Master Degree Project

Language Independent Detector for Auto Generated Tweets

Author: Saeideh Valipour
Supervisor: Jonas Lundberg
Semester: HT 2020
Subject: Computer Science

Abstract

The cross-disciplinary Nordic Tweet Stream (NTS) is a project aiming at creating a multilingual text corpus consisting of tweets published in the five Nordic countries. The NTS linguists are explicitly interested in tweets having a text formulated by a human where each tweet is a personal statement, not in Tweets generated by bots and other programs or apps since they might skew the results. NTS consists of multiple parts and the part we are responsible for is a language-independent approach, using supervised machine learning, to classify every single tweet as auto-generated (AGT) or human-generated (HGT). The objective of this study is to increase data accuracy in sociolinguistic studies that utilize Twitter by reducing skewed sampling and inaccuracies in linguistic data.

We define an AGT as a tweet where all or parts of the natural language content are generated automatically by a bot or other type of program. In other words, while AGT/HGT refers to an individual message, the term bot refers to non-personal and automated accounts that post content to online social networks.

Our approach classifies a tweet using only metadata that comes with every tweet, and we utilize those metadata parameters that are both language and country independent. The empirical part shows that our results show poor success rates when it comes to unseen data. Using a bilingual training set of two languages tweets, we correctly classified only about 60-70% of all tweets in a test set using a third new language, which is still better than nothing, but probably not good enough to be used (as is) in a real-world scenario to identify AGTs in a given set of multilingual tweets.

Keywords: Twitter, Machine Learning, Classification, Bot Detection, Social networks

Preface

I would like to sincerely express my gratitude to my supervisor, Professor Jonas Lundberg, who was guiding me with brilliant thought through the whole process by being patient and sharing his knowledge and experience with me and being always available for me. My effort to complete this thesis would not be fruitful without his support.

I also express my appreciation to the reader, who read my thesis and my examiner for helping me with my problems and guiding me through my thesis improvements.

My heartfelt appreciation goes to my wonderful parents who support me and encourage me to never give up.

Contents

1. Introduction

This chapter presents an introduction to this thesis. It starts with some background information in Section 1.1 and continues with the literature review for this report in Section 1.2. Section 1.3 and 1.4 explain the problem statement and motivation, respectively. Objectives are discussed in Section 1.5. It proceeds by Scope and Target group section in Section 1.6 and Section 1.7 and an outline of the report structure is explained in Section 1.8.

1.1 Background

In recent years, big data from various social media applications have turned the web into a user-generated repository of information in an ever-increasing number of areas. The popularity and remarkable simplicity of Twitter by publishing text-based posts, known as tweets, have attracted a large number of automated programs, known as a bot to many industries. Also, because of easy access to metadata of each tweet, Twitter has become a popular source of data for investigations of a number of phenomena such as studies of the Arab Spring [1], various political campaigns [2][3], of Twitter as a tool for emergency communication [4][5], and using social media data to predict stock market prices [6]. In linguistics, various mono-[7] and multilingual text corpora of tweets [8] have been built recently and used in a wide range of subfields (e.g. dialectology, language variation and change).

The popularity of Twitter as an instrument in public debate has led to a situation in which it has become an ideal target of spammers and automated programs. It has been estimated that around 5-10% of all users are bots[1], and that these accounts generate about 20-25% of all tweets posted on Twitter[2]. For research purposes, bots present a serious problem because they reduce data accuracy and may dramatically skew the results of analyses using social media data [9].

In Computer Science, the various bot detection approaches typically apply machine learning based on tweet metadata. More about Twitter automation and existing techniques to detect auto-generated tweets can be read in Chapter 2. This thesis is aiming to build a language independent bot detection for the Nordic Tweet Stream (NTS) dataset by evaluating different experiments.

[1] www.nbcnews.com/business/1-10-twitter-accounts-fake-say-researchers-2D11655362

[2] sysomos.com/inside-twitter/most-active-twitter-user-data/

1.2 Problem formulation

In previous research, supervised learning has proven to be an effective way to detect automation in Online Social Network (OSNs) [9][10][11][12][13][14]. We are going to present a language independent approach for detecting AGTs. The input to the AGT classifier consists of 16 tweet properties attaining numerical and nominal values that can be computed directly using the tweet metadata. The fact that the actual Twitter text is not used in the classification makes the classification language independent. In this project, datasets contain three different languages and we are exploring different combinations of languages as training and test set to evaluate them.

Lastly, our expected result is: We expect that our multilingual approach will not be as good as the monolingual approach and we would like to evaluate what the accuracy of using a multilingual detector compared to monolingual one would be. In other words, since we are not taking the text into account, we have less information, so we expect the classification to be less precise. We would like to estimate this loss of precision.

1.3 Motivation

Most of the previous work related to auto detection tweets have focused on one language only (monolingual), English tweets [10][11][15][16][17] (or very multilingual random samples of all tweets [18],[19]). (1) Since the classifier is trained on a monolingual set of tweets (English in most cases), taking the text into account, the classifier becomes language dependent. The main disadvantage of being language dependent is that it requires a new classifier for each language. In other words, it cannot be applied on multilingual data sets.

Throughout this thesis we are going to present a language and country independent approach to detect AGTs using a multilingual training set. This idea fascinates many researchers because the actual Twitter text is not used as an input feature in the classifier. Using only properties of Tweet as metadata available for each tweet makes our approach light, also it would be possible to classify tweets in real time, as a part of a Twitter downloading stream.

