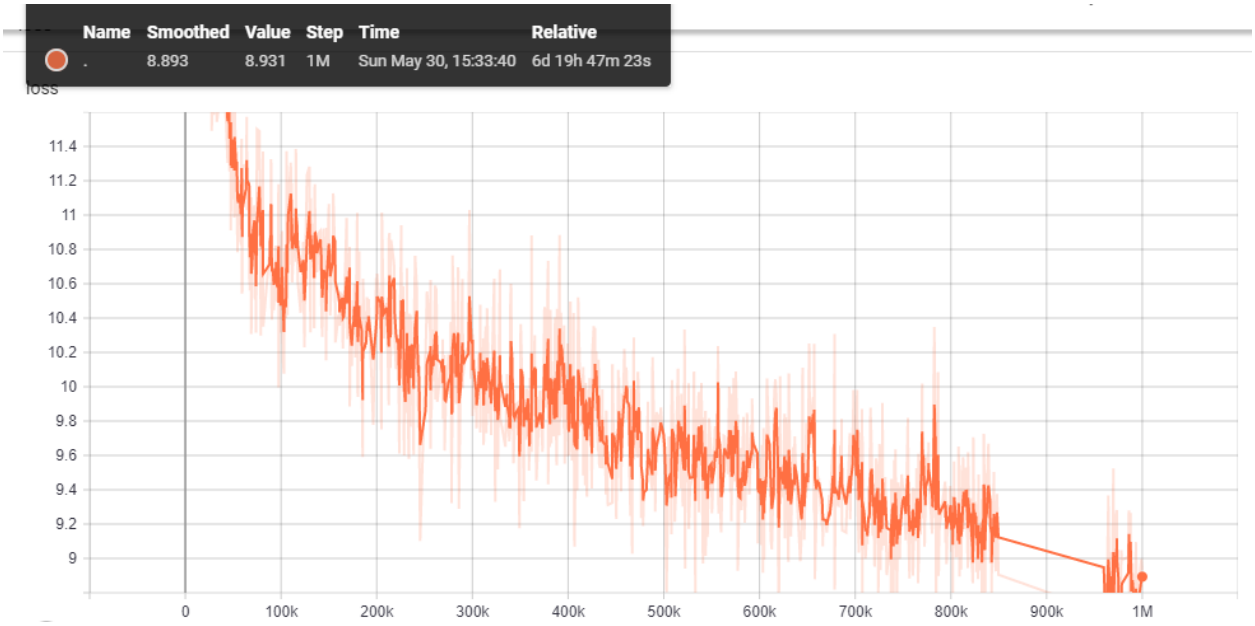

Fig. 14. Training results for version 1 of SD_ELECTRA on Tesla P100.

Transfer learning is used in the training process, where the pre-trained weights of steps are saved for every 200k steps, and the model is initiated from the pre-existing weights in the next stage. This method of saving weights is beneficial during training models using cloud GPUs such as google colab, where the process time out is either 12 hours or 24 hours.

SD_ELECTRA_V2

Version 2, SD_ELECTRA_V2, is trained for 1 million training steps on the data size of 6GB corpus on Tesla V100 with a memory of 16 GB. The configuration values used are tabulated in Table 8. The significant difference compared to the version 1 model configuration is that a fast tokenizer is of distilbert-base-uncased is used in version 2. These fast tokenizers significantly speed up, in particular when doing batch tokenization.

Table 8. Configurations of the SD_ELECTRA_V2.

Configurations	***Value***
tokenizer	distilbert-base-uncased
num_train_steps	1000000
vocab_size	30522
max_seq_length	128

Using the previously mentioned configurations, including distilbert-base-uncased tokenizer as a change compared to the original ELECTRA paper [52], SD_ELECTRA_V2 is trained for 1 million steps. It took almost five and half days to train the entire model, with the pre-training loss reduced, as shown in Fig. 15.

Fig. 15. represents the training process with an X-axis of the number of training steps and a Y-axis with the loss. It is observed that after 700k steps, the loss has been around 9, and since we increased the number of training steps to 1 million, our loss ultimately converged to 8.9.

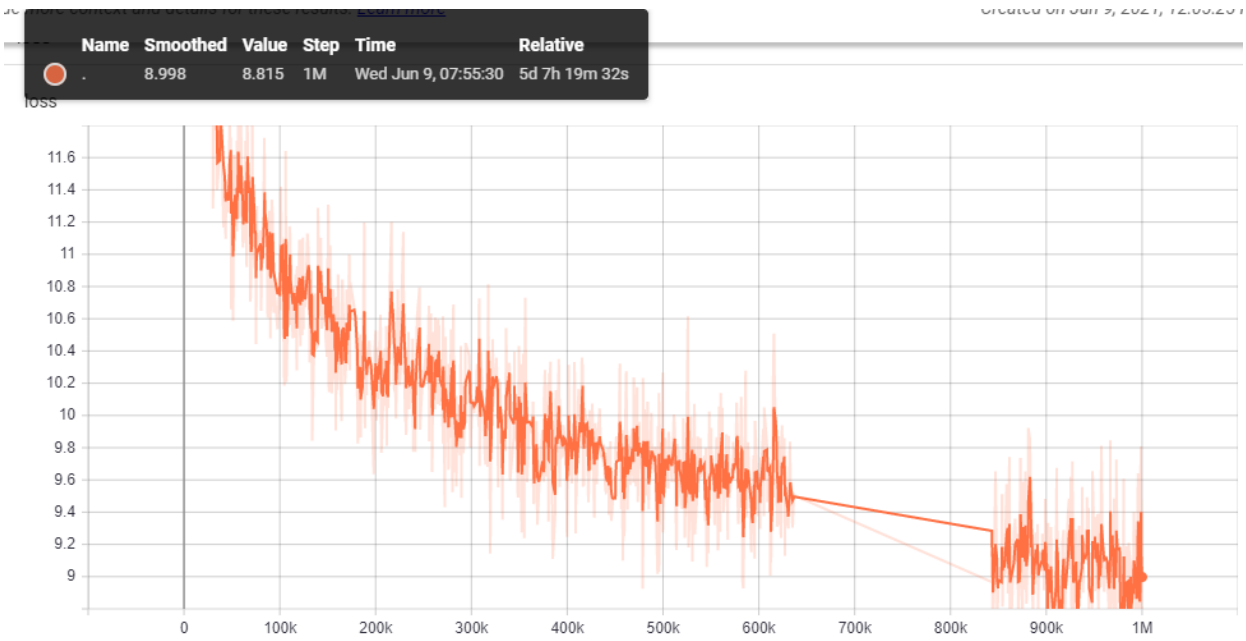

Fig. 15. Training results for version 2 of SD_ELECTRA on Tesla V100.

These pre-trained models of SD_ELECTRA_V1 and SD_ELECTRA_v2 are saved, and the discriminators are fine-tuned for the downstream task, text classification. In the next section, we presented the models' experimental results compared with the base model ELECTRA-small.

4.2 Experiment Results

From the previously mentioned section, it is witnessed that SD_ELECTRA_V2 is trained faster than SD_ELECTRA_V1 as there is a difference in the GPU versions. Thus, it can be concluded that the larger the computational resource, the faster the model is trained. Also, It is to be noted that both the models are trained on a single GPU system, whereas larger language models such as BERT use multiple GPUs and TPUs. Thus, we can say that this increasing computation time and expense highlight a need to develop computationally efficient models with reduced or faster computing retaining the top modeling power. From the results, it can be observed that SD_ELECTRA is computationally efficient. This section observes that the SD_ELECTRA_V2 model trained using distil-bert-base tokenizer performs better than the SD_ELECTRA_V1 trained on the bert-base tokenizer.

This section compares the experimental results with the already existing ELECTRA, and it is prominent that domain-specific ELECTRA performs a bit more efficiently than the google proposed ELECTRA [95].

The generator is often destroyed after training when utilizing the ELECTRA pre-training methodology, and only the discriminator is used. Simple Transformers library will save the generator and discriminator separately once training is completed to help with this [139]. If training is stopped before it is finished, the model will be saved as a Language Modeling Model, including the discriminator and the generator. It is also possible to extract the discriminator and generator independently. By using this technique, discriminators of SD_ELECTRA_V1 and SD_ELECTRA_V2 are saved separately, which are used to publish results on the training dataset.

The saved SD_ELECTRA models are just trained on MLM objectives, and they cannot predict the class labels. As mentioned in chapter 3, both the versions of SD_ELECTRA are fine-tuned on Airbnb host profile labeled data for disclosures [128] for the text classification task, and the results obtained are tabulated (Table 9.). SD_ELECTRA_V1 and SD_ELECTRA_V2 are fine-tuned with a learning rate of 0.0001 and 5 train epochs. Google's published ELECTRA-small model is considered the base model, which is fine-tuned on the same learning rate and epochs mentioned earlier. The results are also compared with higher-end models such as ELECTRA-

base [52], which is fine-tuned on 0.0001 learning rate with three epochs, and Distil BERT model, which is fine-tuned on $3e^{-5}$ learning rate for three epochs [126].

Evaluation metrics such as *Global accuracy, Macro-averaging F1 score, Macro-averaging Precision,* and *Macro-averaging Recall* are used for comparing model performances. The notations used to represent these metrics in our thesis are accuracy, F1 score, precision, and recall.

