Proceedings of the Seventeenth International AAAI Conference on Web and Social Media (ICWSM 2023)

Catch Me If You Can: Deceiving Stance Detection and Geotagging Models to Protect Privacy of Individuals on Twitter

Dilara Dogan[1], **Bahadir Altun**[1], **Muhammed Said Zengin**[1], **Mucahid Kutlu**[1], and **Tamer Elsayed**[2]
[1] TOBB University of Economics and Technology, Ankara, Türkiye
[2] Qatar University, Doha, Qatar
{dilara.dogan, i.altun, muhammedsaid.zengin, m.kutlu}@etu.edu.tr, telsayed@qu.edu.qa

Abstract

The recent advances in natural language processing have yielded many exciting developments in text analysis and language understanding models; however, these models can also be used to track people, bringing severe privacy concerns. In this work, we investigate what individuals can do to avoid being detected by those models while using social media platforms. We ground our investigation in two exposure-risky tasks, stance detection and geotagging. We explore a variety of simple techniques for modifying text, such as inserting typos in salient words, paraphrasing, and adding dummy social media posts. Our experiments show that the performance of BERT-based models fine-tuned for stance detection decreases significantly due to typos, but it is not affected by paraphrasing. Moreover, we find that typos have minimal impact on state-of-the-art geotagging models due to their increased reliance on social networks; however, we show that users can deceive those models by interacting with different users, reducing their performance by almost 50%.

Introduction

Recent developments in artificial intelligence (AI), especially in natural language processing (NLP), bring many opportunities to deploy AI models in real life, such as human-like speaking personal assistants and accurate machine translation models. The massive amount of data available on social media platforms enables the development of increasingly-accurate models that predict lots of "implicit" information about users, e.g., location (Rahimi, Cohn, and Baldwin 2018), stance on various issues (Küçük and Can 2020), age, gender (Morgan-Lopez et al. 2017), mental health (Sekulic and Strube 2019), and ethnicity (Preoţiuc-Pietro and Ungar 2018).

Several of those developed models can be used for a good cause. For instance, we can utilize stance detection models for fact-checking (Baly, Mohtarami, and Glass 2018) and social polarization analysis (Rashed et al. 2020). Similarly, geotagging models can be used to identify affected areas during natural disasters (Ghahremanlou, Sherchan, and Thom 2015), and reduce bias in data collected for public opinion prediction (Dwi Prasetyo and Hauff 2015).

Despite their beneficial use cases, many people have a legitimate privacy concern, as those platforms have access to too much information about people. While we also share similar concerns about using private information of people for commercial activities such as targeted ads, we believe that there is a more severe problem: *the data is accessible by anyone, not only by the social media companies*.

Understandably, many people opt for having public profiles instead of protected ones. Therefore, any person can crawl massive amounts of data these platforms provide and develop AI models for various reasons, including unethical ones. For instance, automatic stance detection methods might be problematic for people living in countries with limited freedom of speech or in a highly polarized society. Similarly, geotagging methods can be used to identify and expose where a particular individual lives. Therefore, these models can be easily weaponized if used by the wrong people. Furthermore, recent developments in NLP, e.g., transformer models like BERT (Devlin et al. 2019) and GPT-3 (Brown et al. 2020), enabled the development of effective NLP models with a minimal effort, requiring only a few lines of code and a labeled dataset. Therefore, the data can be used even by people with limited NLP knowledge and coding skills.

Due to privacy concerns, many social media users tend to hide their identity behind nicknames or their location behind anonymous terms, e.g., "earth". However, too much information is still subject to be revealed (hence exposing their owners) due to the success of modern models in leveraging unintentionally-available or implicit clues in their social media traces. *We believe that individuals should be able to opt-out of being tracked by those AI models if they so desire.* However, exploiting the available data over social media platforms leaves individuals vulnerable.

In this work, we investigate how users can protect their privacy against AI models by themselves while using social media platforms. We focus on two "exposure-risky" tasks, stance detection and geotagging, because individuals' physical location and stance on various issues can potentially be used against them as mentioned above. We identify state-of-the-art models for each task and explore how to fool them using simple text manipulation techniques (such as inserting strategic typos, paraphrasing, and adding extra text) to provide a list of recommendations to reduce the likelihood of being detected by such AI models.

In particular, we address the following research questions:

RQ1: *What are the most effective text manipulation methods which fool state-of-the-art stance detection models without changing the semantics of the text?* We found that BERT-based models are vulnerable to typos; their performance decays significantly for stance detection when typos are introduced in social posts by additional spaces or changing/shuffling their letters. However, we were not able to fool BERT-based models by paraphrasing.

RQ2: *Which method to fool models has the least side effects regarding readability and semantics?* In our analysis, we found that shuffling characters can cause unreadable tweets, and carelessly removing hashtags might change the semantics of tweets. Furthermore, inserting space characters rarely changes semantics but makes tweets unreadable. However, the other methods we apply automatically do not cause any semantic change or unreadable tweets.

RQ3: *What are the most effective methods to fool state-of-the-art geotagging models?* Our experiments show that state-of-the-art geotagging models are slightly affected by adding typos and mentioning various city names due to the models' reliance on social networks in addition to the posts' content. However, we found that interacting with various users at different locations is an effective strategy for deceiving geotagging models.

Our contribution in this work is three-fold. (1) While there exist studies exploring how to fool AI models, we address the problem from a different perspective. We investigate what a random social media user, who might have no idea about how these AI models work, can do to protect his/her privacy. (2) We investigate the impact of 15 different methods to fool stance detection and geotagging models, providing recommendations accordingly. (3) We release our code and data to support the reproducibility of experiments[1].

Related Work

A number of researchers investigated adversarial attacks and defense mechanisms for various tasks (Ren et al. 2020). Our work can also be considered an investigation of adversarial attacks against NLP models. However, we have a thoroughly opposite perspective, such that social media users are not "attackers" because we believe that they are potential victims and explore how they can "defend" their privacy. Ignoring different perspectives on this issue, we now compare our study against prior work on adversarial attacks, especially attacks for NLP models.

Chen et al. (2020) investigate *backdoor attacks* in which training data of models are manipulated such that targeted NLP models fail when specific triggers (e.g., words) are used, but work as usual with clean data. Yang et al. (2021) show that changing only a single word embedding vector is an effective method to hack sentiment analysis and sentence-pair classification models without causing any deterioration in the results of the existing clean samples. Dai, Chen, and Li (2019) demonstrate that a backdoor attack by inserting trigger sentences into training data of an LSTM-based sentiment analysis model is highly effective. Kurita, Michel, and Neubig (2020) compare various backdoor attacks mentioned by Gu, Dolan-Gavitt, and Garg (2017) for sentiment analysis, toxicity detection, and spam detection tasks. They show that attack successes change for each NLP task. Sun (2020) introduces *natural backdoor attacks* that are hard to be noticed by humans and grammar correction systems. Sun shows that natural backdoor attacks are highly successful for text classification problems. In our work, we assume a black-box model such that we do not have access to training data, and we aim to fool already trained models. However, backdoor attacks assume that they can affect the training phase of AI models.

A number of researchers also explored vulnerabilities of NLP models in a black-box setting by changing the test data. The methods prior work investigated can be grouped into three categories: 1) character-level changes in which words are written with various forms of spelling errors, 2) word-level changes in which words are replaced, removed, or added, and 3) sentence-level changes in which new sentences or phrases are added or existing ones removed or paraphrased. **Table 1** shows these adversarial attack methods investigated by prior work.