1.4 Research Questions

This research is going to answer the following questions:

RQ1. What is a suitable method to detect Auto Generated Tweet (AGT) in a multilingual dataset of tweets?

RQ2. What is the accuracy of using a multilingual classifier compared to a monolingual classifier?

1.5 Objectives

A list of objectives needed to complete this project is shown below:

O1	Build a supervised machine learning based tweet classifier which classifies individual tweet rather than Twitter user account.
O2	Implement and evaluate Monolingual and Multilingual classifiers using machine learning models based on 25,000 manually labeled tweets.
O3	Implement and evaluate Bilingual classifier (test on an unseen data) using different combinations (e.g. train on two different languages, evaluate using another new language)

Table 1.1 List of Objectives

After all the objectives are met, a language independent detector for AGT is expected to be the result. The approach is language independent since the actual Twitter text is not used as an input feature in the classifier. In fact, the algorithm classifies each tweet using only selected attributes in the Twitter metadata available for each tweet.

1.6 Scope/Limitation

During the work on this project, there were limitations which made it impossible to do everything that was initially planned. The first limitation was the quality of data. According to the results our language independent parser does not work properly since English dataset is rather odd compared to Swedish and Finnish dataset, the former has a much higher rate of AGTs.

Since machine learning models need data to learn from, it is necessary to have equal distribution on our dataset. However, we did not. English dataset has number of bots that are publishing once every hour which is not common for Swedish and Finnish datasets. We learned that the idea of language independent Twitter bot is not as straightforward as we wanted it to be. We did not reach a result which we aimed/hoped for. Moreover, trying to develop a multilanguage classifier using a dataset containing only three languages is, of course, a limitation. However, creating a new dataset for a new language means to manually classify 5000-10000 tweets as AGT/HGT, is very time consuming and beyond the scope of this project. Hence, we used the three datasets (English, Swedish, Finnish) that were available.

1.7 Target group

Different parties could potentially be interested in this project from different perspectives. The thesis topic belongs to data scientist in the field of analysis of social media data by using statistical methods. Also, it could be interesting mainly to a group of sociolinguists studies that utilize Twitter by reducing skewed sampling and inaccuracies in linguistic data.

1.8 Outline

This thesis is organized as follows: Chapter 1 is an introduction to the research topic. Chapter 2 discusses automation in Twitter and literature review to provide a background for the topic under research by explaining existing different approaches and technologies. Methodology, datasets and a motivation for the experiments are presented in Chapter 3.

Chapter 4 explains framework and tools that are used during the experiment and describe different parts of the implementation.

Chapter 5 is devoted to experiments and evaluation. Firstly, the three experiments will be presented and motivated. Later the result for each experiment is presented and analyzed. This chapter explains them in the proper context and a summary is included at the end of each experiment.

In Chapter 6 the final conclusion from this study is summarized and suggestions for further research is presented.

2. Background and Related work

In this chapter, a review of automation on Twitter research will be presented. First of all, Section 2.1 explains about fundamentals of Twitter and is divided into three subsections. Fundamentals of Twitter as subsection 2.1.1, Twitter Automation as subsection 2.1.2 and Bot Detection vs AGT Detection as subsection 2.1.3. Section 2.2 reviews the related work based on three different categories and divided into three subsections. Tweet Behavior as subsection 2.2.1, Tweet Content as subsection 2.2.2 and Account properties as subsection 2.2.3.

2.1 Twitter Review

It is helpful to start with a fundamental introduction to our domain and the problems approached in this thesis. This section introduces the concepts and principles of Twitter.

2.1.1 Fundamentals of Twitter

Twitter is popularized as a microblogging platform, since its launch in 2006. Users communicate on Twitter by publishing 280-character limited text-based posts, known as "Status updates" in microblogging communities, also called "tweets"[3]. The two key components of Twitter are Tweets and users.

Users can subscribe to other users' message flows, which is known as "following" and a subscriber is referred to as a "follower". Unlike most Online Social Networks (OSNs) like Facebook and LinkedIn, a "following" on Twitter is not mutual. Hence, the user being followed is not required to follow back. Users can tweet via the Twitter website or by using external applications[4].

Moreover, there several Twitter features that empowers Twitter users to reach an even larger audience with their tweets. The functions include hashtags, mentions and retweets. Hashtags, strictly speaking, non-spaced phrases prefixed with a # symbol, are used to categorize tweets by keywords or topics. A hashtag used in many user tweets will make it to the list of trending topics on Twitter. For example, #OccupyCentral and #BlackLivesMatter [20] were two hashtags trending in 2014. By having enough tweets with hashtag in their tweets, they will make it into a list of trending topics on Twitter. In December

[3] www.twitter.com

[4] Like applications for smart-phones and other mobile devices.

2018 Twitter was ranked as the 11th most popular website in the world by the Alexa[5] ranking, publishing more than 4 million tweets each day.

In numerous research fields, Twitter is popular because of the wide and easy open policy access to all tweets through a service called Twitter Streaming API[6]. It enables developers to connect to the Twitter server and download tweets in real time from a certain region or in a certain language. Due to the increase in the diversity of the user pool, Twitter supports the public timeline API to collect the information, as well. For instance, in our case study, we have downloaded tweets to catch six months of tweets in five Nordic countries.

Furthermore, Twitter has released a timeline mechanism (a REST API) function that gives access to the latest tweets published by a given user.