In the Macro averaging precision and recall metric, the global scores by averaging individual classes are calculated. Some detailed explanations of the metrics are presented in the following.

Confusion Matrix: The predicted vs actual classification can be tabulated as confusion matrix represented in Table 9.

Table 9. Representation of confusion matrix.

		PREDICTED CLASS	
		Negative	Positive
ACTUAL CLASS	Negative	TN	FP
	Positive	FN	TP

True Positives (TP): These are the correctly predicted positive values.

True Negatives (TN): These are the correctly predicted negative values.

False Positives (FP): These are the incorrectly predicted positive values.

False Negatives (FN): These are the incorrectly predicted negative values.

Accuracy: It is a ratio of correctly classified observations over the total observations. It can be represented in equation (6).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

Accuracy is not a great measure when datasets are non-symmetric. Therefore, other metrics are required to calculate the performance of the model.

Precision: It is a ratio of correctly predicted positive observations to the total predicted positive observations as represented in equation (7).

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

Recall: It is a ratio of correctly predicted positive observations to all the observations in the actual class, as shown in equation (8).

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

F1 score: It is a weighted average of Precision and Recall. The formula to calculate F1 score is given in equation (9).

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

It is observed from Table 10. that SD_ELECTRA_V1 is showing an increase in F1score, Recall, and Precision compared to the google proposed ELECTRA model. On the other hand, the version two model proposed, SD_ELECTRA_V2, has outperformed the base model in all the metrics considered.

Table 10. Scores of our SD_ELECTRA compared with base model and other models.

Models	***Accuracy***	***F1 score***	***Recall***	***Precision***
Google Electra -small	93.96%	0.7019	0.6710	0.7495
SD_ELECTRA_V1	93.75%	0.7197	0.6966	0.7547
SD_ELECTRA_V2	**94.13%**	**0.7120**	**0.6741**	**0.7602**
BERT-base	94.01%	0.7292	0.7174	0.7433
Google Electra-base	94.57%	0.7574	0.7189	0.7889
Distil bert-base	94.41%	0.7415	0.7152	0.7817

When comparing results, SD_ELECTRA_V2 has a considerable gain in Accuracy, Precision, and F1score and a minor rise in the Recall when compared to base model ELECTRA-small. With comparison with BERT-base model, SD_ELECTRA_V2 achieves better accuracy and precision, whereas BERT performed better in F1score and Recall. As mentioned in Section 4.1, we have used ELECTRA-small configurations, which have a hidden size of 256 and 14M parameters, to train SD_ELECTRA and hence we and hence we compare our results with the base model ELECTRA-small for evaluating performance of SD_ELECTRA. Table 11. shows the performance of best-performed model SD_ELECTRA_V2 on individual labels.

Table 11. Performance of SD_ELECTRA_V2 on individual labels.

Label Name	***Accuracy***	***F1 score***	***Recall***	***Precision***
Interest disclosure	90.89%	0.7806	0.7612	0.8009
Personal Values disclosure	97.33%	0.4615	0.3428	0.7058
Education & Work disclosure	97.04%	0.8556	0.8535	0.8578
Relationship disclosure	96.47%	0.7909	0.7070	0.8974
Personality disclosure	94.47%	0.6027	0.5432	0.6769
Residence disclosure	91.42%	0.8387	0.8731	0.8461
Travel plan disclosure	95.14%	0.8486	0.8773	0.8218
Hospitality disclosure	87.90%	0.7669	0.8038	0.7570
Others	96.47%	0.3728	0.3055	0.4782

Table 12. represents the metric scores of the base model used Google ELECTRA-small on individual metrics. Again, it can be noticed that SD_ELECTRA_V2 outperformed the base model in predicting individual label metrics.

Table 12. Evaluation results for base model Google ELECTRA-small.

Label Name	***Accuracy***	***F1 score***	***Recall***	***Precision***
Interest disclosure	91.04%	0.7892	0.7927	0.7857
Personal Values disclosure	97.33%	0.33	0.25	0.5
Education & Work disclosure	95.23%	0.8677	0.8282	0.9111
Relationship disclosure	96.19%	0.8105	0.7777	0.8472
Personality disclosure	95.23%	0.666	0.6172	0.7246
Residence disclosure	91.61%	0.8339	0.8246	0.8435
Travel plan disclosure	94.76%	0.8424	0.8305	0.8546
Hospitality disclosure	87.61%	0.7711	0.7849	0.7577
Others	96.66%	0.4067	0.3333	0.5217

SD_ELECTRA_V1 and SD_ELECTRA_V2 are also compared with Google ELECTRA-base, BERT and Distil-bert. However, SD_ELECTRA_V2 could not achieve performance higher than ELECTRA-base since it has been trained by adopting techniques from ELECTRA-small as published by Clark, et al. in 2020. As part of future work, we would train higher version of SD_ELECTRA using ELECTRA-base configurations, which have a hidden size of 768 and 110M parameters, and a larger version of SD_ELECTRA using ELECTRA-large configurations, with hidden size of 1024 and 335M parameters. We would also train and experiment with models with different hyper parameters and maximum sequence lengths.

4.3 Analysis of potential risks associated with each disclosure

This section presents the analysis of underlying risks associated with each type of self-disclosure predicted using SD_ELECTRA on our considered dataset. Since self-disclosure is a privacy concern, there are privacy-related risks associated with the disclosure of personal information. The dataset consists of disclosures such as Interest, Personal Values, Education and Work, Relationships, Personality, Residence, Travel Plan, Hospitality, and other disclosures [128]. Table 13. gives a detailed list of different types of disclosures and the corresponding risks involved.

Table 13. List of Types of disclosures and corresponding potential risks.

Type of disclosures	Potential Risks
Interest	In social media, sharing interests can be beneficial as it will allow users to connect with like-minded people. However, there is also a possibility of connecting with people who have wrong intentions. For, example children and teens are vulnerable to share information about interests and hobbies online, which provides an easy way for offenders to harass, stalk and bully them.
Personal Values ,Personality and Relationships	People tend to share their behaviours, thinking, goals, achievements and relationships online to motivate others and sometimes to attract attention. One of the most significant internet risks is sharing personal information with people unknown. Sometimes, personal information might be used to crack a password or answer security questions correctly. Also, nowadays, a notable amount of peer pressure is observed in users who tend to follow people online, adding to their emotional stress.

Education and Work	Education and work details are being shared online for various purposes, such as job profiles, interviews, and college admissions. Sometimes, the details about work history, finances, and degree can be ended up in the wrong hands. There can be a possibility of identity theft using these details. Huge scams can be possible where people are approached for work-related queries and tend to attain other sensitive information.
Residence	Location disclosure is the biggest threat to individuals. People tend to share online photos, geo-tagging, and residence addresses online, allowing criminals to reach them easily. There are a considerable number of cases worldwide where stalking, harassment, theft, and crimes are recorded because of location disclosure.
Travel plan and hospitality	Similar to the other disclosures disclosing travel information, hospitality behaviours would attract the wrong attention of predators. Since the information posted online cannot be deleted and visible globally, there is a chance of theft at homes based on unavailable people and other crimes taking place.

Many other types of disclosures on social media are impacted by phishing, scams, and other types of cybercrimes. Hence, in this thesis, we have analyzed potential privacy risks according to our knowledge of self-disclosures. In future work, we target to disclose both self-disclosure and the corresponding privacy risk to the users using the user interface designed and provide awareness on privacy-related problems.

4.4 User Interface

Furthermore, a user-interface application is designed using framework streamlit to test SD_ELECTRA in the functional interface [54]. Currently, the Beta version of the software is released, and this is used to create a web application, merging the python language model with the front end-user interface. Fig. 16. shows the front-end screen of our web application.

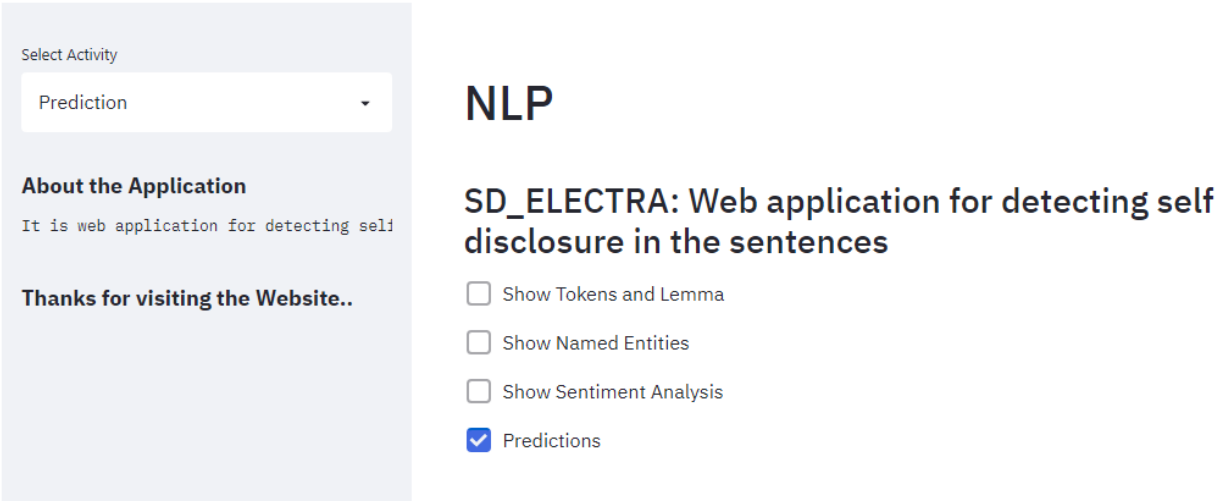

Fig. 16. User interface proposed to test our proposed methodology.