Among the methods used by prior work, we also use middle character shuffle (Belinkov and Bisk 2018) and inserting a space character (Sun et al. 2020; Li et al. 2019) methods. In addition, some of these methods can be considered similar to ours. For instance, Dai, Chen, and Li (2019); Li et al. (2019); Morris et al. (2020), and Liang et al. (2018) replace some letters with visually similar ones, but we use a different replacement scheme.

Liang et al. (2018) detect the most frequent phrases in the respective training dataset and remove/add them to fool NLP models, assuming they have access to the training data. However, we remove/add hashtags without any analysis of the training data. Jin et al. (2020), Li et al. (2019), and Ebrahimi et al. (2018) replace words with semantically similar ones using word embeddings. We replace words with their synonyms or use uncommon names of famous people.

Jia and Liang (2017) add manually selected sentences to create adversarial examples for reading comprehension systems. Our mentioning of a particular city method can be considered a similar approach but used for fooling geotagging models.

Niu and Bansal (2018) paraphrase sentences using Pointer-Generator Networks. Liang et al. (2018) paraphrase phrases using the approach of Barzilay and McKeown (2001). We also use various paraphrasing methods, such as using idioms. However, our paraphrasing methods focus on fooling methods instead of just expressing statements in a different way.

Schiller, Daxenberger, and Gurevych (2021) investigate the robustness of stance detection models using three different adversarial attacks which are 1) adding the tautology "and false is not true" at the beginning of each sentence, 2) introducing spelling errors by character swaps and substitutions, and 3) paraphrasing by back-translation. They report that transformer-based models have serious robustness problems due to the overfitting of biases of training data. They specifically focus on the stance detection task, but our work

[1] https://github.com/dilaradogan/ICWSM-2023

	Method	Example	Study
Character Level	Insertion	apple → applee	(Sun et al. 2020)
	Deletion	school → schol	(Sun et al. 2020; Li et al. 2019)
	Character swap	hello → hlelo	(Sun et al. 2020; Li et al. 2019)
	Using different words pronounced same/similar	egg → agg	(Sun et al. 2020)
	Replacing characters with the nearby ones in a keyboard	shy → why	(Schiller, Daxenberger, and Gurevych 2021; Sun et al. 2020; Li et al. 2019; Belinkov and Bisk 2018)
	Replacing letters w/ visually similar characters	foolish → fo0lish	(Dai, Chen, and Li 2019; Li et al. 2019; Morris et al. 2020; Liang et al. 2018)
	Inserting a space within a word	school → sc hool	(Sun et al. 2020; Li et al. 2019)
	Mistyping any character	talk → taln	(Sun et al. 2020; Ebrahimi et al. 2018)
	Common misspelling	film → flim	(Liang et al. 2018)
	Middle Character shuffle	noise→ nisoe	(Belinkov and Bisk 2018)
Word Level	Replace words with semantically similar ones	awful → terribly	(Jin et al. 2020; Li et al. 2019; Niu and Bansal 2018; Ebrahimi et al. 2018)
	Swap adjacent words	"I don't want you to go" → "I don't want to you go"	(Niu and Bansal 2018)
	Remove Stopwords	Ben ate ~~the~~ carrot	(Niu and Bansal 2018)
	Insert a word	The Uganda Securities Exchange (USE) is the **historic** principal stock exchange of Uganda.	(Liang et al. 2018)
	Remove a word	The Old Harbor Reservation Parkways are three ~~historic~~ roads in the Old Harbor area of Boston.	(Liang et al. 2018)
Sentence Level	Add a sentence	The Old Harbor Reservation Parkways are three historic roads in the Old Harbor area of Boston. **Some exhibitions of Navy aircrafts were held here.**	(Jia and Liang 2017; Liang et al. 2018)
	Paraphrase a sentence	"How old are you" → "What's your age"	(Niu and Bansal 2018)
	Paraphrasing a phrase	the actual composer is ~~different from~~ **not** the artist	(Liang et al. 2018)
	Removing a phrase	~~promotion of world security~~, improvement of economic conditions	(Liang et al. 2018)
	Grammar Errors	"He ~~doesn't~~ **don't** like cakes"	(Niu and Bansal 2018)

Table 1: Adversarial attacks used in prior work. The words in boldface represent the added words.

covers both stance detection and geo-tagging tasks. While their methods to fool the models can be considered similar to some of our methods, we use additional fooling methods such as using idioms.

To our knowledge, some of our methods have not been used by prior work, including interaction with other users, removing spaces, and using idioms. In addition, our targeted tasks are different from prior work. In particular, prior work investigated adversary attacks for various NLP tasks, including sentiment analysis (Jin et al. 2020; Dai, Chen, and Li 2019; Li et al. 2019), question answering (Jia and Liang 2017), dialogue generation (Niu and Bansal 2018), machine translation (Belinkov and Bisk 2018), toxicity detection (Kurita, Michel, and Neubig 2020), textual entailment (Jin et al. 2020), and spam detection (Kurita, Michel, and Neubig 2020). However, to our knowledge, this is the first study focusing on the geotagging task.

Our study diverges from prior work in terms of its goal. Specifically, prior work focused on increasing the robustness of NLP models by exploring their vulnerabilities (Liang et al. 2018), generating adversarial examples for training (Li et al. 2019; Jin et al. 2020), and enhancing models' architectures for noisy data (Muller, Sagot, and Seddah 2019; Niu and Bansal 2018). By contrast, our research seeks to identify techniques that enable individuals to conceal their personal data from AI models on social media platforms.

Our work is also related to studies in ethics in NLP. Mieskes (2017) investigated ethical issues in data collection and sharing in NLP. She suggests anonymizing users instead of directly using their sensitive data. However, Feyisetan, Ghanavati, and Thaine (2020) state that using anonymized data does not actually solve the privacy problem. In addition, Silva et al. (2020) show that NLP tools such as NLTK[2], Stanford CoreNLP[3], and SpaCy[4] can tag personal information on anonymized data. In our work, we focus on how to protect privacy while using social media platforms

Targeted Tasks and Models

We focus on two exposure-risky tasks, namely stance detection, and geotagging. Here we define the tasks and describe

[2]https://www.nltk.org/

[3]https://stanfordnlp.github.io/CoreNLP/

[4]https://spacy.io/

the state-of-the-art work we used in our study.

Stance Detection

Stance detection is the task of determining whether the author of a given text is favoring, against, or neutral towards a target or proposition (Mohammad et al. 2016). Ghosh et al. (2019) compare various stance detection models and report that fine-tuned BERT model yields the best prediction performance on the widely used SemEval 2016 Task 6A dataset (Mohammad et al. 2016). Hence, we also utilize the fine-tuned BERT model as one of our models. However, BERT is pre-trained on clean text with a few typos, while some of our methods insert typographical errors deliberately. As such, as an additional model, we use a fine-tuned Twitter-RoBERTa,[5] which is pre-trained with 58M tweets, making it potentially more robust to typographical errors.