2.1.2 Twitter Automation

The increasing popularity of Twitter has also made it a target for spam and automated programs, known as bots. Although Twitter imposes strict anti-spam policies [21] loads of bot activity are present on Twitter. Automation on Twitter can be observed as accounts that automatically post tweets with the use of external software programs. Recently, it has become common to spread content through these accounts. As Chu, et al. mentioned [10], automation is a double-edged sword to Twitter. On one hand, automated software is used to produce a high number of harmless tweets, such as blog updates and news. For instance, there is an automated software on Twitter that detects earthquakes promptly and sends email to registered users [5]. It also could be an account (bot) posting weather forecast for Stockholm.

On the other hand, automated bots might present a serious problem. One of the most common automated software is spammers to spread spam, viruses and other malicious content [10]. Chu, et al, declare that these malicious bots often randomly add users as their friends, expecting some users to follow them back. Another definition that is declared by Mowbray is that the bots are explicitly programmed to attract followers themselves to follow the twittering machine back [22].

Moreover, as it is mentioned in [23], 77% of the automated spammer accounts on Twitter were suspended from the first tweet on the first day. However, Boshmaf et al [24] declared that “social bots” are designed to imitate ordinary human behavior on OSNs to infiltrate online communities, gain their trust and send private messages to sway opinions and get intended action [25].

[5] www.alexa.com/topsites

[6] dev.twitter.com/streaming/overview

Furthermore, Chu, et al. [10] found that 10.5% of all Twitter accounts in the data set are bots and 36.2% are “cyborgs”, semi-automated bots defined as being either “human-assisted bots” or “bot-assisted humans” making a distinction between bots and humans harder. Zhang and Paxson [11] stated that 16% of Twitter accounts display huge levels of automation.

It is worth mentioning that a majority of bots is simply publishing certain information that might be useful for certain followers. For example, sports scores or a bot publishing the current Växjö weather every hour. In any case, bot generated tweets might skew the results in research activities assuming that each tweet is a personal statement.

2.1.3 Bot Detection vs AGT Detection

Bot accounts are a particular characteristic of different social media applications and Twitter as well. Non-personal and automated accounts that post content to online networks. A bot refers to a heterogeneous collection of account types, which posts tweets automatically.

In this thesis, we are planning to present a language independent approach, which classifies every single tweet to be either Auto-Generated Tweet or Human-Generated Tweet. What we mean by AGT is an individual tweet that all or part of the natural language content of the tweet is generated automatically by a bot or other type of program [26]. In addition to bot accounts, another source of AGTs are applications that human users use once in a while to post a message on Twitter. For example, the application Runkeeper can be used to publish a tweet to present one’s efforts when working out. The goal of this study is to classify individual tweets as an AGT or not, rather than trying to identify bot accounts.

2.2 Related work

Literature review for bot detection has shown that, in the past, various solutions with different approaches to this topic exist. We categorize earlier approaches among papers in Computer Science, which used to recognize the bot generated tweets. As background materials, there is a list of approaches they have solved similar problems earlier on. Also, in some papers a combination of these categories is used to detect the bot generated tweets:

- Tweet Behavior: detect the periodic or regular timings between tweets
- Tweet Content: detect text patterns of known spam on Twitter
- Account Properties: use account-properties in order to distinguish bot generated tweets

In comparison to related work, in various papers, authors mostly focus on classifying whether a user account is a bot or not, our approach focus on classifying a tweet itself not a user account, plus in language independent approach.

2.2.1 Tweet Behavior

An entropy component measures the tweeting behavior as periodic and regular timing between tweets. Similarly, Grier et al. [15] proposed behavior-principle features to detect spammers on Twitter. Timing patterns extracted from time-stamp information associated with each tweet were used to test non-uniform tweeting behavior. The author also conducted connection between timing patterns and Twitter clients, which allowed pre-schedule tweets at specific time intervals. A x^2-test was used to evaluate whether tweets from an account appeared to be drawn uniformly across a second-of-the-minute and minutes-of-the-hour distribution. They also consider repetition in tweet content and links as a detecting automated behavior [15].

Amleshwarm, et al. [16] also presented a feature based on entropy between tweets. Moreover, content principle characteristics like repetitiveness in tweets and entropy feature to distinguish spam accounts from legitimate accounts.

Paxson et al. [11] proposed an approach whether timing patterns could exclusively be used for spam bot detection on Twitter. Second-of-the-minute and minutes-of-the-hour distribution with Pearson's x^2-test were used apart from non-uniform timing patterns indicating certain degrees of automation.

2.2.2 Tweet content

Text patterns of known spam on Twitter and compared tweets based on the tweet content is also used to detect automation on Twitter. In [18] Araujo et al. presented an approach to detect spam tweets in isolation and without previous information of the user. It was based on language in trending topics. It is chosen based on the topics of conversation that are on everybody's lips.

Benevenuto et al. [17] studied features of spammers in regard to tweet content and social behavior. A classification model was built to categorize spammers and non-spammers on a manually labeled data collection.

2.2.3 Account properties

Account-related properties such as account age, username length, number of user mentions/replies, the number of retweets and the number of followers are generally used in order to distinguish between bots and humans. Many of these properties have been discussed in various papers [9-11, 14, 20, 26]. This is the approach we used in this project, and a set of thirteen tweet properties attaining numerical and nominal values will be presented in Section 4.6.

3. Method

The method chapter explains the methods we used to solve the problem under this study. These two are literature review and controlled experiment. This chapter consists of three sections. In section 3.1 we will explain how literature review helped to answer research question that this thesis focuses on. Section 3.2 will clarify our specific research method. Section 3.3 explains how we collected our data and increased their validity.