This application is analyzed further where users can type a sentence in a text box, and predictions of the type of disclosure existing in the user sentence are published. Other Natural language features such as tokenization, identifying named entities, and sentiment analysis of the sentences are included in the application for deeper analysis. Fig. 17. shows an example of the disclosure class identified for the user given sentence. The interface is also equipped with an application where the grammar of the sentence is auto-corrected.

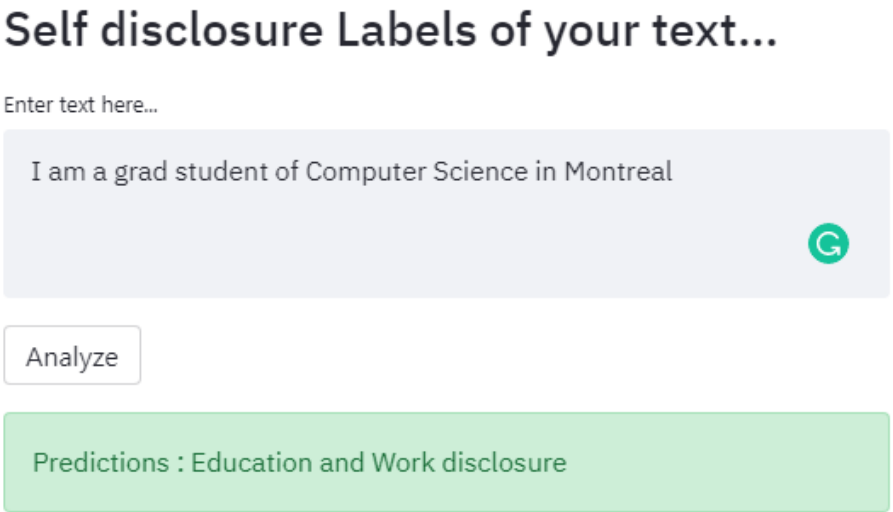

Fig. 17. Example of predictions for the user typed sentence.

It is observed that the model was able to predict correct classes for about 90% of the sentences from the considered test dataset. Fig. 18. shows the predictions of the model for multiple sentences. It is noted that the model is capable of also predicting multiple classes.

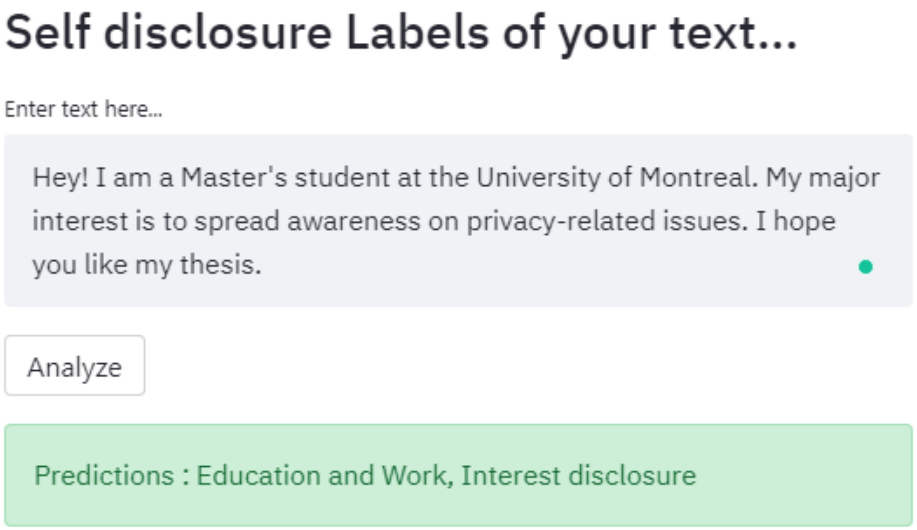

Fig. 18. Example of predictions for the multiple sentences.

Though the model could generate correct predictions with greater efficiency, we also observed incorrect predictions when the user enters text that the model does not recognize. Fig. 19. shows one such example of false class predicted. The predicted class is Education and work disclosure, whereas the actual class is Personal Values disclosure.

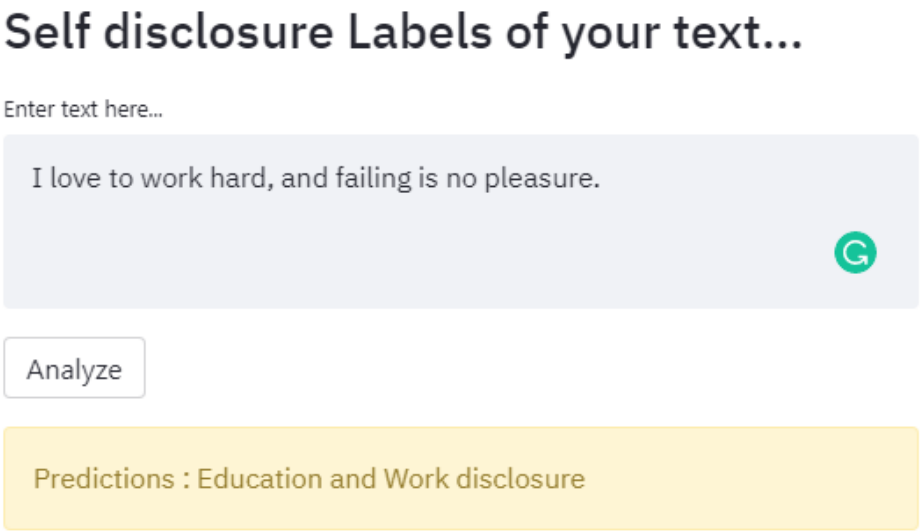

Fig. 19. Example of incorrect predictions.

Another limitation noticed is that since the language models are extensive and need faster machines to compute, the web application took more time to show predictions while running

on a CPU machine. More examples of multiple options in the user interface are demonstrated in Appendix C.

4.5 Discussion

As discussed in the previous sections, the experimental results have produced a significant improvement on considered metrics compared to the base model ELECTRA-small. The contribution for this thesis work includes pre-training a language model from scratch on a corpus of 6 GB and fine-tuning the model for text classification, and achieving significantly higher results when trained on context-specific data. It is a known fact that bigger language model such as BERT is trained on large corpus such as English Wikipedia [95]. The proposed model SD_ELECTRA has considered a significantly small corpus and achieved performance better than the existing ELECTRA model.

Clark et al. have published that ELECTRA-small was able to outperform BERT-small by 5 points on GLUE. SD_ELECTRA_V2 have bettered the published ELECTRA-small results, and also SD_ELECTRA_V2 has also exceeded the performance of BERT-base in Accuracy and Precision. Future work involves exploring larger ELECTRA models on larger data sets and comparing the performance with existing language models.

Advantages of SD_ELECTRA over other models

Significant advantages of the proposed model when compared to the other pre-trained models considered are discussed in detail in the following paragraphs.

Quickest during fine-tuning

The SD ELECTRA model trains the quickest during fine-tuning and requires the least amount of memory, as well as being the quickest during inference. The speed of the model training depends on the size (including the number of parameters) of the model. The SD_ELECTRA and ELECTRA-small being smaller in size, are quicker than the rest of the models considered for comparison in our results. Table. 14. compares the relative training times during fine-tuning for different models .

Table 14. Comparison of training time of models

Model Name	Training time
SD_ELECTRA_V2	19m 37s
Google ELECTRA-small	20m 9s
Distil-bert-base	32m 58s
Google ELECTRA-base	1h 20m 51
BERT-base	1h 21m 16

From the above results, we can infer that the proposed method is faster in training and inference. In applications such as interacting with users, quicker models are preferable to reduce the response's waiting time.

Better convergence in training loss

Fig. 20. below represents the comparison of fine-tuning training loss between ELECTRA-small and SD_ELECTRA for 20 epochs. The graph shows that SD_ELECTRA converges better than ELECTRA-small and demonstrates that SD_ELECTRA performance on fine-tuning training data is significantly better.

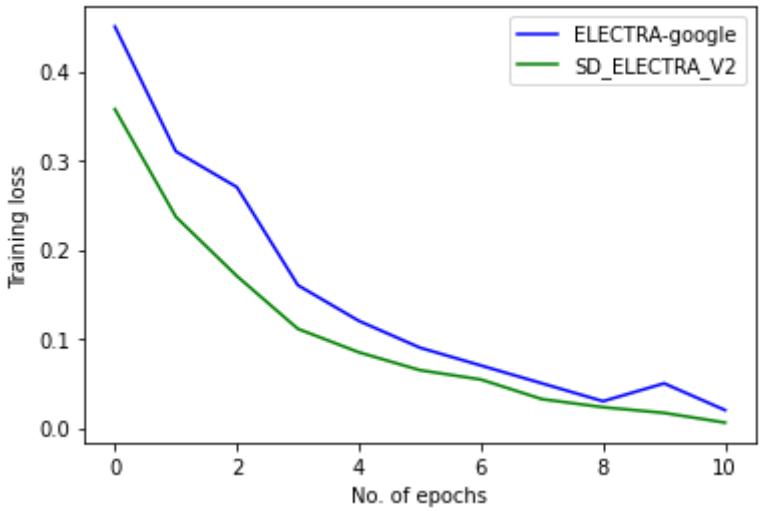

Fig. 20. Plot showing training loss of SD_ELECTRA_V2 and Google ELECTRA-small

A loss is a number that represents how bad the model's prediction is in a single example. In other terms, lowering the loss would give better predictions.