Geotagging

Research on geotagging can be broadly categorized into two groups: tweet-level geotagging (Yavuz and Abul 2016) and user-level geotagging (Rahimi, Cohn, and Baldwin 2018). In this study, our emphasis is on the latter. We use two state-of-the-art geotagging studies in our work. The first is based on Graph Convolutional Networks (**GCN**) (Rahimi, Cohn, and Baldwin 2018), which is a hybrid model that uses both textual content and user network information to improve prediction accuracy. The textual content is represented as bag-of-words and graphs are created from user mentions. The second approach, **MLP-TXT+NET** (Rahimi, Cohn, and Baldwin 2018), utilizes a multilayer perceptron in which each timeline is represented by concatenating the bag-of-words vector of tweets and user networks.

Problem Definition

The primary objective of our research is to explore strategies that enable social media users to share their posts without risking the exposure of their personal information by AI models. To accomplish this, we leverage techniques that modify the content of the posts or profiles while preserving the semantics, or at least maintaining similar semantics. Formally, let t be a tweet posted by a user u, f be our text or profile manipulation method, m_s be a stance detection model, and m_g be a geotagging model. An ideal method should have the following properties:

- **Maintaining Semantics:** Semantics of $f(t)$ should be similar to the semantics of t. Similarly, $f(u)$ should have the same or similar tweets to u.
- **Minimal Side Effect:** To make methods applicable in real-life, the side effects of using them should be minimal. For instance, while the inclusion of intentional typos may effectively deceive AI models, it may come at the cost of reduced readability and a less professional appearance of tweets.
- **Deceiving AI Models:** The modified tweets or user profiles should be able to deceive AI models, yielding inaccurate predictions. In particular, if $m_s(t)$ yields a correct stance, then $m_s(f(t))$ should yield a different one. In addition, the accuracy of geotagging models is usually defined by the distance between predicted and actual locations; therefore, $m_g(f(u))$ should be farther from the actual location than $m_g(u)$.

[5]https://huggingface.co/cardiffnlp/twitter-roberta-base

Methods to Deceive the Models

This section presents the methods we investigate to deceive stance detection and geotagging models. These methods can be broadly classified into three categories including 1) inserting typographical errors, 2) paraphrasing tweets, and 3) adding additional tweets to user profiles. As we focus on stance detection of tweets, we apply only inserting typographical errors and paraphrasing tweets for the stance detection task. However, geotagging models predict locations based on user profiles. Therefore, in addition to methods altering the content of tweets, we also explore the impact of methods that add additional tweets to user profiles.

To identify effective methods for changing the content of tweets, we conducted a manual text modification study to explore how we can post messages while concealing our personal information. In particular, we first fine-tuned the BERT model for stance detection using SemEval Task 6 dataset (Mohammad et al. 2016). Subsequently, we randomly sampled tweets that the fine-tuned BERT model could predict their stance correctly. Then, we manually modified the contents of the tweets to deceive the model and identify effective methods. We heuristically developed methods that add additional tweets to user profiles.

Now, we explain the methods used in this work. We consider that tweets are modified manually for now. In the following section, we discuss how we can automate some of these methods. To facilitate a better understanding of our methods, we provide a sample tweet for each of the methods in **Table 2**.

Intentional Typographical Errors

BERT models generate embeddings for subwords based on their vocabulary. If it encounters an out-of-vocabulary word, it slices the word into subwords and create an embedding for each of them. For instance, writing the word "against" as "aganist" would cause BERT to produce embeddings for "ag", "-ani", and "-st" tokens instead a single embedding for the word "against". By intentionally inserting typographical errors, our goal is to increase the number of out-of-vocabulary words and prompt BERT and RoBERTa models to generate embeddings for unrelated subwords.

Moreover, the geotagging models we use in our study utilize bag-of-words representation for tweet contents. Therefore, by inserting typos, we can reduce the number of words represented in bag-of-words vectors because the words with typos are less likely to appear in the training datasets. Now we explain our methods for various typographical errors.

Remove Spaces: The proper spacing between words is crucial for natural language processing models to comprehend the text. In this method, we aim to modify tweet content by eliminating certain space characters to merge adjacent words. However, removing all spaces would make the text

	Method	Original Tweet	Modified Tweet
Typos	Remove Spaces	also what's up with this ridiculous weather ? ? it was raining this morning and now it's like super hot ! #weather problems #lame	also what's up with this **ridiculousweather** ? ? it **wasraining** this morning and now it's **likesuper** hot ! #weather problems #lame
	Add Spaces	breaking 911 probably she made a promise to support gun rights to one citizen , while promising to ban guns to the other	**b reaking** 911 pro bably she made a promise to **su pport g un** rights to one **cit izen** , while **pro mising** to **b an** guns to the other
	Shuffle Word Letters	adam smith usa because clearly hillary clinton is a champion for us all	adam smtih usa because **clarely hlliary clitonn** is a **champoin** for us all
	Change Character	men and women should have equal rights, we are all human	men **änd w0men** should have **equä| r!ghts** , we are all **humän**
	Add Hash Signs	hillary clinton hillary for nh hope to see her in not cool soon	hillary clinton hillary **#for** nh **#hope** to **#see** her **#in** not **#cool** soon
Paraphrasing	Use Uncommon Names	hillary clinton hillary for nh hope to see her in not cool soon	hillary **diane** clinton for us hope to see her in not cool soon
	Use Antonyms Together	there's no more normal rains anymore always storms, heavy and flooding	contrary to **normal** there is more **abnormal** rain now always storms, heavy and floods
	Add Hashtag	it's time that we move from good words to good works, from sound bites to sound solutions hillary clinton #ready for hillary	it's time that we move from good words to good works, from sound bites to sound solutions hillary clinton #ready for hillary **#usa #decision #time**
	Remove Hashtags	#fiona bruce wants a government that forces women to have children, and then refuses to financially help them #body autonomy	bruce wants a government that forces women to have children , and then refuses to financially help them
	Use Synonyms	generate belief in quality existence for everyone especially children in that community kitti ngt on 2016	generate belief in quality existence for everyone especially **kids** in that community kitti ngt on 2016
	Use Idioms	also what's up with this ridiculous weather ? ? it was raining this morning and now it 's like super hot ! #weatherproblems #lame	also what's up with this ridiculous weather ? ? it was raining this morning and now **it 's dog days** ! #weatherproblems #lame
	Remove Words	success hillary clinton said she 's receiving a constant barrage of attacks from the right great job , guys keep it up !	success hillary clinton said she 's receiving a constant barrage of attacks from the right great job
	Use Negations	the irish national school system is secular under law we can reaffirm secularism by going through the courts ! humanism ireland	the irish national school system is **not non secular** under law we can reaffirm secularism by going through the courts ! humanism ireland

Table 2: Sample tweets along with the modified versions to deceive models. The modified words are boldfaced.

illegible. Therefore, we select critical words that are likely to be effective for accurate predictions and combine them with the preceding or succeeding word, depending on the context. This process is continued until we believe that the tweet remains readable.

Add Spaces: In this method, we add a space character within the letters of critical words we select to increase out-of-vocabulary words.

Shuffle: Similar to the character swapping method employed by prior work (Sun et al. 2020; Li et al. 2019), we rearrange the order of letters in selected words while keeping the first and last letters intact, inspired by the urban legend known as "Typoglycemia"[6]. While this method might render the text illegible in certain cases[7], we apply this method as long as the words remain recognizable in our modifications.