3.1 Literature Review

Collecting data from what other authors have published on the same or similar topic is a method that is called Literature review. In this case, it involved discovering books and articles to get visions on various materials.

In this research, I started with five papers given to me by my Supervisor, then (using Google Scholar) by investigating what papers they are referencing or references by, going backwards and forwards, I collect relevant papers not only about bot detection and tweet automation, but also about malicious use and spamming in a slightly larger area.

It was highly valuable when it comes to methods of data preprocessing. Furthermore, model selection and different types of machine learning approaches for various types of problems, such as unsupervised and supervised learning and others. Since in different circumstances different models perform more efficiently than the others. Last but not least, literature review had an influence on tools and frameworks used during the project. For instance, several Python libraries such as Scikit-learn [27] or Tensor flow [28] were explored and used in different development process.

3.2 Controlled Experiment

We use controlled experiments to answer the main research question of this thesis. A controlled experiment method contains two kinds of variables, independent and dependent that affect the input and output of the experiment, respectively. The independent variables are those that an experimenter manipulates to have a direct effect on the dependent variables that are the outcomes (i.e. results).

For solving this problem, three experiments were conducted. The purpose of the first one was to establish a baseline to know more about individual languages and their properties, also to have something to compare with. Then we did one more experiment to check the accuracy of mixing languages. In the third experiment, training in two languages dataset and test on a new unseen language, we were aiming to discover how they handle unknown languages.

Performing these experiments in machine learning method provides a well-established process. Machine learning algorithms are recommended for use because of their ability to learn from data and make predictions on the dataset. By dividing the dataset to train and test set, an independent test set is used to evaluate the results in terms of accuracy, precision, recall, and confusion matrix in this study.

3.3 Reliability and Validity

When a research uses an appropriate tool for measurement, it can be counted as validity and when the result and the experiment can be repeated in other environments and conditions, it would be considered as reliability [29]. Therefore, the results of research highly depend on the data and approaches used. If the same data and implementation are used in another project, the expected outcomes should be almost similar or the same.

In order to investigate the reliability of the research for experiments in the same environment, the succeeding steps need to be followed: first, download 10K tweets in different languages, taken from the Nordic countries. Than It has been followed the markup procedure that is outlined in Section 4.4 and implement thirteen properties as described in Section 4.6. The experiments can be repeated by anyone.

It is important to note that, one issue that could affect the reliability of the project refers to selected random samples of data in the NTS dataset.

Furthermore, in this study, we do experiments with reliable and standard tools.

4. Implementation

This chapter describes our AGT Detector implementation. Section 4.1 describes the tools we used during the experiments. In Section 4.2 the dataset that was used in this project is discussed. Then in Section 4.3, the need of an AGT definition is explained and some examples are described in Subsection 4.3.1. After that, in Section 4.4, the rules for annotating a set of tweets as AGT or HGT are described and Section 4.5 gives more insights in text processing. Subsection 4.5.1 to 4.5.4 is dedicated to the different steps in Natural Language processing in text classifier implementation, such as Predefined Entities, removal stop words and stemming. Section 4.6 presents all properties used from metadata of each tweet in the AGT detector. In Section 4.7 the three different used algorithms in the classifier component are discussed in three subsections 4.7.1 to 4.7.3 A brief introduction about Machine Learning (ML) field is presented and explained in Section 4.8. The last section 4.9 clarifies more about Evaluation performance, with subsection 4.9.1 and 4.9.2, Cross-validation and Recall, Precision and Accuracy respectively.

4.1 Used frameworks

In these experiments, due to the availability of several relevant libraries, we choose the Python programming language. The most used library is Scikit-learn, which is free for Python programming language. Scikit-learn is built upon the SciPy (Scientific Python), which is asked to be installed in advance [30]. This library presents a large variety of supervised and unsupervised machine learning algorithms.

Natural Language Toolkit (NLTK) [31] is another library used in this study for text classifier for the Python programming language that is useful in dealing with natural language [6]. It includes functions such as tokenization, word stemming, and removal of stop words described in more detail in section 4.5.

4.2 Nordic Tweet Stream (NTS) Dataset

The used dataset in this thesis is collected using the same parameters as in the Nordic Tweet Stream (NTS) corpus [8][16]. The NTS uses the free Twitter Streaming API to collect tweets by specifying a geographical region covering the five Nordic countries. This corpus is a real-time monitor corpus designed for sociolinguistic studies of language variability in the Nordic region. In previous studies [9][26][32][33] this dataset has primarily been used to chart the use of English in the area, investigating its grammatical variability, and modelling social networks in multilingual settings [34][35]. The data stream has specific characteristics that influence bot-recognition tools. First, it consists of high velocity data, as we capture nearly 40,000 tweets per day. Second, an additional characteristic is heterogeneity, and we work with a natural language stream that is highly multilingual. To illustrate, in the first 301 days of streaming, there were nearly 70 languages present, but 20 most frequent languages made up of 98.2% of the material [26]. The most frequently used languages were English, Swedish, and Finnish, and the ensuing work focuses on these languages to develop our AGT detector. We used three datasets of: (1) 10,000 English tweets, (2) 10,000 Swedish tweets and (3) 5000 Finnish tweets.