Performs better in predicting classes for certain pattern of examples

While analyzing the performance on sentence-level using Google ELECTRA-small and SD_ELECTRA, we found that Google ELECTRA-small gave wrong predictions for a certain pattern of disclosures, such as hospitality, as shown in Table 15.

Table 15. Pattern of examples where Google ELECTRA-small fails to predict the correct class.

Sentence	Original class	Predicted class
The satisfaction of hosting comes in the process of knowing you're going to enhance someone's travel experience and make them happier!	Hospitality	Travel_plan
I'm new to Airbnb but, I'm originally from Mississippi so, I like to think that hospitality is in my DNA.	Residence, Hospitality	Residence
My goal is to make every fellow traveler feel at home in the City of Angels.	Residence, Hospitality	Residence
We believe in the culture of travel, trust and community and feel Airbnb offers an amazing opportunity for people like us to connect with others.	Travel_plan, Hospitality	Personality

On the same examples, it is observed that SD_ELECTRA was able to predict the classes correctly. From the above pattern, we can suggest that domain-specific models can perform better on a specific type of data associated with their domain, in our case, privacy-related data. In future work, we would test our model on different disclosure-related datasets and analyze the performance of our proposed model.

Challenges

While designing and implementing SD_ELECTRA, there were challenges such as not obtaining desired results when training the model for 200k steps. The model was trained step by step by creating smaller models, for 200k, 500k, and 700k steps to address this challenge. It was noticed that increasing the duration of training by increasing the number of steps has increased the performance gradually. This thesis also studied the relation between GPU and the training process; with higher RAM, the training process was quicker.

Comparison with state of art models

Since very little work is accomplished in the field of self-disclosure detection using language models, the process of finding references and labeled datasets was a tedious process. Though few references are found to compare our results, those studies have been performed on a completely different dataset. In Table 16. we compared methods and metrics of the proposed methodology with different authors who worked on self-disclosure using natural language processing [120] [121] [123].

Table 16. Comparison of SD_ELECTRA with other state of art models.

Authors	Akiti et al. 2020 [120]	Akiti et al. 2020 [121]	Dadu et al. 2020 [123]	Proposed model (2021)
Models	BERT, CNN	FrameSemantic Model	Ensemble Roberta, Alberta	SD_ELECTRA
Data Set	OffMyChest	OffMyChest	OffMyChest	Airbnb Host profiles
Accuracy	NA	NA	85.55%	94.13%
F1-score	0.525	0.64	0.558	0.7120
precision	0.45	0.59	0.623	0.7602
Recall	0.655	0.69	0.515	0.6741

Benefits and Limitations

The proposed methodology is one of its kind, extending the thesis contribution novel to the best of our knowledge. The intention behind using the dataset, as mentioned in section 3.5. is to explore different types of disclosures such as Interest disclosure, Personal Values disclosure, Education and Work-related disclosure, Relationships, Personality disclosure, Residence disclosure, Travel Plan disclosures, and Hospitality. The dataset considered in this thesis is one of its kind, and it is mainly designed by the authors to analyze various disclosures. The proposed methodology has been successful in predicting these classes with greater

efficiency. It is noticed from the training dataset that though it is related to the host profiles of Airbnb where users describe themselves, the users have significantly shared information deviating from the original purpose. The Data analysis shows that users have shared interests, education, relationships, and other personal disclosures in excess.

One of the research objectives is to design an interpretive user platform to combine SD_ELECTRA with a front-end web application to test the model on real-time social media posts. With the latest developments in technology, we built a web application using python as the backend technology (not directly accessible by the user and is often used to store and manipulate data) and combined it with the interactive layer. As a result, SD_ELECTRA has performed significantly better on the considered fine-tuning test data set and predicted the categories for real-time user data with reasonable accuracy.

While many research studies as stated in Table 12. in this area mainly focus on classifying textual data as information or emotional disclosure, only a few are concerned with other types of disclosures. Addressing the need of disclosure detecting solutions, In this thesis, we have built SD_ELECTRA, which is trained on domain-specific data to identify various types of disclosures. The proposed solution is also integrated with a web application which detects disclosures as users types their text messages making it a global solution for end-users privacy management problem. We have also analyzed privacy risks associated with every disclosure identified to identify serious dangers involved while performing self-disclosure. As part of future work, we target to build a user solution that identifies the disclosure and notifies users about the underlying risks.

The limitation of the proposed design is that SD_ELECTRA is slow in giving predictions while running on a CPU. In the future, we would dive deeper and enhance the speed of model predictions and design a potentially stable interface that can detect privacy-related disclosures instantly.

In the next chapter, we conclude the thesis, and discuss all the achieved results of the objectives, along with the future aspects of the research work.

Chapter 5

Conclusion and Future Work

This chapter summarizes the thesis by justifying all the research goals outlined in chapter 1 and providing insight toward future studies.

5.1 Conclusion

The main objective of this thesis was to detect various self-disclosures by users in social media platforms, which is indeed a privacy concern by using Natural Language Processing techniques. More specifically, the following objectives were attained:

- Proposing a method to detect self-disclosure from a user-generated text on social media, using natural language processing. In Section 3.1, we presented the steps of the methodology to achieve this objective. With these steps, we design the architecture step by step using a language model. First, unlabeled data was collected from AIRBNB and then a large pre-training corpus was built, as explained in section 3.2.
- Using current natural language processing techniques, designing an effective text classification model SD_ELECTRA (Self-disclosure ELECTRA). For this purpose, pre-processing methods were initially performed as explained in section 3.3, including tokenizing (3.3.1) and creating vocabulary (3.3.2). Then, the language model SD_ELECTRA was pre-trained by applying the training strategy presented in section 3.4.2. Following that, the model was fine-tuned on the dataset discussed in 3.5.
- Identifying the disclosure category, including Interest disclosure, Personal disclosure, Education and Work disclosure, Relationship disclosure, Personality disclosure, Residence and Travel plan disclosure, and Hospitality disclosure. For this purpose, the entire pre-trained architecture was trained using considered training data (section 3.7.). In addition, an extensive search on learning rates was performed to achieve different disclosure categories with higher performance.
- Experimenting and evaluating the designed methodology on different metrics improve performance compared with the base models. For this objective, an

experimental setup is implemented, as presented in section 4.1. Later, various experiments are conducted on the test data set and recorded the performance of SD_ELECTRA_V1 and SD_ELECTRA_V2 on considered metrics as in section 4.2. It is observed that SD_ELECTRA_V2, a context-specific model, performs better in all the considered metrics. The advantages of SD_ELECTRA are described in section 4.5 in detail.

- Analyzing the potential risks of disclosing information online and discussing risks associated with each disclosure in detail. For this purpose, section 4.3. (Table. 13) discusses the disclosures considered in this thesis and their underlying privacy concerns in detail. As part of future work, we target to display disclosure and their corresponding risk to users to create awareness and warn them against disclosing personal information.
- Designing an ideal user interface platform where the SD_ELECTRA is combined with a front-end web application to test the method's feasibility on real-time social media posts. As demonstrated in section 4.4 (Fig. 16.), a web application is designed to achieve this objective. In addition, the designed application can illustrate multiple NLP techniques such as tokenization, NER and analyze the disclosures in user data. As part of future prospective, we would build a complete NLP application that tests different techniques of NLP along with different models allowing users to use a single application for multiple purposes.
- Evaluating user interface on correctness to identify disclosures and enlighten users about the possible disclosures in their social media posts. This objective is accomplished by evaluating the user interface with real-time social media posts, as shown in section 4.4. (Fig. 17.) and predicting disclosure categories. In this way, we establish an awareness platform for users to display real-time disclosure categories.

To summarize, we built a language model specially trained on social media data such as Airbnb in this thesis. This thesis aims to create a context-specific language model that outperforms general language models on Airbnb reviews and comments. SD_ELECTRA is trained using the Airbnb corpus and fine-tuned using the Airbnb host profile dataset to achieve better efficiency. We also presented two models with different versions, comparing their outcomes on NLP downstream tasks like text classification.

Training language model from scratch is tedious work, and it requires TPUs and GPUs with very high computing power to train them from scratch. This work addresses high computational requirement concerns by training the SD_ELECTRA model on a single GPU that took approximately six days to train on readily available GPU such as Tesla P100 and five days on Tesla V100. The main intention behind this objective is to discover the possibilities of pre-training language models on big corpora, eventually reducing computation cost and achieving better performance.

SD_ELECTRA was tested on the considered test data set, and this method shows state-of-the-art performance in results. It is evident from the manifested results that SD_ELECTRA addressed all research objectives.

In this thesis, an illustrative user interface solution implemented enables social media users to test reviews or posts with any disclosures. This way, an option for the users is provided to examine any privacy breach before posting on the Internet. Finally, as discussed earlier in section 4.5, future work is needed to focus on enhancing faster predictions and experiment with additional datasets.