Change Characters: We use popular writing styles commonly found on social media platforms, where certain letters are substituted with others that have a similar appearance or pronunciation. In particular, our replacement procedure is as follows: a $\rightarrow$ ä, i$\rightarrow$!, l$\rightarrow$ |, o$\rightarrow$ 0, ae$\rightarrow$ æ, $to \rightarrow 2$, $for \rightarrow 4$, and $great \rightarrow gr8$. Although the modified tweets are still comprehensible, it should be noted that they may not appear as professional as the original tweets, which limits the practical application of the method in real-life situations.

Add Hash Signs: Hashtags (HT) are often used to indicate the significance of specific topics. In this method, we add # sign in front of words that are deemed *unimportant* for stance detection. Therefore, the model might be distracted by giving more attention to unimportant words, possibly yielding inaccurate predictions.

Paraphrasing

In this set of methods, our goal is to make significant changes in the tweet contents while maintaining their semantics. The main intuition in these methods is to leverage the inherent bias that models acquire from the datasets they are trained on. For instance, even though BERT models generate contextualized word embeddings, Niven and Kao (2019) report that BERT's predictions are affected by the presence of cue words, especially the word "not". Therefore, the presence of a frequently-occurring word in the training data that is associated with a particular label can impact the outcome, even if the word is not directly related to the label.

One of the main challenges we encountered during our

[6]https://en.wikipedia.org/wiki/Typoglycemia

[7]https://www.mrc-cbu.cam.ac.uk/people/matt.davis/cmabridge/

manual text modification study was that we tried to modify texts written by others. Furthermore, social media posts, which served as the basis for our study, may incorporate dialects, incomplete sentences, and an improper use of language that often contains numerous grammatical mistakes. Hence, we faced difficulty in ensuring that the resulting text was both meaningful and coherent. In some instances, our modifications may have led to unusual language use despite our careful efforts. However, we note that our primary objective is to investigate the effects of specific expressions. Now we explain these methods for the stance detection task.

Use Uncommon Names: In lieu of referencing individuals by their commonly known names, we opt to employ either an abbreviation of their name (e.g., "HC" for "Hillary Clinton") or a less commonly known one (e.g., "Hillary Diane Clinton").

Use Antonyms Together: Using the antonym of a word reverses the meaning. Thus, using two antonyms (e.g., normal and abnormal) together might confuse models while maintaining the meaning.

Add Hashtag: Hashtags might be beneficial to predict the stance of a given tweet. Therefore, to perplex models, we add hashtags that are "neutral" to the stance of the tweet, e.g., "#monday" and "#future".

Remove Hashtag: In this method, we remove hashtags that will not spoil the meaning.

Use Synonyms: We replace words with their synonyms whenever possible (e.g., $children \rightarrow kids$).

Use Idioms: Semantically analyzing idiomatic expressions poses a challenge for language models. Therefore, in this method, we use idioms whenever applicable, such as using "brass monkey" to denote "extremely cold weather" or "raining cats and dogs" to signify "heavy rain".

Remove Words: In this method, we remove tweets' words that do not significantly contribute to the meaning.

Use Negations: The presence of negations may pose a challenge for models since they can reverse the meaning of the words they modify. (e.g., "not" and "without"). Therefore, in this method, we replace positive expressions with negation words and the opposite of the original expression (e.g., "is religious" $\rightarrow$ "is not nonreligious").

Additional Tweets to User Profiles

Mention City: We can talk about a city even though we do not live there. This can potentially deceive models due to mentioning a particular city regularly. In this method, we add a predefined set of tweets in which a particular location is mentioned (e.g., "Hawaii is beautiful!" and "The most expensive houses are in Hawaii") to the given user profile.

Mention Users: State-of-the-art geotagging models utilize text and social networks to predict the location of an individual, as described earlier. Therefore, in this method, we add tweets with dummy text, mentioning other users and changing their social network graph. However, mentioning random users can be considered spamming. In real life, people might get in touch with their friends or celebrities living in different locations or local entities (e.g., local news channels) to apply this method.

	Train			Test		
Topic	F	A	N	F	A	N
AT	92	304	117	32	160	28
CC	212	15	168	123	11	35
FM	210	328	126	58	183	44
HC	112	361	166	45	172	78
LA	105	334	164	46	189	45

Table 3: Label Distribution in Stance Detection Dataset. F: Favor, A: Against, N: None

Experimental Evaluation

Experimental Setup

Datasets. For the stance detection task, we use the dataset of SemEval 2016 Task-6 (Mohammad et al. 2016), which consists of five topics: Atheism (AT), Climate Change (CC), Feminism (FM), Hillary Clinton (HC), and Legalization of Abortion (LA). Each tweet is labeled as one of the three labels: Against (A), Favor (F), and None (N). The label distribution of training and test data is shown in **Table 3**.

For the geotagging task, we use the popular GEOTEXT (Eisenstein et al. 2010) dataset, which includes 9K users and 370K tweets. Each user has a varying number of tweets, and their corresponding latitude and longitude information is provided as labels. We use the same train, validation, and test sets shared by the original dataset creators. The ratio of the train, validation, and test sets are 60%, 20%, and 20%, respectively.

Evaluation Metrics. To measure the performance of stance detection models, we report macro-average F_1 score across favor, against, and none classes for each topic. For the geotagging task, we report mean error (i.e., the distance in miles to the actual location) as in prior work (Rahimi, Cohn, and Baldwin 2018).

Implementation. For the stance detection task, we fine-tune large uncased pre-trained BERT model[8] and pre-trained Twitter-RoBERTa-base model[9] using the train set of each topic. We use 11 epochs with a batch size of 16. We found out that oversampling the rare classes by two improves the performance of BERT for Climate Change and Hillary Clinton topics due to the imbalanced distribution of labels. Therefore, we employed oversampling for these topics. For the geotagging task, we utilized the implementation of GCN and MLP-TXT+NET[10] provided by Rahimi, Cohn, and Baldwin (2018).

Manual Modification for Stance Detection

While we initially used the tweets that were accurately predicted by the BERT model to develop our tweet modification methods, we need a more representative sample which also consists of tweets misclassified by the models, to reliably evaluate the effectiveness of our methods. Hence, we randomly sampled additional tweets to be manually modified.

[8]https://huggingface.co/bert-large-uncased

[9]https://huggingface.co/cardiffnlp/Twitter-RoBERTa-base

[10]https://github.com/afshinrahimi/geographconv

Table 4 shows the distribution of topics in our final sample and the corresponding accuracy of BERT and Twitter-RoBERTa models.

Topic	Tweets	BERT	Twitter-RoBERTa
AT	21	0.952	0.905
CC	11	0.909	0.727
FM	16	0.813	0.688
HC	19	0.947	0.737
LA	15	0.600	0.867
Overall	82	0.854	0.793

Table 4: The number of tweets we manually modified and accuracy of fine-tuned BERT and Twitter-RoBERTa models when original tweets are used for prediction for each topic.

Obviously, the results of manual text modifications depend on the person who performs the modifications. In order to reduce this bias in our results, the modifications are conducted by multiple people. In particular, one of the authors of this paper initiated the process by manually modifying the tweets using the methods explained in the previous section. For each tweet, the author developed three modified versions, applying a different method for each. Subsequently, two other authors of this work also manually modified the tweets using the same methods as the first author. For instance, if the first author applied shuffling, adding hashtags, and changing characters for a particular tweet to create the three modified versions, the others also applied the same techniques for that tweet. While they are required to apply the same method for a particular case, they were free on *how* to apply it. For instance, they can change the characters of different words and come up with different hashtags. This approach allowed us to control the number of trials for each method while creating different versions of a tweet by applying the same method. Eventually, we developed a total of 738 ($= 82 \times 3 \times 3$) modified tweets.