4.3 Defining auto generated tweets AGT

We follow [9][26] and define auto-generated tweets (AGT) as Tweets where all or part of the natural language content is generated automatically by a bot, an application or any other type of program, are defined as AGT. Moreover, by definition, we do not automatically include tweets posted by an application, since we only include those for which the application supplements some natural language content to the tweet. For example, a bot (or an application) that is retweeting a non-AGT is not producing a new AGT since it is not adding any natural language. Thus, AGTs in our definition come in two flavors. Tweets generated from pure bot accounts, such as weather bots, job bots, news bots, etc. The second type consists of tweets generated by applications and programs that are maintained and managed by humans. An opposite of an AGT is HGT (a human-generated tweet). Table 4.1 presents three examples of AGTs and HGTs according to our definition.

Example tweet	Comment	Class
I was out walking 8.02 Km with #something #something http://somewhere.com	This tweet is generated by an app and by adding 'I was out walking' it adds natural language to the tweet.	AGT
New year perfect photo frame!!#something #something @location https://somwhere.com	This tweet is generated by an app but not considered an AGT since it does not add any natural language. The natural language was originally produced in the app by the user.	HGT
Wind 0.3 m/s W, Barometer 1016.0 hPa Rising slowly, Temperature 1.2 C, Rain today 2.7 mm, Humidity 99%	This tweet is generated by a weather bot posting a new forecast every hour.	AGT

Table 4.1: Examples of AGTs and non-AGTs

4.3.1 Examples of AGTs

- Retweets: A retweet of an AGT is an AGT, a retweet of a HGT is a HGT.
- A bot publishing famous quotes is an AGT.
- A Social Media Management tool scraping a news website and posting a tweet with the title as the text and the article as attached media is AGT.
- News is also an AGT also in the case the article is posted (copied) manually using an iPad.
- News is a HGT if the publisher uses a unique summary (rather than the title) of the article content as a text message.

4.4 Annotating Tweet as AGT or HGT

The manual AGT/HGT annotation is made by persons very knowledgeable (native preferably) in the language they handle, and also familiar with the current situation (what is going on) in the corresponding country. For example, it helps to be aware of major events (sports, politics, etc.) going on in the country at that time.

In addition to the AGT definition and a lot of examples and recommendations, each annotator (or group of) is giving an excel sheet which, for each tweet, contains the username, the actual Twitter text and a web link of type

`https://twitter.com/anyuser//status/930432195436609536`

giving the annotator a chance to see the tweet in context, among other tweets published by the same user. The web link gives the annotator a very good understanding of what type of user that was publishing the tweet.

Finally, assume that we are interested in getting (say) 5000 manually labeled tweets. The annotator will in that case get an excel sheet with 5000+ tweets and be asked to annotate the first 5000 tweets belonging to still active users. A user being no longer active is the only reason for a tweet not being annotated.

4.5 Text Classifier

Classifiers based on textual content have successfully been used in several studies to detect spam and bots [9][10][36]. Text classifiers plays an important role for spam and bots detection on Twitter and a few studies have addressed the problem already [3][4][12]. We believe that a classifier based on textual content could perform as a good indicator of AGTs due to our data containing several bots (e.g. weather bots, job bots, etc.) that constantly post tweet texts with very similar content. Also, several AGTs posted by humans through applications share textual patterns that can be exploited by a text-based classifier. Moreover, Lundberg et al. [9] used the textual content of a tweet as part of a feature set for the purpose of detecting AGTs.

In our study, we apply traditional Natural Language Processing (NLP) techniques to the textual content of a tweet using a bag-of-words model based on the most frequent words in our data set. The output of the Text classifier is the estimated probability that a certain tweet is an AGT. Henceforth, this is called a tweet's *AGT text probability*.

4.5.1 Natural Language Processing

NLP is a technique that enables a machine to process a natural language (language used between humans) like English and turn all into the things that a human can do. In short, NLP helps in automating things [37].

In this thesis, three concepts of NLP have been used through NLTK to the content of tweets to normalize in the classification phase. The entire work uses some techniques described in the following subsections. A summary of the used Natural Language Process presented in Figure 4.1.

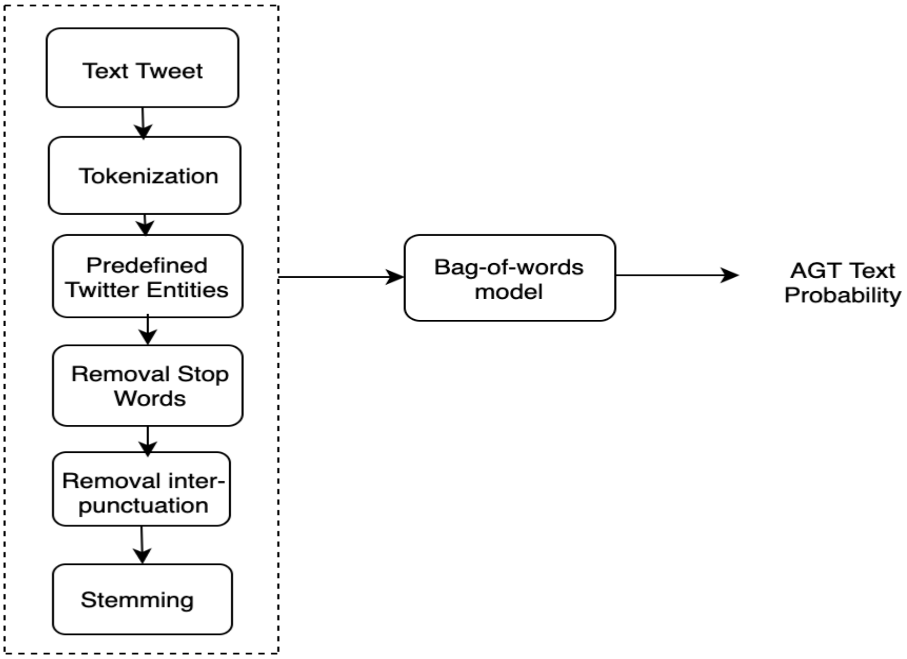

Figure 4.1: The pipeline for NLP implementation

4.5.2 Tokenization and predefined Twitter Entities

A tokenizer or tokenization function breaks up a text into a list of ”tokens”. In this project, we use NLTK's tokenizers function `TweetTokenizer()`. Besides the text content of tweet, the input also includes predefined Twitter entities such as hashtag, user mentions and URLs. For example, we have a corpus of tweets that have # and @ in order to mention hashtags and users. All these entities are replaced by a token which is treated as regular words by the classifier:

- xhashtagx
- xuserx
- xnumberx
- xurlx

It should also be noted that emojis are kept as is (i.e. each emoji is interpreted as a single word), as we believe that they might carry valuable information for the classifier.

4.5.3 Stop Word Removal

Stop words are words that are very common in the dataset and are therefore not very likely to give away a lot of information. Examples of stop words, that can be removed in English are "a, an, the, if, for" and so on. Silva and et al. [38] also pointed out that stop word removal can have a very positive impact on the recall of the resulting classifier. In this project, we only used a general-purpose stop words remover and removed all inter-punctuation in our tweets.

4.5.4 Stemming

Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form generally a written word form. For example, ran, running, runs are all derived from the word ”run". Commonly used stemming algorithms for the English language are ”Porter" and "Snowball" Algorithms. In this study, we used NLTK for stemming. We use "SnowballStemmer" [16] for all three languages in order to convert individual words to their stem. A similar approach is used in [39][40].

4.6 Tweet Properties used in this classification

The input to the AGT Detector consists of thirteen tweet properties attaining numerical and nominal values that can be computed directly from the tweet metadata. These properties are selected as indicators that can be used (one at a time, or in combination) to identify non-human behavior. For instance, one should expect that humans have more followers than bots, or that AGTs tend to contain more URLs. Many of the properties used in this classifier have been discussed in [9][10] [11][14][26] [41]. You can see the summary in Figure 4.2:

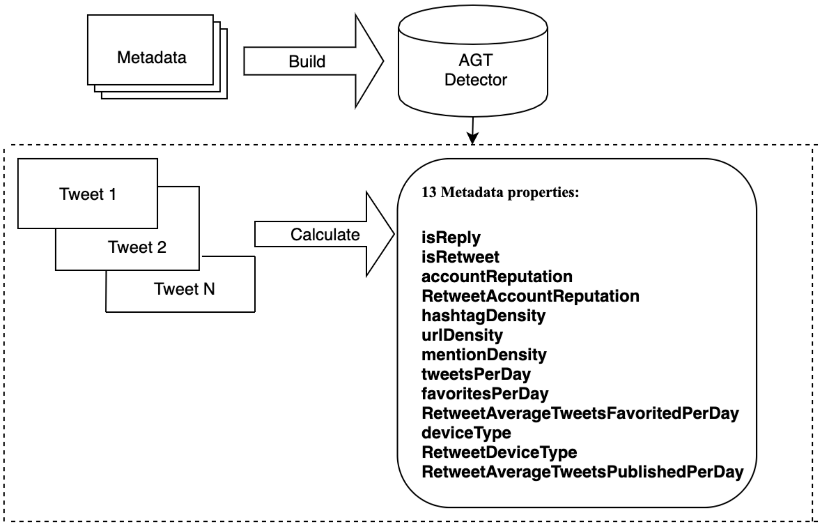

Figure 4.2: AGT Detector implementation summary

The thirteen used properties are:

- **isReply** - boolean indicating if the tweet is a reply
- **isRetweet** - boolean indicating if the tweet is a retweet
- **accountReputation** - number of followers divided by the number of friends and followers
- **RetweetAccountReputation -** number of followers divided by the number of friends and followers for a retweeted one
- **hashtagDensity** - number of hashtags divided by the total number of words in the tweet
- **urlDensity** - number of URL divided by the total number of words in the tweet
- **mentionDensity** - number of mention entities, divided by the total number of words in the tweet
- **tweetsPerDay** - total number of user's tweets divided by account age in days
- **favoritesPerDay**- number of tweets favorited by user divided by account age
- **RetweetAverageTweetsFavoritedPerDay -** number of tweets favorited by retweeted user divided by account age in days
- **deviceType**- nominal variable based on the type of source used to post the tweet:
    1. mobile: Twitter for Iphone, Twitter for Android etc.
    2. web: Twitter Web Client, Tweetbot for Mac etc.
    3. app: Instagram, Tumblr, Foursquared etc.

4. SMM: Falcon Social Media Management, TweetDeck, dlvr.it, etc.
5. bot: Trendsmap Alerting, SpotifyNowPlaying, etc.
6. other: newly observed not classified sources.

- **RetweetDeviceType** - nominal variable based on the type of source used to post the retweet. We use the same type values as deviceType.
- **RetweetAverageTweetsPublishedPerDay -** total number of user's retweets divided by account age in days.

The tweet metadata contains a `source` attribute that identifies what type of an app, a program or a device was used to post the tweet or retweet. We manually classified the 150 most frequently used sources in our training set in one of the five categories, (1) - (5), defined in the device Type attribute. These 150 sources cover about 97 % of all sources in the training set, while the remaining (unlabeled) sources were automatically classified as unknown. The device type SMM stands for Social Media Management. That is, tools for managing content on multiple accounts on social networks.