5.2 Future Work

The following are the future aspirations considered to accomplish:

- Training distinct models using Google's proposed Electra-base and Electra-large models as base models and predict the results. Improving the sequence length and training more expensive models referring to the existing configurations by enhancing the efficiency of the predictions also explores the computation resources needed for higher-end models.
- Fine-tuning SD_ELECTRA on other NLP tasks, such as Named Entity Recognition and Question Answering and comparing the model predictions along with evaluating the results on different datasets.
- Disclosing both self-disclosure and the corresponding privacy risk to the users using the user interface designed and provide awareness on privacy-related problems.

References

[1] F. Kittler, “The History of Communication Media,” *CTheory,* pp. 6-10, 1996.

[2] L. Kleinrock, “An early history of the internet [History of Communications],” *IEEE Communications Magazine,* pp. 26-36, 2010.

[3] J. S. Cook, “History of Supercomputing and Supercomputer Centers,” in *Research and Applications in Global Supercomputing*, 2015, pp. 33-55.

[4] T. Berners-Lee, “The world-wide web,” *Computer Networks and ISDN Systems,* pp. 454-459, 1992.

[5] N. Couldry, Media, society, world : social theory and digital media practice, Cambridge: Malden, MA : Polity, 2012.

[6] M. Hillbert and P. Lopez, “The World’s Technological Capacity to Store, Communicate, and Compute Information,” *Science, Volume 332, Issue 6025,* pp. 60-65, 2011.

[7] F. Xia, C.-H. Hsu, X. Liu, H. Liu, F. Ding and W. Zhang, “The power of smartphones,” *Multimedia Systems,* p. 87–101 , 2015.

[8] A. Kaplan and M. Haenlein, “Users of the World, Unite! The Challenges and Opportunities of Social Media,” *Business Horizons ,* pp. 59-68, 2009.

[9] c. Pew research, “World Wide Web Timeline,” March 2014. [Online]. Available: https://www.pewresearch.org/internet/2014/03/11/world-wide-web-timeline/.

[10] M. Iqbal, “facebook-statistics,” 24 May 2021. [Online]. Available: https://www.businessofapps.com/data/facebook-statistics/.

[11] K.-Y. Lin and H.-P. Lu, “Why people use social networking sites: An empirical study integrating network externalities and motivation theory,” *Computers in Human Behavior,* pp. 1152-1161, 2011.

[12] K. Quinn, “Why We Share: A Uses and Gratifications Approach to Privacy Regulation in Social Media Use,” *Journal of Broadcasting & Electronic Media,* pp. 61-86, 2016.

[13] G. Blank, G. Bolsover and E. Dubois, “A New Privacy Paradox: Young People and Privacy on Social Network Sites,” in *Annual Meeting of the American Sociological Association*, San Francisco, 2014.

[14] M. Madden, A. Lenhart, S. Cortesi, U. Gasser, M. Duggan, A. Smith and M. Beaton, “Teens, Social Media, and Privacy,” 21 May 2013. [Online]. Available: https://www.pewresearch.org/internet/2013/05/21/teens-social-media-and-privacy/.

[15] F. Bélanger and R. E. Crossler, “Privacy in the Digital Age: A Review of Information Privacy Research in Information Systems,” *MIS Quarterly,* pp. 1017-1041, 2011.

[16] E. Horvitz and D. Mulligan, “Data, privacy, and the greater good,” *Science,* pp. 253-255, 2015.

[17] S. J. Milberg, S. J. Burke, H. J. Smith and E. A. Kallman, “Values, personal information privacy, and regulatory approaches,” *Communications of the ACM,* pp. 65-73, 1995.

[18] J. Phelps, G. Nowak and E. Ferrell, “Privacy Concerns and Consumer Willingness to Provide Personal Information,” *Journal of Public Policy & Marketing,* pp. 27-41, 2000.

[19] C. V. Slyke, J. T. ShimRichard, D. JohnsonRichard, D. JohnsonJames and J. Jiang, “Concern for Information Privacy and Online Consumer Purchasing,” *Journal of the Association for Information Systems ,* pp. 415-444, 2006.

[20] H. J. Smith, T. Dinev and H. Xu, “Information Privacy Research: An Interdisciplinary Review,” *MIS Quarterly,* pp. 989-1015, 2011.

[21] S. Thielman, “Yahoo hack: 1bn accounts compromised by biggest data,” New York, 2016.

[22] R. McMillan and R. Knutson, “Yahoo Triples Estimate of Breached Accounts to 3 Billion,” New York, 2017.

[23] N. Confessore, “Cambridge Analytica and Facebook: The Scandal and the Fallout So Far,” New York, 2018.

[24] S. Vaidhyanathan, Antisocial Media: How Facebook Disconnects Us and Undermines Democracy, New York: Oxford University Press, 2018.

[25] H. Tuttle, “Concerns, Facebook Scandal Raises Data Privacy,” *Risk Management,* 2018.

[26] H. Wijoyo, N. Limakrisna and S. Suryanti, “The effect of renewal privacy policy whatsapp to customer behavior,” *Insight Management Journal,* pp. 26-31, 2021.

[27] Dan Goodin, “WhatsApp gives users an ultimatum: Share data with Facebook or stop using the app,” 2021.

[28] K. Bhalla, “whatsapp-claims-chats-to-stay-encrypted-as-privacy-scrutiny-intensifies,” 11 Jan 2021. [Online]. Available: https://inc42.com/buzz/whatsapp-claims-chats-to-stay-encrypted-as-privacy-scrutiny-intensifies/.

[29] C. Sindermann, B. Lachmann, J. D. Elhai and C. Montag, “Personality associations with WhatsApp usage and usage of alternative messaging applications to protect one’s own data.,” *Journal of Individual Differences,* pp. 2151-2299, 2021.

[30] F. Bert, M. R. Gualano, E. Camussi and R. Siliquini, “Risks and Threats of Social Media Websites:Twitter and the Proana Movement.,” *Cyberpsychology, Behavior, and Social Networking,* pp. 233-238, 2016.

[31] A. Acquisti and R. Gross, “Predicting Social Security numbers from public data,” *proceedings of national academy of sciences of the United States of America,* pp. 10975-10980, 2009.

[32] J. E. Phelps, G. J. Nowak, G. J. Nowak and E. Ferrell, “Privacy Concerns and Consumer Willingness to Provide Personal Information,” *Journal of Public Policy & Marketing,* pp. 27-41, 2000.

[33] M. Aljohani, A. Nisbet and K. Blincoe, “A survey of social media users privacy settings & information,” in *Australian Information Security Management Conference*, Perth, Western Australia, 2016.

[34] G. R. Milne, “Privacy and Ethical Issues in Database/Interactive Marketing and Public Policy: A Research Framework and Overview of the Special Issue,” *Journal of Public Policy & Marketing,* p. 1–6, 2000.

[35] V. J. Derlega, S. Metts, S. Petronio and S. T. Margulis, Sage series on close relationships., Thousand Oaks, CA, US: Thousand Oaks, CA, US: Sage Publications, Inc, 1993.

[36] N. L. Collins and L. C. Miller, “Self-disclosure and liking: A meta-analytic review,” *Psychological Bulletin,* p. 457–475, 1994.

[37] M. Luo and J. T. Hancock, “Self-disclosure and social media: motivations, mechanisms and psychological well-being,” *Current Opinion in Psychology,* pp. 110-115, 2020.

[38] F. Stutzman, R. Capra and J. Thompson, “Factors Mediating Disclosure in Social Network Sites,” *Computers in Human Behavior,* pp. 590-598, 2011.

[39] A. Acquisti and R. Gross, “Imagined Communities:Awareness, Information Sharing, and Privacy on the Facebook,” in *Conference: Privacy Enhancing Technologies*, Cambridge, 2006.

[40] A. Gruzd and Á. Hernández-García, “Privacy Concerns and Self-Disclosure in Private and Public Uses of Social Media,” *Cyberpsychology, Behavior, and Social Networking,* pp. 418-428, 2018.

[41] M. Taddicken, “The ‘Privacy Paradox’ in the Social Web: The Impact of Privacy Concerns, Individual Characteristics, and the Perceived Social Relevance on Different Forms of Self-Disclosure,” *Journal of Computer-Mediated Communication,* p. 248–273, 2014.

[42] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon and H. Jeong, “Analysis of topological characteristics of huge online social networking services,” in *Proceedings of the 16th international conference on World Wide Web*, New York, 2007.

[43] C. Cheung, Z. W. Y. Lee and T. K. H. Chan, “Self-disclosure in social networking sites: The role of perceived cost, perceived benefits and social influence,” *Internet Research,* pp. 279-299, 2015.

[44] T. Dienlin and S. Trepte, “Is the privacy paradox a relic of the past? An in-depth analysis of privacy attitudes and privacy behaviors,” *European Journal of Social Psychology,* pp. 285-297, 2014.

[45] P. a. Norberg and D. R. Horne, “The Privacy Paradox: Personal Information Disclosure Intentions versus Behaviors,” *Journal of Consumer Affairs,* pp. 100-126, 2007.

[46] G. Govindarajan and N. Ravindar, “Freedom of expression on social media: myth or reality,” *Global Media Journal ,* p. 2249 – 5835 , 2016.

[47] K. Lin and H. Lu, “Predicting mobile social network acceptance based on mobile value and social influence,” *Internet Research,* pp. 1066-2243, 2015.

[48] C.-W. Chang and J. Heo, “Visiting theories that predict college students’ self-disclosure on Facebook,” *Computers in Human Behavior,* pp. 79-86, 2014.