Table 5 shows the number of trials conducted for each method and the ratio of changes in the output of BERT and Twitter-RoBERTa models from a true/false prediction to a true/false one. The number of trials varies for each method because some methods are applicable only to particular instances. For example, in order to apply the "Use Idioms" method, there should be a specific phrase that can be expressed using an idiom. Due to the varying number of trials for each method, we report the ratio of each case with respect to the number of trials.

Regarding **RQ1**, our results indicate that paraphrasing tweets has a limited effect on the predictions of both models, suggesting that both models are able to catch the semantics of tweets even though we use different words. We observe that using uncommon names is effective in changing the predictions of the models in some cases. However, note that it changes incorrect predictions of Twitter-RoBERTa to correct ones in 19% of the cases. Interestingly, while using antonyms together has no impact on BERT predictions, it deceives Twitter-RoBERTa in 22% of the cases.

We also observe that both models are vulnerable to typos, echoing the findings of Sun et al. (2020) for the BERT model. We are able to deceive the BERT model in around one-third of the cases when we change characters with visually similar ones, split important words by adding spaces, and shuffle the letters in the middle of words. We observe that removing spaces is less effective than other typo-based methods for the BERT model. This might be because the BERT tokenizer is able to correctly tokenize two consecutive words written without any space for some cases (e.g., "ridiculousweather"). While Twitter-RoBERTa has lower performance than the BERT model on the original tweets (See Table 4), its performance is less affected by our typographical error based methods compared to the BERT model. This might be because Twitter-RoBERTa has been pre-trained with (typically noisy) tweets, enabling it to handle typographical errors more effectively.

Hashtags appear to be important for the BERT model. We could change a correct prediction to a wrong one in 20% of the cases by removing hashtags. However, it is noteworthy that removing hashtags changed incorrect predictions of both models to correct ones in 5% of the cases. Adding neutral hashtags or converting some words into hashtags also causes inaccurate predictions in 9% and 10% of the BERT predictions, respectively. In contrast, Twitter-RoBERTa seems to be more robust to hashtag changes than BERT, as its performance is not affected by adding hashtags and is slightly affected by adding hash signs and removing hashtags.

For people who do not want to be tracked due to their stances on various issues, changing the predicted stance to neutral might be more important than changing it to an opposite stance. None of the tweets we manually changed has a neutral label; however, when we use the original tweets for prediction, the number of tweets predicted as neutral is six and zero for BERT and Twitter-RoBERTa, respectively. When we use our modified tweets, BERT and Twitter-RoBERTa predict as neutral for 126 and 129 cases (out of 738), respectively, suggesting that modified versions are somewhat effective to change predictions to neutral ones.

As manual modifications are biased to the people who modify the tweets, we investigate whether the effectiveness of these methods varies among individuals who carry out the modifications. **Table 6** shows the number of prediction changes of BERT for each method and each person who modified the tweets. We omit the results for Twitter-RoBERTa due to space limitations. In general, we observe that the performance of methods is similar across people, not changing any of our conclusions about the comparison of methods. However, we also observe that P1, P2, and P3 could change correct predictions to false ones in 46, 51, and 41 cases, suggesting that it is also important how methods are applied. In general, paraphrasing methods have more stable results across people than typo-based methods. For instance, in the "Remove Words" and "Using Antonyms Together" methods, all modifiers have exactly the same performance, likely due to the limited flexibility of these approaches.

	Methods	# Trials	BERT				Twitter-RoBERTa			
			$T \rightarrow T$	$F \rightarrow F$	$T \rightarrow F$	$F \rightarrow T$	$T \rightarrow T$	$F \rightarrow F$	$T \rightarrow F$	$F \rightarrow T$
Typos	Change Character	125	52%	14%	32%	2%	50%	20%	22%	7%
	Add Spaces	84	62%	7%	31%	0%	64%	10%	23%	4%
	Shuffle	93	52%	16%	32%	0%	66%	11%	18%	5%
	Remove Spaces	48	69%	25%	6%	0%	75%	13%	13%	0%
	Add Hash Signs	90	81%	9%	10%	0%	74%	18%	4%	3%
Paraphrasing	Remove Hashtag	55	64%	11%	20%	5%	73%	16%	9%	2%
	Use Synonyms	81	79%	14%	7%	0%	72%	17%	9%	2%
	Add Hashtag	75	71%	16%	9%	4%	76%	24%	0%	0%
	Use Antonyms Together	9	100%	0%	0%	0%	78%	0%	22%	0%
	Use Uncommon Names	21	76%	0%	24%	0%	48%	24%	10%	19%
	Use Idioms	27	96%	0%	4%	0%	93%	0%	0%	7%
	Remove Words	12	75%	25%	0%	0%	50%	50%	0%	0%
	Use Negations	18	72%	17%	6%	6%	67%	22%	6%	6%

Table 5: The impact of our manual text modifications on the performance of BERT and TwitterRoberta models. T stands for True, and F stands for False. $T \rightarrow F$ shows the ratio of the cases we could change the correct prediction of the corresponding model to a false prediction by using the respective text modification method. Similarly, $F \rightarrow T$ shows the number of cases where a false prediction is changed to a correct prediction. $F \rightarrow F$ and $T \rightarrow T$ show the number of cases that do not change the prediction at all. Each method has been applied by three people.

	Methods	$T \rightarrow T$			$T \rightarrow F$			$F \rightarrow T$		
		P1	**P2**	**P3**	**P1**	**P2**	**P3**	**P1**	**P2**	**P3**
Typos	Change Character	24	20	21	12	14	14	1	0	1
	Add Spaces	18	15	19	8	11	7	0	0	0
	Shuffle	16	13	19	10	13	7	0	0	0
	Remove Spaces	11	12	10	1	0	2	0	0	0
	Add Hash Signs	22	26	25	5	1	3	0	0	0
Paraphrasing	Remove Hashtag	11	13	11	4	3	4	1	1	1
	Use Synonyms	20	21	23	3	2	1	0	0	0
	Add Hashtag	18	16	19	2	4	1	2	1	0
	Use Antonyms Together	3	3	3	0	0	0	0	0	0
	Use Uncommon Names	6	4	6	1	3	1	0	0	0
	Use Idioms	9	9	8	0	0	1	0	0	0
	Remove Words	3	3	3	0	0	0	0	0	0
	Double Negations	5	5	3	0	0	1	1	0	0
Total		166	160	170	46	51	42	5	2	2

Table 6: The impact of manual text modifications of each person involved in the experiment (represented as P1, P2, and P3) on the predictions of the BERT model. We show the number of cases for each prediction change type. We omit the results for $F \rightarrow F$ for simplification.

Automatic Modification for Stance Detection

The previous experiment involves the manual modification of a subset of tweets. In this section, we investigate the impact of our methods when applied automatically to the entire dataset. To this end, we undertake the following steps to conduct this set of experiments.