Device Type	Swedish	Finnish	English
Mobile (1)	65.70 (1.56 %)	31.60 (18.11 %)	51.23 (0.35 %)
Web (2)	21.64 (21.63%)	11.16 (15.47 %)	18.7 (0.35 %)
App (3)	2.74 (4.07 %)	0.39 (3.58 %)	20.20 (57.26 %)
Smm (4)	6.06 (39.02 %)	2.35 (10.75 %)	0.27 (0.62 %)
Bot (5)	0.19 (0 %)	0.05 (0 %)	9.26 (41.04 %)
Unknown (6)	3.67 (33.69 %)	3.45 (52.07 %)	0.32 (0.35 %)

Table 4.2: Percentage of used types and percentage of AGTs in each type

The deviceType property turns out to be the backbone of our AGT classification. Table 4.2 shows the different device types that were used in the different datasets and what percentage of AGTs we find in each type. For example, 65.70% of all tweets in the Swedish dataset were posted using device type 1 (mobile) and only 1.56% of these tweets are labelled as AGTs. A noteworthy feature is that device types 1 (mobile) and 2 (web) dominate. They are used to post about 85% of all tweets. Notice also that the percentage of AGTs for these tweets (especially for type 1) is very low (0.35-18.11% depending on language). Hence, a tweet of type 1 or 2 is very likely to be a HGT. This leaves four device types to be problematic. They are 3 (app), 4 (smm), 5 (bot) and 6 (unknown), all of which can be either AGTs or HGTs. For these types, additional information drawn from the other properties is needed to make a classification.

4.7 Machine Learning

Generally, Machine learning is considered as a subfield of Artificial Intelligence which has strong links to statistics, probability theory and optimization [42][43]. It learns from former experience to present computational methods in order to improve performance in performing some actions. The performance is measured by how well the actions indicate the correct items, such as decisions or predictions.

Machine learning is presenting practical and accurate prediction algorithms and models. In general, it is split into three broad categories: (1) supervised learning, (2) unsupervised learning and (3) reinforcement learning. This thesis focuses on supervised learning which uses labeled examples to make predictions. Furthermore, regarding the designed output of models, machine learning can be classified adversely. Three practically common ones are: (1) regression, (2) clustering and (3) classification [43]. For example, in OSNs, classification can be used for news articles to be classified into classes such as sports, weather, business or politics. In this study, we used a supervised learning method in a classification approach to assign a class to each sample of NTS dataset.

4.8 Classifiers Component

Rather than evaluating all applicable machine learning algorithms, we tested a few models and soon realized that tree-based models often outperformed the other algorithms available. Similar results (i.e. tree-based models are suitable for tweet classification) are reported in [9][10][41][44]. In this section, we decided to have a brief description of three used algorithms in the thesis, such as (1) Random Forest (RF), (2) Decision Tree (DT) and (3) Support Vector Machine (SVM). They have all been used to handle similar problems [9][10][41][44] and they represent two different categories of machine learning models.

4.8.1 Decision Tree

One of the simplest algorithms for humans to intuitively understand is called Decision Tree. It develops by categorizing samples based on rules of the type "if x then y else z". Decision trees can be used both for classification and regression approaches. There are several algorithms for building decision trees from data, one of the most well-known ones is the C4.5 algorithm by Quinlan [46]. Given a set of training data the algorithm returns a decision tree in which each split is made to increase the information gain by use of entropy [47].

4.8.2 Random Forest

Random Forest [45] is an ensemble classifier that was introduced in 2001, which utilizes decorrelated decision trees to produce a consensus model. It is built by combining several decision trees, as the name indicates. In Random Forest, each node is split by using the best split between a subset of randomly chosen features. It has shown to be effective and has good variance reduction. For a more detailed description see [45].

4.8.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) was introduced in 1995 [48]. The idea behind SVM is to represent the different data points in space and then find the hyperplane with maximum-margin that separates the two different classes, AGT and HGT. The hyperplane namely maximizes the distance from the hyperplane to the nearest spots of the different classes. For instance, in a two-dimensional problem, the maximum-margin hyperplane would be a line.

Nowadays, linearly separable data is not frequently used, and SVMs apply a kernel function which is used to map the non-linearly separable data into a feature space where the data can be separable linearly [49]. There are several kernel functions that are most common, such as, (1) Linear, (2) Polynomial, (3) Radial Basis Function (RBF) and (4) Sigmoid. To capture any non-linearity in our data we apply a RBF/Gaussian kernel which is a weighted linear combination of the kernel function calculated between a data point and each of the support vectors [50].

4.9 Evaluation Performance

In machine learning, a standard approach for evaluating classifiers is to split the labeled data into both a training and test set. In this section, we explain the principle and the measuring performance used in this study for the evaluation of classifiers.

4.9.1 K-Fold Cross-Validation

One of the traditional ways of evaluating the performance of a classifier is K-Fold Cross-Validation. For n-fold cross-validation, the training set is divided into n disjoint sets. The classifier is then trained using $n-1$ of these sets and tested with some choice of metric against the nth set. This procedure is repeated until all n sets are tested. In this way, all instances are used for training on $n-1$ occurrences and tested for in 1 occurrence. This is very common to use ten folds, and in this project we have used ten folds for our first experiment, as well [16][10][51].