[49] N. Aharony, “Relationships among attachment theory, social capital perspective, personality characteristics, and Facebook self-disclosure,” *Aslib Journal of Information Management,* pp. 2050-3806, 2016.

[50] M. Taddicken, “The privacy paradox in the social web,” *Mediated communication,* 2014.

[51] P. Wisniewski, A. N. Islam, H. R. Lipford and D. C. Wilson, Framing and Measuring Multi-Dimensional Interpersonal Privacy Preferences of Social Networking Site.., Communications of the Association for Information Systems, 2016.

[52] K. Clark, M.-T. Luong, Q. V. Le and C. D. Manning, “ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators,” in *International Conference on Learning Representations*, 2020.

[53] X. Ma, J. T. Hancock, K. L. Mingjie and M. Naaman, “Self-Disclosure and Perceived Trustworthiness of Airbnb Host Profiles,” in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, New York, 2017.

[54] A. Treuille, T. Teixeira and A. Kelly, Accessed on 25-05-2021. [Online]. Available: https://streamlit.io/about.

[55] E. Liddy, Natural Language Processing, New York: In Encyclopedia of Library and Information Science, 2001.

[56] A. Chopra, A. Prashar and C. Sain, “Natural Language Processing,” *International Journal of Technology Enhancements and Emerging Engineering Research,* pp. 131-134, 2013.

[57] P. M. Nadkarni, L. Ohno-Machado and W. W. Chapman, “Natural language processing: An introduction,” *Journal of the American Medical Informatics Association,* p. 544–551, 2011.

[58] M. Haspelmath and A. Sims, Understanding Morphology, London: Routledge, 2010.

[59] C. Gussenhoven and H. Jacobs, Understanding Phonology, London: Routledge, 2017.

[60] A. R. Hippisley, Handbook of Natural Language Processing, Linguistics Faculty Publication, 2010.

[61] D. Sportiche, H. Koopman and E. Stabler, An Introduction to Syntactic Analysis and Theory, UK: John Wiley & Sons, 2013.

[62] C. Goddard, Semantic Analysis: A Practical Introduction, New York: Oxford University Press, 2011.

[63] B. Johnstone, Discourse Analysis, John Eilry & Sons, 2018.

[64] G.Duffy, Pragmatic Analysis, London: Palgrave Macmillan, 2008.

[65] E. D. Liddy, Natural Language Processing, New York: In Encyclopedia of Library and Information Science,, 2001.

[66] W. Khan, A. Daud, J. Nasir and T. Amjad, "A survey on the state-of-the-art machine learning models in the context of NLP," *Kuwait journal of Science,* pp. 95-113, 2016.

[67] A. Kilgarriff, "Thesauruses for natural language processing," in *International Conference on Natural Language Processing and Knowledge Engineering*, Beijing,China, 2003.

[68] P. M. Nadkarni, L. O. Machado and W. W. Chapman, "Natural language processing: an introduction," *Journal of the American Medical Informatics Association,* p. 544–551, 2011.

[69] S. Sarawagi, Information Extraction, USA: now Publishers Inc., 2008.

[70] V. T. Chakaravarthy, H. Gupta, P. Roy and M. K. Mohania, "Efficient Techniques for Document Sanitization," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, California, 2008.

[71] D. Sánchez, M. Batet and A. Viejo, "Detecting Sensitive Information from Textual Documents: An Information-Theoretic Approach," in *Conference: Modelling Decisions in Artificial Intelligence*, 2012.

[72] J. Tang, M. Hong, D. L. Zhang and J. Li, Information Extraction: Methodologies and Applications, China, 2008.

[73] D. Roth, "Learning to Resolve Natural Language Ambiguities:A Unified Approach," in *AAAI*, 1998.

[74] Y. A. Solangi, Z. A. Solangi, S. Aarain, A. Abro, G. A. Mallah and A. Shah, "Review on Natural Language Processing (NLP) and Its Toolkits for Opinion Mining and Sentiment Analysis," in *IEEE 5th International Conference on Engineering Technologies and Applied Sciences*, 2018.

[75] Guru99, "tokenize-words-sentences," accessed 05-02-2021. [Online]. Available: https://www.guru99.com/tokenize-words-sentences-nltk.html.

[76] V. a. L.-Y. E. Balakrishnan, "Stemming and lemmatization: A comparison of retrieval performances," in *In: Proceedings of SCEI Seoul Conferences*, Seoul, 2014.

[77] Pythonprogramming.net, "https://pythonprogramming.net/," Accessed 06-02-2021. [Online]. Available: https://pythonprogramming.net/stemming-nltk-tutorial/.

[78] C. Anish, "forming-a-feature-vector-for-natural-language-processing," 27 Jan 2021. [Online]. Available: https://medium.com/spidernitt/forming-a-feature-vector-for-natural-language-processing-b49486e1c637.

[79] B. Mohit, "Named Entity Recognition," in *Natural Language Processing of Semitic Languages*, Springer, Berlin, Heidelberg, 2014, pp. 221-245.

[80] Y. Si, W. Zhou and J. Gai, “Research and Implementation of Data Extraction Method Based on NLP,” in *14th International Conference on Anti-counterfeiting, Security, and Identification*, Xiamen, China, 2020.

[81] T. S, B. J and T. Geetha, “Semi-Supervised Bootstrapping Approach for Named Entity Recognition,” *International Journal on Natural Language Computing,* pp. 01-14, 2015.

[82] S. K. Sienčnik, “Adapting word2vec to Named Entity Recognition,” in *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, Vilnius, Lithuania, 2015.

[83] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, “Neural Architectures for Named Entity Recognition,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, 2016.

[84] J. Liu, J. Yao and G. Wu, “Sentiment classification using information extraction technique,” in *Proceedings of the 6th international conference on Advances in Intelligent Data Analysis*, 2005.

[85] R. Feldman, “Techniques and applications for sentiment analysis,” *Communications of the ACM,* p. 82–89, 2013.

[86] Q. Zhang, M. Chen and L. Liu, “A Review on Entity Relation Extraction,” in *Second International Conference on Mechanical, Control and Computer Engineering*, Harbin, China, 2017.

[87] N. T. Huu and R. Grishman, “Employing Word Representations and Regularization for Domain Adaptation of Relation Extraction,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, Maryland, Association for Computational Linguistics, 2014, pp. 68-74.

[88] S. Kaur and R. Agrawal, “A Detailed Analysis of Core NLP for Information Extraction,” *International Journal of Machine Learning and Networked Collaborative Engineering,,* pp. 33-47, 2018.

[89] J. Giorgi, X. Wang, N. Sahar, W. Y. Shin, G. D. Bader and B. Wang, “End-to-end Named Entity Recognition and Relation Extraction using Pre-trained Language Models,” *Computation and Language,* pp. 1-12, 2019.

[90] M. Mintz, S. Bills, R. Snow and D. Jurafsky, “Distant supervision for relation extraction without labeled data,” in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Singapore, 2009.

[91] G. Ji, K. Liu, S. He and J. Zhao, “Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[92] C. d. Santos, B. Xiang and B. Zhou, “Classifying Relations by Ranking with Convolutional Neural Networks,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational*

Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 2015.

[93] R. Socher, B. Huval, C. D. Manning and A. Y. Ng, "Semantic Compositionality through Recursive Matrix-Vector Spaces," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* , Jeju Island, Korea, 2012.

[94] M. Miwa and M. Bansal, "End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* , Berlin, Germany, 2016.

[95] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *Computing Research Repository,* p. arXiv:1810.04805, 2018.

[96] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," in *Advances in Neural Information Processing Systems*, Curran Associates,Inc., 2019.

[97] A. K. M. N. Mehdy, C. Kennington and H. Mehrpouyan, "Privacy Disclosures Detection in Natural-Language Text Through Linguistically-Motivated Artificial Neural Networks," in *Security and Privacy in New Computing Environments*, China, Springer, Cham, 2019.

[98] P. Shi and J. Lin, "Simple BERT Models for Relation Extraction and Semantic Role Labeling," *Computation and Language,* p. arXiv:1904.05255, 2019.

[99] P. Umar, A. Squicciarini and S. Rajtmajer, "Detection and Analysis of Self-Disclosure in Online News Commentaries," in *The World Wide Web Conference*, 2019.

[100] S.-M. Kim and E. Hovy, "Extracting opinions, opinion holders, and topics expressed in online news media text," in *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, United States, 2006.

[101] D. J. Houghton and A. N. Joinson, " Linguistic markers of secrets and sensitive self-disclosure in Twitter," in *45th Hawaii International Conference on System Science*, 2012.

[102] A. Ravichander and A. W. Black, "An Empirical Study of Self-Disclosure in Spoken Dialogue Systems," in *Proceedings of the 19th Annual {SIG}dial Meeting on Discourse and Dialogue*, Melbourne, Australia, Association for Computational Linguistics, 2018, pp. 253-263.

[103] A. Ram, R. Prasad, C. Khatri, A. Venkatesh, R. Gabriel, Q. Liu, J. Nunn, B. Hedayatnia, M. Cheng and A. Nagar, "Conversational AI: The Science Behind the Alexa Prize," *Alexa.Prize.Proceedings,* p. arXiv:1801.03604 , 2018.

[104] J. Leskovec, A. Rajaraman and J. Ullman, Mining of Massive Datasets, NewYork: Cambridge University Press, 2011.