Our previous experiments show that our paraphrasing methods are not effective in deceiving the models. In addition, it is challenging to apply them automatically. Consequently, in this set of experiments, we turn our attention to methods that are potentially effective and can easily be applied automatically. In particular, we employ the following methods: "Add Hash Signs", "Add Hashtag", "Remove Hashtag", "Change Character", "Shuffle Word Letters", "Add Spaces", and "Remove Space".

In the manual modification, we did not put any restriction on the number of words that needed to be changed. However, for automatic modification, we introduce a parameter denoted by N, which specifies the number of words to be modified and the number of hashtags to be removed/added. We vary N from 0 to 4 in our experiments.

In our previous experiments, we manually selected "critical" words to modify. In this set of experiments, we adopt the following methodology to select the words to be modified. We first rank all words in a tweet based on their cosine similarity to the respective topic words (e.g., abortion) using fastText word embeddings (Bojanowski et al. 2017). Next, we pick the closest N words for modification. However, applying the "Remove Space" method to consecutive words might cause unreadable tweets. Hence, we select N non-consecutive words for this method. Similarly, in order

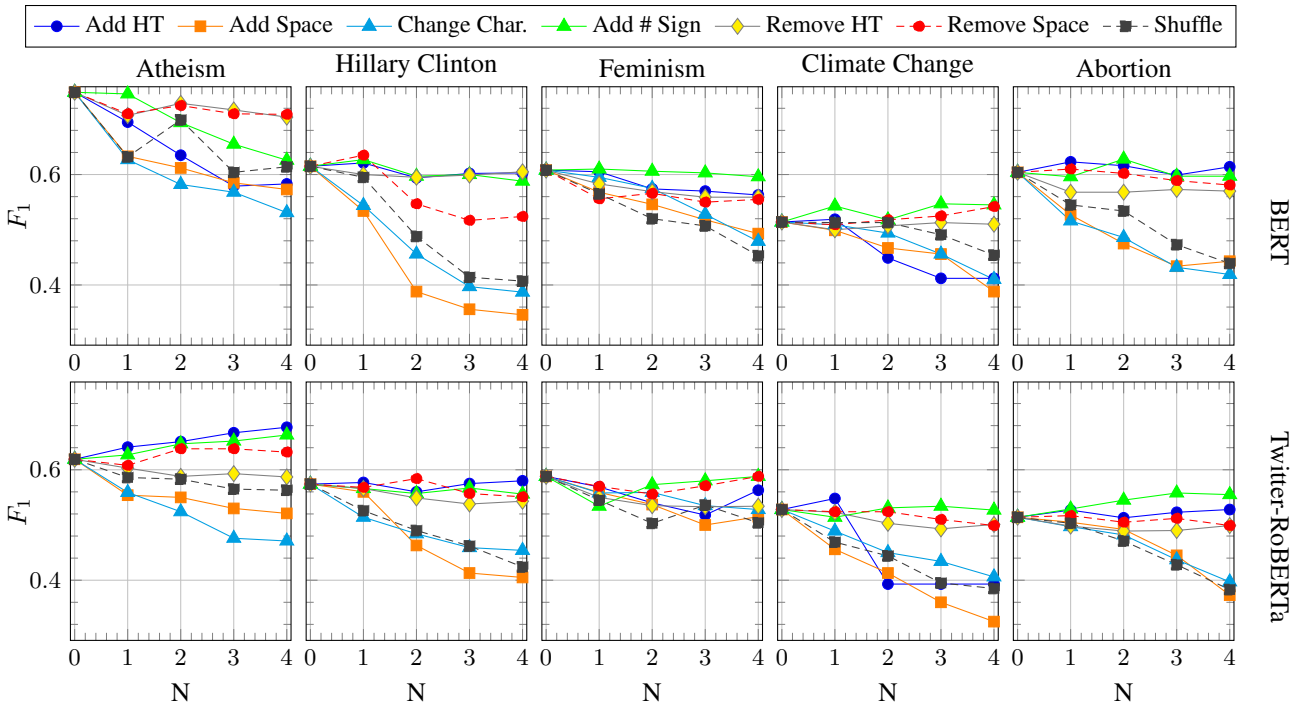

Figure 1: Performance of fine-tuned BERT and Twitter-RoBERTa models in stance detection task on the test data of the respective dataset of SemEval2016 for varying distortion numbers. For instance, $N = 4$ means that the respective method has been applied for four words. The upper row shows F_1 score of BERT and the bottom row shows the F_1 score of Twitter-RoBERTa.

to detect words to be converted to hashtags in the "Add Hash Sign" method, we use the most distant ones to the topic words, because we convert the unimportant ones as explained in the previous section.

In the "Add Hashtag" method, we manually defined a hashtag list for each topic and used N of them. For instance, the hashtags for the abortion topic are #MondayMotivation, #goals, #opinion, and #thoughts. In addition, in the "Shuffle" method, if the selected word has 7 or more letters, we only change the position of the 2^{nd}, 3^{rd}, 4^{th}, and 5^{th} letters, keeping the distance between the original position of a letter and its position in the jumbled version at most three. This is because it is likely that words will become unreadable when the position of a letter is changed a lot.

In this experiment, we modify each tweet in the test set as explained above and report the performance of BERT and Twitter-RoBERTa model fined tuned on the original train data. The results are shown in **Figure 1**. Our observations regarding the results are as follows. Firstly, "Remove Hashtag" seems the least effective method among others. On the other hand, similar to our experiments with manually modified texts, the "Change Character", "Add Space", and "Shuffle" methods seem to be the most effective ones, decreasing the performance of the BERT model by 28%, 27%, and 23% on average across five topics when $N = 4$, respectively. Similarly, "Change Character", "Add Space", and "Shuffle" decrease the performance of Twitter-RoBERTa by 20%, 25%, and 20% on average across five topics when $N = 4$, respectively.

Our investigation of the "Add Hashtag" method produced mixed findings. While it has a slight impact on both models' performance in most cases, for the topic of atheism, it decreases the performance of the BERT model but improves Twitter-RoBERTa's performance. We observe a similar pattern for the "Add Hash Sign" method. These results suggest that hashtags might have a correlation with the labels in the training data. Therefore, it is risky to use hashtags without knowing the training data of models.

Regarding BERT vs. Twitter-RoBERTa, BERT yields higher performance than Twitter-RoBERTa when $N = 4$. Our first expectation was that Twitter-RoBERTa would be less affected by the typos we introduced because it is pretrained with noisy data. However, there is no meaningful difference between the models' relative performance changes in our experiments with automatic modifications.

Regarding **RQ2**, we investigate whether the resultant texts after our automatic modifications are readable and have the same/similar meaning to the original tweet. In particular, we first randomly sampled five tweets for each method and each N value (i.e., $5 \times 7 \times 4 = 140$ cases in total). Subsequently, two authors of this paper manually inspected each tweet. If a tweet contains at least one word which could not be identified by at least one of the authors, we consider that tweet as not readable. **Figure 2** shows the ratio of tweets that are readable and have the same/similar meaning as the original tweets among the tweets we inspected for each case. We see that all methods except "Shuffle" and "Add Space" yield readable tweets. "Shuffle" makes 40% of tweets un-

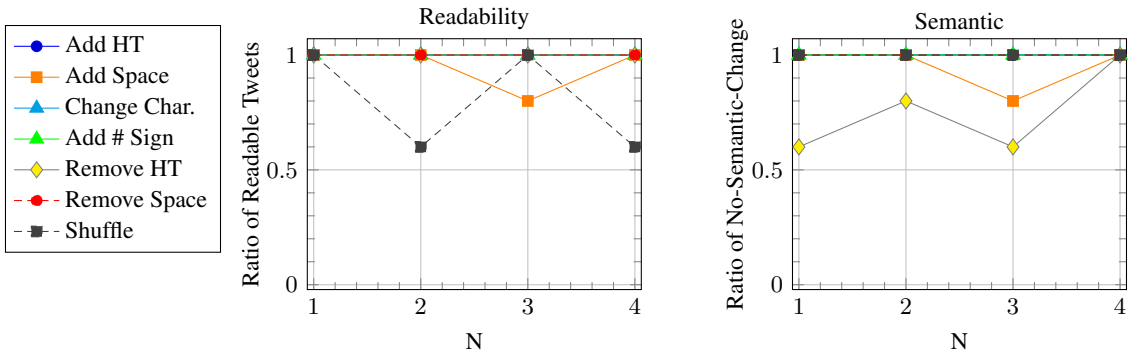

Figure 2: Readability and semantic change analysis in stance detection task for the varying number of distorted words. We analyzed 140 cases to understand whether the resultant tweets are readable and have the same semantics as the original tweets when our methods are applied automatically. y-axis shows the ratio of readable tweets and the ratio of tweets without any semantic change among tweets we manually inspected. The x-axis represents the number of words affected by our methods.

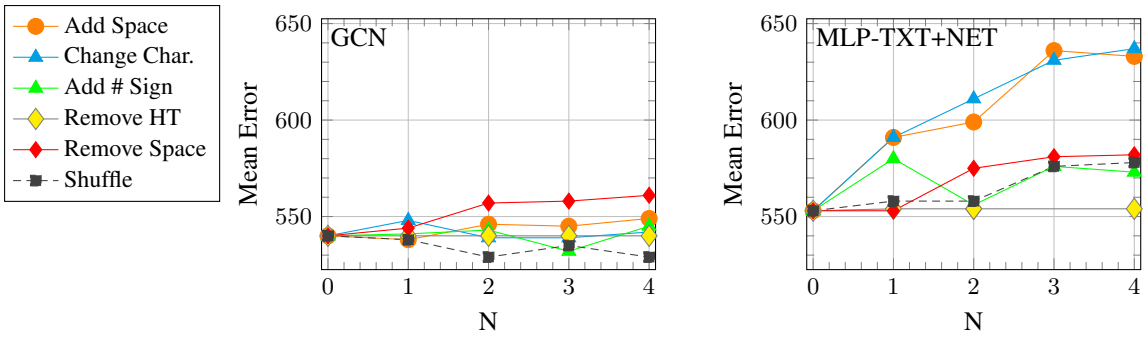

Figure 3: Mean error scores of geo-tagging models when our methods are applied for a varying number of times in all tweets of users in the test data of GEOTEXT dataset.

readable when $N = 2$ and $N = 4$, reducing its applicability. For instance, the following tweet is one of those unreadable ones where correct versions of words are written in parenthesis: "*ny investing big bkaenr (banker) bdus (buds) need to ratchet up their haillry (hillary) cares about the little polepe (people) propaganda*".

Regarding the semantic change, we observe that none of our methods, except "Add Space" and "Remove Hashtag", change the semantics of tweets. "Remove Hashtag" causes semantic change because we notice that people use hashtags for important words in a tweet. For example, in the following tweet, the removed hashtag (shown as strikethrough text) is essential in the meaning of the sentence: "*agent 350 this is not a fantasy this is negligence collusion with criminal corporations acting with negligence to ~~#ecocide~~*". One might ask how "Shuffle" does not cause any semantic change when tweets are not readable. In those cases, the changed words do not mean any other meaningful word. Therefore, we assumed that if a person can correctly read them, it would not cause any change in the meaning. Nevertheless, our qualitative analysis suggests that both the "Shuffle" and "Remove Hashtag" methods require special attention to ensure that the tweets are readable and their semantics remain unchanged.

Automatic Modification for GeoTagging

Regarding **RQ3**, we apply the automatic methods used in the previous experiment for the geotagging task in this set of experiments. Similarly, we use the parameter N to control the number of modified words and hashtags that are added/removed. We apply our methods to all tweets of users in the test set. As there is no topic in the geotagging task, we randomly select words to be modified instead of relying on our fastText-based similarity calculation. In our modifications, we do not change any mentioned user to avoid altering the social network used by the models. Moreover, we do not use the "Add Hashtag" method in geotagging, because there is no specific topic to be neutral. The results are shown in **Figure 3**.

Generally, MLP-TXT+NET's performance decreases as N increases in all methods, except "Remove Hashtag". In fact, the "Remove Hashtag" method has no impact on the performance of both models. This might be because both models represent texts as bag-of-words and hashtags in the test set might not appear in the train set. Furthermore, the results indicate that modifications to the tweet content have a slight impact on the performance of GCN, implying that the social network plays a critical role in its prediction.

Next, we increase the number of tweets of each user us-

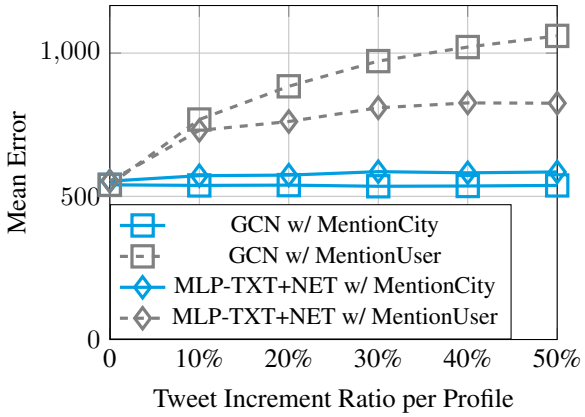

Figure 4: The impact of adding additional tweets created by our methods on the performance of two geo-tagging models, GCN and MLP-TXT+NET.

ing our "Mention City" and "Mention Users" methods, separately. We vary the increment ratio per profile from 10% to 50%. The results are shown in **Figure 4**. We observe that having many tweets mentioning cities has a limited impact on both models' performance. However, when we introduce modifications to the social network structure by mentioning random users, their performance decreases dramatically. Overall, our experiments suggest that users who want to conceal their location from AI models can interact with users (e.g., celebrities and local entities in different places) located in various places, rather than adding typographical errors or mentioning location names explicitly.

Ethical Discussion

The primary objective of our research is to investigate methods that could potentially mitigate the negative consequences of AI models, which can easily be weaponized against individuals. However, as weapons, the same AI models can be utilized for harmful purposes such as surveillance of individuals, or conversely, for benevolent purposes such as preventing the dissemination of misinformation and hate speech. Therefore, individuals who spread misinformation or hate can also use similar techniques to evade AI models that might detect their toxic messages. On the other hand, our methods will also be helpful for individuals who just do not want to be tracked by people they even do not know. Using the analogy of AI models being weapons, our approaches can be considered armors that can protect against these models. We firmly believe that there should be available armors in the market if we know that there are people with weapons. Our study makes a modest step towards this goal. We anticipate that our work will inspire other researchers to work on this important research direction and will develop more effective solutions than ours.