[105] K. Doell, "The Word Feel as an Indicator of Enacted Social Support in Personal Relationships," *International Journal of Psychological Studies ,* pp. 107-121, 2013.

[106] J. B. R. J. K. &. B. K. Pennebaker, “The Development and Psychometric Properties of LIWC2015,” *LIWC2015,* pp. 1-25, 2015.

[107] J. Bak, C.-Y. Lin and A. Oh, “Self-disclosure topic model for {T}witter conversations,” in *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, Baltimore, Maryland, Association for Computational Linguistics, 2014, pp. 42-49.

[108] A. Barak and O. Gluck-Ofri, “Degree and Reciprocity of Self-Disclosure in Online Forums,” *CyberPsychology & Behavior,* pp. 407-417, 2007.

[109] D. M. Blei, A. Y. Ng and M. I. Jordan, “Latent Dirichlet Allocation,” *Journal of Machine Learning Research,* pp. 993-1022, 2003.

[110] J. Zhu, A. Ahmed and E. P. Xing, “MedLDA: Maximum Margin Supervised Topic Models,” *Journal of Machine Learning Research,* p. 2237−2278, 2012.

[111] Y. R. Tausczik and J. W. Pennebaker, “The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods,” *Journal of Language and Social Psychology ,* pp. 24-54, 2010.

[112] Y. Jo and A. H. Oh, “Aspect and sentiment unification model for online review analysis,” in *Proceedings of the fourth ACM international conference on Web search and data mining*, New York, 2011.

[113] O. Owoputi, B. Connor, C. Dyer, K. Gimpel, N. Schneider and N. A. Smith, “Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters,” in *Proceedings of the 2013 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, Association for Computational Linguistics, 2013, pp. 380-390.

[114] M. M. Tadesse, H. Lin, B. Xu and L. Yang, “Detection of Depression-Related Posts in Reddit Social Media Forum,” *IEEE Access,* pp. 44883-44893, 2019.

[115] I. Pirina and Ç. Çöltekin, “Identifying Depression on Reddit: The Effect of Training Data,” *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task,* p. 9–12, 2018.

[116] D. Preoţiuc-Pietro, J. Eichstaedt, G. Park, M. Sap, L. Smith, V. Tobolsky, H. A. Schwartz and L. Ungar, “The role of personality, age, and gender in tweeting about mental illness,” in *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, Denver, Colorado, 2015.

[117] P. Resnik, A. Garron and R. Resnik, “Using Topic Modeling to Improve Prediction of Neuroticism and Depression in College Students,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, 2013.

[118] J. W. Pennebaker, R. J. Booth, R. L. Boyd and M. E. Francis, “Linguistic Inquiry and Word Count,” *LIWC2015,* 2015.

[119] C. Jin, H. Kaur, A. Khatun and S. Uppalapati., “Intelligent Computing,” in *Detecting Traces of Bullying in Twitter Posts Using Machine Learning*, Springer, Cham, 2019, pp. 796-803.

[120] C. R. Akiti, S. Rajtmajer and A. Squicciarini., “Contextual representation of self-disclosure and supportiveness in short text,” *CEUR Workshop Proceedings,* pp. 179-206, 2020.

[121] C. R. Akiti, A. Squicciarini and S. Rajtmajer, “A Semantics-based Approach to Disclosure Classification in User-Generated Online Content,” in *Findings of the Association for Computational Linguistics*, 2020.

[122] C. Baker, M. Ellsworth and K. Erk, “Frame Semantic Structure Extraction,” in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic, 2007.

[123] T. Dadu, K. Pant and R. Mamidi, “BERT-based Ensembles for Modeling Disclosure,” in *AffCon@AAAI 2020*, New York, 2020.

[124] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” in *ICLR*, 2019.

[125] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma and R. Soricut, “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations,” in *International Conference on Learning Representations*, 2019.

[126] V. Sanh, L. Debut, J. Chaumond and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” *ArXiv,* p. arXiv:1910.01108, 2019.

[127] I. Airbnb, “http://insideairbnb.com,” Accessed on 01-03-2021. [Online]. Available: http://insideairbnb.com/get-the-data.html.

[128] X. Ma, J. T. Hancock, K. Lim Mingjie and Naaman, “Self-Disclosure and Perceived Trustworthiness of Airbnb Host Profiles,” in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, New York, 2017.

[129] A. Lampinen and C. Cheshire, “Hosting via Airbnb: Motivations and Financial Assurances in Monetized Network Hospitality,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, New York, 2016.

[130] A. 2021, “/about/about-us,” Accessed on 28-04-2021. [Online]. Available: https://www.airbnb.com/about/about-us.

[131] S. Ruder, M. Peters, S. Swayamdipta and T. Wolf, “Transfer Learning in Natural Language Processing Tutorial,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, Minneapolis, Minnesota, 2019.

[132] W. Kouw and M. Loog, “An introduction to domain adaptation and transfer learning,” *ArXiv preprint,* p. arXiv:1812.11806, 2019.

[133] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing and R. Feris, “SpotTune: Transfer Learning Through Adaptive Fine-Tuning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[134] B. Werness, R. Hu, S. Zhang and Y. Tay, “Dive into Deep Learning,” in *Fine-Tuning*, accessed on 05-05-2021.

[135] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi and N. Smith, “Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping,” *ArXiv,* p. arXiv:2002.06305, 2020.

[136] A. C. a. K. L. Iz Beltagy, “Scibert: Pretrained contextualized embeddings for scientific text.,” *EMNLP 2019,* no. arXiv:1903.10676, 2019.

[137] D.-C. C. M. D. Andrei Paraschiv, “UPB at SemEval-2020 Task 11:Propaganda Detection with Domain-Specific Trained BERT,” in *SEMEVAL* , 2020.

[138] X. H. J. G. Amittai Axelrod, “Domain Adaptation via Pseudo In-Domain Data Selection,” in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., 2011.

[139] T. Rajapakse, Accessed on 17-05-2021. [Online]. Available: https://simpletransformers.ai/about/.

[140] S. Lukasik, “Why the Arpanet Was Built,” *IEEE Annals of the History of Computing,* pp. 4-21, 2011.

[141] Technopedia, “information-privacy,” accessed 29/01/2021 2021. [Online]. Available: https://www.techopedia.com/definition/10380/information-privacy.

[142] Bitext, “what-is-the-difference-between-stemming-and-lemmatization,” 05 Apr 2021. [Online]. Available: https://blog.bitext.com/what-is-the-difference-between-stemming-and-lemmatization/.

[143] R. Mitkov, “Part-of-speech tagging,” in *The Oxford Handbook of Computational Linguistics*, 2003, pp. 220-227.

[144] S. Bird, E. Klein and E. Loper, 4 Sep 2019. [Online]. Available: https://www.nltk.org/book/ch05.html.

[145] J. Nivre and S. Kübler, “Dependency Parsing,” *Synthesis Lectures on Human Language Technologies,* pp. 1-5, 2009.

[146] J. D. Choi and A. McCallum, “Transition-based Dependency Parsing with Selectional Branching,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Bulgaria, 2013.

[147] K. Jaidka, S. C. Guntuku and L. H. Ungar, “Facebook versus Twitter: Differences in Self-Disclosure and Trait Prediction,” in *International AAAI Conference on Web and Social Media*, 2018.

[148] M. H. Davis, “Measuring individual differences in empathy: Evidence for a multidimensional approach,” *Journal of Personality and Social Psychology,* pp. 113-126, 1983.

[149] S. Cohen, R. C. Kessler and L. U. Gordon, Measuring stress: A guide for health and social scientists., Oxford University Press., 1997.

[150] A. Z.Broder, S. C.Glassman, M. S.Manasse and G. Zweig, “Syntactic clustering of the Web,” *Computer Networks and ISDN Systems,* pp. 1157-1166, 1997.

[151] J. W. Pennebaker, M. E. Francis and R. J. Booth, “Linguistic inquiry and word count,” *LIWC,* 2001.

[152] H. A. Schwartz, S. Giorgi, M. Sap, P. Crutchley, L. Ungar and J. Eichstaedt, “DLATK: Differential Language Analysis ToolKit,” in *Conference: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Copenhagen, Denmark, 2017.

[153] A. Kirzinger, L. Hamel, C. Munana, A. Kearney and M. Brodie, “https://www.kff.org,” 24 Apr 2020. [Online]. Available: https://www.kff.org/coronavirus-covid-19/issue-brief/kff-health-tracking-poll-late-april-2020/.

[154] A. Squicciarini, S. Rajtmajer, P. Umar and T. Blose, “A Tipping Point? Heightened self-disclosure during the Coronavirus pandemic,” in *2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI)*, Atlanta, GA, USA, 2020.

[155] M. Jurgens and I. Helsloot, “The effect of social media on the dynamics of (self) resilience during disasters: A literature review.,” *Journal of Contingencies and Crisis Management,* p. 79–88, 2017.

[156] T. Blose, P. Umar, A. Squicciarini and S. Rajtmajer, “Privacy in Crisis: A study of self-disclosure during the Coronavirus pandemic,” *Social and Information Networks,* p. arXiv:2004.09717, 2020.

[157] H. Zhong, D. J. Miller and A. Squicciarini., “Flexible Inference for Cyberbully Incident Detection,” in *European Conference,ECML*, Dublin, Ireland, 2019.

[158] D. Gildea and D. Jurafsky, “Automatic Labeling of Semantic Roles,” in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong, 2000.

[159] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” in *Neural Computation* , 1997, p. 1735–1780.

Appendix A

Loss functions of the Generator G and Discriminator D of the ELECTRA

The authors of the ELECTRA [52] ,the loss functions of Generator G and Discriminator D can be represented as $L_G(x, \theta_G)$ (10) and $L_D(x, \theta_D)$ (11) respectively, where x^{masked} is the masked token and $x^{corrupt}$ is the corrupted token. They also proposed to replace masked tokens with generator samples.

$$L_G(x, \theta_G) = \mathbb{E}(\sum_{i \in m} -logP_G(x_i|x^{masked})) \tag{10}$$

$$L_D(x, \theta_D) = \mathbb{E}(\sum_{t=1}^{n} -\mathbb{1}(x_t^{corrupt} = x_t)logD(x^{corrupt}, t) - \mathbb{1}(x_t^{corrupt} \neq x_t)log(1 - D(x^{corrupt}, t)) \tag{11}$$

Appendix B

Pre-Training and Fine-Tuning parameters

B.1. Pre-Training Parameters (SD_ELECTRA_V1 and SD_ELECTRA_V2)

Table 17. Pre-training parameters used for the proposed model.

Hyperparameter	***Value***
Debug	FALSE
disallow_correct	FALSE
disc_weight	50
do_eval	FALSE
do_lower_case	TRUE
do_train	TRUE
electra_objective	TRUE
electric_objective	FALSE
embedding_size	128
eval_batch_size	128
gcp_project	None
gen_weight	1
generator_hidden_size	0.25
generator_layers	1
iterations_per_loop	200
keep_checkpoint_max	5
learning_rate	0.0005
lr_decay_power	1
mask_prob	0.15
max_predictions_per_seq	19
max_seq_length	128
model_hparam_overrides	{}
model_name	electra_SD_V4
model_size	small
num_eval_steps	100
num_tpu_cores	1
num_train_steps	1000000
num_warmup_steps	10000
save_checkpoints_steps	1000

Temperature	1
tpu_job_name	None
tpu_name	None
tpu_zone	None
train_batch_size	128
two_tower_generator	FALSE
uniform_generator	FALSE
untied_generator	TRUE
untied_generator_embeddings	FALSE
use_tpu	FALSE
vocab_size	30522
weight_decay_rate	0.01

B.2. Fine-tuning Parameters (SD_ELECTRA_V1 and SD_ELECTRA_V2)

Table 18. Fine-training parameters used for the proposed model.

Hyper parameter	***Value***
Batch size	32
Sequence length	128
Learning rate	0.0001
Number of training epochs	5
Weight decay rate	0.01

Appendix C

User Interface

The following are the other examples of NLP functionality included in the user interface platform.

C.1. Sentiment Analysis

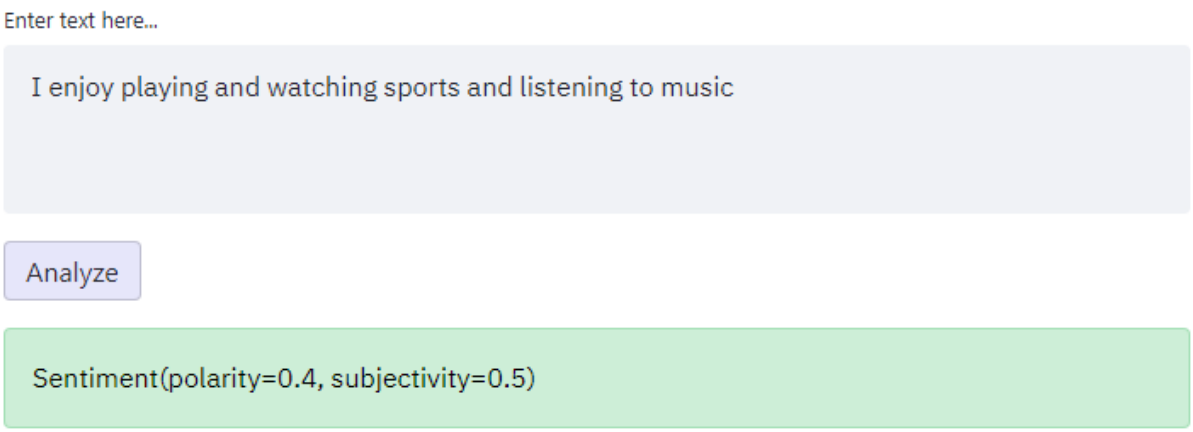

Fig. 21. Sentiment Analysis screen in the proposed user interface.

C.2. Tokenization

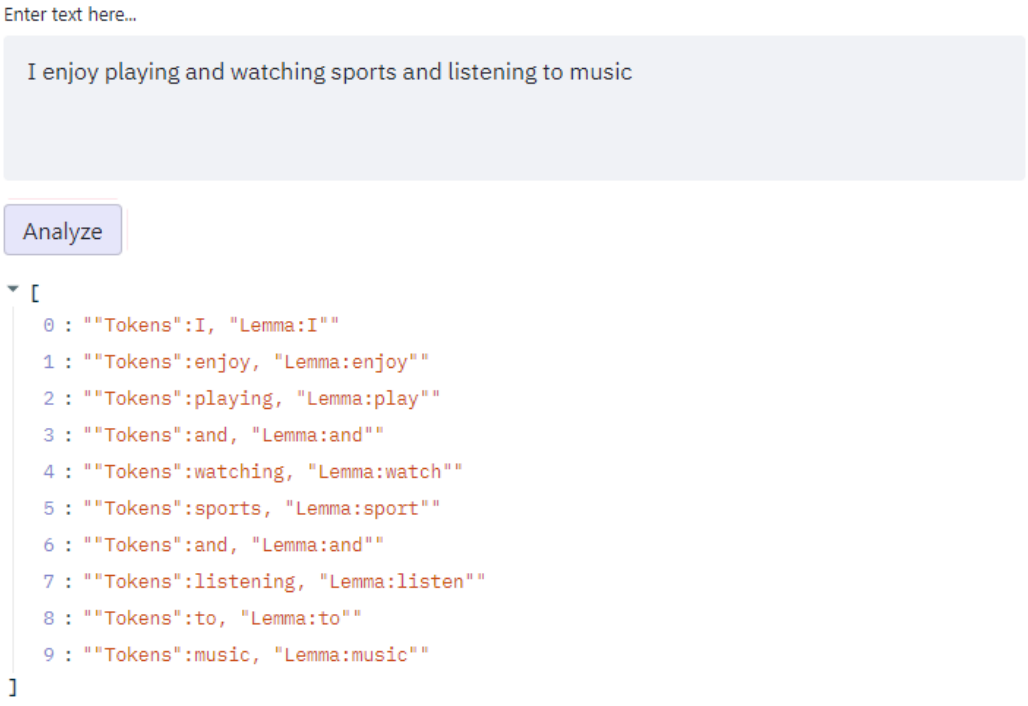

Fig. 22. Tokenization screen in the proposed user interface.

C.3. Named Entity Recognition

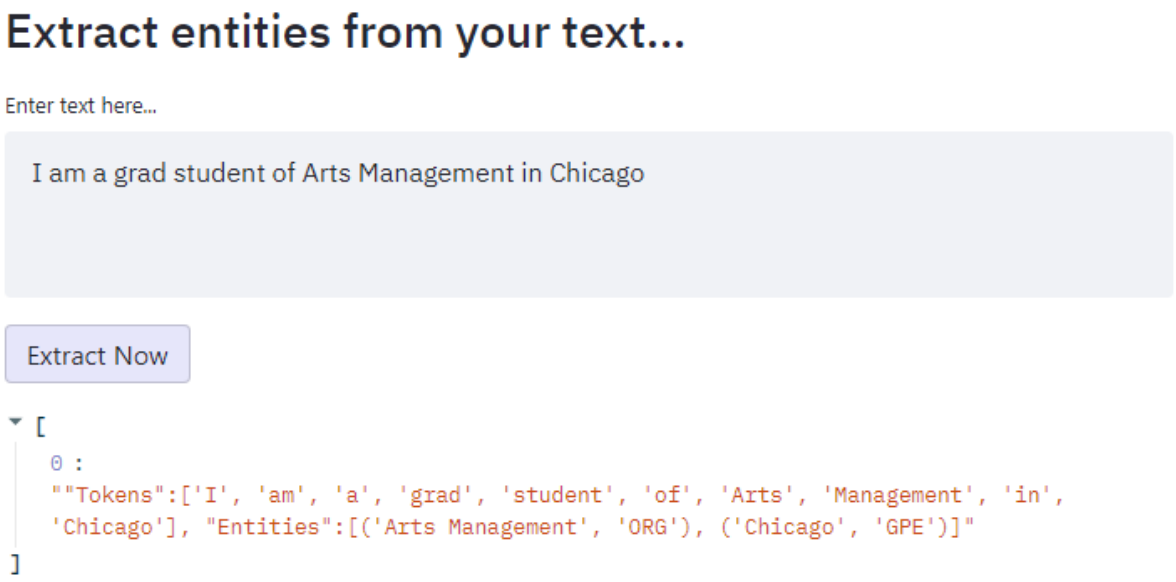

Fig. 23. NER in the proposed user interface.