Conclusion

In this work, we investigated how individuals can protect their privacy from AI models while using social media platforms. We focused on stance detection and geotagging tasks and explored fifteen different text-altering methods, such as inserting typographical errors in strategic words, paraphrasing, changing hashtags, and adding dummy social media posts. Based on extensive experiments we conducted, our recommendations for people who do not want to be tracked by AI models on social media platforms are as follows. Firstly, paraphrasing methods were found to be ineffective in deceiving the models. While other language models besides BERT may be used in real-life applications, other large models are likely to have comparable performance in identifying text semantics. Secondly, changing characters with visually similar ones, splitting words by adding spaces, and shuffling the character order are effective in decreasing stance detection models' performance. However, these methods require special attention because the resultant text might be unreadable. Lastly, to deceive geotagging models, the most effective way is to interact with a diverse set of users.

In the future, we plan to extend our work in several directions. Firstly, we plan to increase the dataset size and involve a larger number of individuals in modifying the texts to ensure the robustness and generalizability of our findings. We will explore other tasks focusing on predicting personal information about individuals such as race, ethnicity, and mental health. We also plan to develop more sophisticated methods to fool AI models. Additionally, we plan to conduct a user study to investigate whether people are aware of AI models and their capabilities in detecting personal information. Furthermore, we will explore other datasets based on social media platforms other than Twitter to reduce platform-specific bias in our experiments. Moreover, we plan to reach vulnerable communities, such as immigrants, and extend our work based on their specific needs. Lastly, we will develop an automated tool to modify messages to prevent tracking. We plan to leave the development of such a tool as our final goal because a tool that does not work well might be harmful by giving false hopes to people who would like to use it. Therefore, we will also explore explainable AI techniques so that the users will be able to interpret its output and act accordingly.

Acknowledgments

This study was funded by the Scientific and Technological Research Council of Turkey (TUBITAK) ARDEB 3501 Grant No 120E514. The statements made herein are solely the responsibility of the authors.

References

Baly, R.; Mohtarami, M.; and Glass, J. 2018. Integrating Stance Detection and Fact Checking in a Unified Corpus. In *Proceedings of NAACL-HLT*, 21–27.

Barzilay, R.; and McKeown, K. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, 50–57.

Belinkov, Y.; and Bisk, Y. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. In *International Conference on Learning Representations*.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5: 135–146.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv:2005.14165*.

Chen, X.; Salem, A.; Backes, M.; Ma, S.; and Zhang, Y. 2020. Badnl: Backdoor attacks against nlp models. *arXiv:2006.01043*.

Dai, J.; Chen, C.; and Li, Y. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7: 138872–138878.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Dwi Prasetyo, N.; and Hauff, C. 2015. Twitter-based election prediction in the developing world. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, 149–158.

Ebrahimi, J.; Rao, A.; Lowd, D.; and Dou, D. 2018. HotFlip: White-Box Adversarial Examples for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 31–36.

Eisenstein, J.; O'Connor, B.; Smith, N. A.; and Xing, E. 2010. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, 1277–1287.

Feyisetan, O.; Ghanavati, S.; and Thaine, P. 2020. Workshop on Privacy in NLP (PrivateNLP 2020). In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 903–904.

Ghahremanlou, L.; Sherchan, W.; and Thom, J. A. 2015. Geotagging twitter messages in crisis management. *The Computer Journal*, 58(9): 1937–1954.

Ghosh, S.; Singhania, P.; Singh, S.; Rudra, K.; and Ghosh, S. 2019. Stance detection in web and social media: a comparative study. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, 75–87. Springer.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv:1708.06733*.

Jia, R.; and Liang, P. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2021–2031.

Jin, D.; Jin, Z.; Zhou, J. T.; and Szolovits, P. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 8018–8025.

Küçük, D.; and Can, F. 2020. Stance detection: A survey. *ACM Computing Surveys (CSUR)*, 53(1): 1–37.

Kurita, K.; Michel, P.; and Neubig, G. 2020. Weight Poisoning Attacks on Pretrained Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2793–2806.

Li, J.; Ji, S.; Du, T.; Li, B.; and Wang, T. 2019. TextBugger: Generating Adversarial Text Against Real-world Applications. In *26th Annual Network and Distributed System Security Symposium*.

Liang, B.; Li, H.; Su, M.; Bian, P.; Li, X.; and Shi, W. 2018. Deep text classification can be fooled. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 4208–4215.

Mieskes, M. 2017. A quantitative study of data in the NLP community. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, 23–29.

Mohammad, S.; Kiritchenko, S.; Sobhani, P.; Zhu, X.; and Cherry, C. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 31–41.

Morgan-Lopez, A. A.; Kim, A. E.; Chew, R. F.; and Ruddle, P. 2017. Predicting age groups of Twitter users based on language and metadata features. *PloS one*, 12(8): e0183537.

Morris, J.; Lifland, E.; Lanchantin, J.; Ji, Y.; and Qi, Y. 2020. Reevaluating Adversarial Examples in Natural Language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 3829–3839.

Muller, B.; Sagot, B.; and Seddah, D. 2019. Enhancing BERT for Lexical Normalization. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, 297–306.

Niu, T.; and Bansal, M. 2018. Adversarial Over-Sensitivity and Over-Stability Strategies for Dialogue Models. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 486–496.

Niven, T.; and Kao, H.-Y. 2019. Probing Neural Network Comprehension of Natural Language Arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4658–4664.

Preoţiuc-Pietro, D.; and Ungar, L. 2018. User-level race and ethnicity predictors from twitter text. In *Proceedings of the 27th International Conference on Computational Linguistics*, 1534–1545.

Rahimi, A.; Cohn, T.; and Baldwin, T. 2018. Semi-supervised User Geolocation via Graph Convolutional Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2009–2019.

Rashed, A.; Kutlu, M.; Darwish, K.; Elsayed, T.; and Bayrak, C. 2020. Embeddings-Based Clustering for Target Specific Stances: The Case of a Polarized Turkey. *arXiv:2005.09649*.

Ren, K.; Zheng, T.; Qin, Z.; and Liu, X. 2020. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3): 346–360.

Schiller, B.; Daxenberger, J.; and Gurevych, I. 2021. Stance detection benchmark: How robust is your stance detection? *KI-Künstliche Intelligenz*, 35(3): 329–341.

Sekulic, I.; and Strube, M. 2019. Adapting Deep Learning Methods for Mental Health Prediction on Social Media. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, 322–327.

Silva, P.; Gonçalves, C.; Godinho, C.; Antunes, N.; and Curado, M. 2020. Using NLP and Machine Learning to Detect Data Privacy Violations. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 972–977. IEEE.

Sun, L. 2020. Natural backdoor attack on text data. *arXiv:2006.16176*.

Sun, L.; Hashimoto, K.; Yin, W.; Asai, A.; Li, J.; Yu, P.; and Xiong, C. 2020. Adv-BERT: BERT is not robust on misspellings! Generating nature adversarial samples on BERT. *arXiv:2003.04985*.

Yang, W.; Li, L.; Zhang, Z.; Ren, X.; Sun, X.; and He, B. 2021. Be Careful about Poisoned Word Embeddings: Exploring the Vulnerability of the Embedding Layers in NLP Models. *arXiv:2103.15543*.

Yavuz, D. D.; and Abul, O. 2016. Implicit location sharing detection in social media turkish text messaging. In *International Workshop on Machine Learning, Optimization, and Big Data*, 341–352.