4.9.2 Precision and Recall and Accuracy

In this study, the metrics we use to evaluate the results of the classifier are precision, recall and accuracy. Although there are various metrics in machine learning to assess the performance of the model, all measures are defined based on four features that are obtained while assessing our model on test set: (1) true positives, (2) true negatives, (3) false positives and (4) false negatives. These four entries construct a confusion matrix, demonstrated in below Table 4.3.

		Predicated Class	
		Positive	Negative
Actual Class	Positive	true positives (tp)	false negatives (fn)
	Negative	false positives (fp)	true negatives (tn)

Table 4.3: An example of a confusion matrix

Precision is the positive predictive value, i.e. the proportion of correctly classified instances among the total number of instances classified as AGTs. Precision can be calculated by using the formula from the Equation (4-1):

$$P = tp\ /(tp + fp) \quad \text{4-1)}$$

Recall or true positive rate is the proportion of correctly classified instances among the total number of AGTs in the ground truth set. Recall can be calculated by using the formula from the Equation 4-2):

$$R = tp\ /(tp + fn) \quad \text{4-2)}$$

Another metric to measure performance is accuracy, the fraction of correctly classified instances. Accuracy can be calculated by using formula from the Equation 4-3):

$$Accuracy = (tp + tn)\ /(tp + tn + fp + fn) \quad \text{4-3)}$$

5. Evaluation and Discussion

This chapter is devoted to our experiments and evaluation. We divided this chapter into three sections. In Section 5.1 we will explain our three evaluation Scenarios. The first experiments will be presented in Section 5.2 where we will establish our Monolingual approach as a baseline. A summary and conclusion of the first experiment will be discussed in Subsection 5.2.1. Later on, Section 5.3 we will continue our Multilingual experiment by mixing all training data and test data as our best approach. Finally, in subsection 5.4 we present our last experiments as Bilingual experiment, the classifiers were trained on a dataset with two languages (e.g. Swedish and Finnish) and tested on a third language (e.g. English). We consider all permutations in this experiment.

5.1 Evaluation Scenarios

Our idea is to use single language results as a baseline in this study. That is, we train and test a model in the same language. This is the optimal case and we therefore consider the monolingual case as a baseline to which other cases will be compared. Later on, we mix all datasets as a Multilingual experiment and train on 80% and test on 20% of the mixed dataset to check the accuracy of mixing language. We finally look at how classifiers trained on a finite set of languages would react to tweets written in new unseen language by using two languages as a train set and test on unseen new language. This is expected to be the hardest case and the actual test of whether a language independent detector is possible in practice since our objective is to develop a light language independent application for AGT detection, an application that would work on any language no matter if they are used in the training phase or not.

5.2 First Experiment: Monolingual

In our first experiment we train and test on the same language. As you can see in Figure 5.1, in order to have a baseline in our study, we add text-based classifiers in addition to the thirteen metadata properties to train supervised machine learning models to recognize the AGTs.

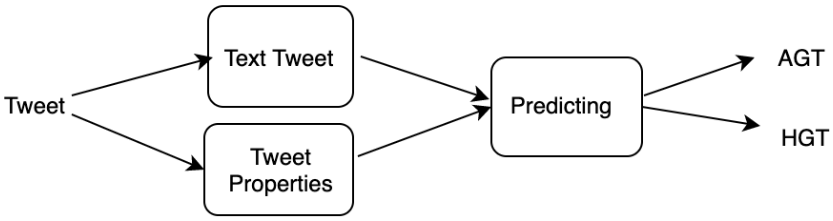

Figure 5.1: Overview of the Baseline Experiment

From our NTS data set, we took a random sample of 10K tweets in English, 10K tweets in Swedish and 5K tweets in Finnish. Using the textual content of these and the rules for annotating a tweet as it is explained in Section 4.4 we could annotate them as belonging to either class AGT (3,419, 13.6 %) or class HGT (21581, 86.3 %). Moreover, the number of tweets labeled as AGT varies between languages: English (22.5%), Swedish (6.3%), and Finnish (10.6%).
In the first phase for monolingual AGT detection, we calculate text probability for each tweet from text classifier, as it is indicated in Table 5.1.

Tweet	AGT Text Probability
I'm at `xuserx` in Kuopio, Eastern Finland w/ `xuserx xurlx`	68% [AGT]
I just finished `xnumberx` of doing circuit training with `xhashtagx xhashtagx xurlx`	71 % [AGT]
Discover hotels around somewhere in Norway from `xnumberx` EUR per night: `xurlx`	47 % [AGT]

Table 5.1: Examples of tweets and Text Classifier results

These three examples of AGT texts and corresponding AGT probabilities were also used in Lundberg et al. [9] article using Weka with slightly lower probabilities 39%, 47 % and 7 % respectively. It indicates that different tools do not perform identically, even though we have used the same dataset as was used in the article.
In the second phase, we implement thirteen metadata properties (see Section 4.6) for each single tweet in our data set, as it is shown in Table 5.2.

Attribute	Value
tweet_id	987198797309775872
isReply	0
isRetweet	1
accountReputation	0.712814
hashtagDensity	0.00
urlDensity	0.00
mentionDensity	0.066667

tweetsPerDay	7.168605
favoritesPerDay	1.487645
deviceType	1.0
re_tweetsPerDay	0.948612
re_favoritesPerDay	0.384471
re_accountReputation	0.940991
re_deviceType	2.0
text_probability	0.6667

Table 5.2: Example of calculated property values for one Finnish tweet

It is worth to mention that in all monolingual experiments we did not do any fine-tuning and we used the default settings in Scikit-learn. The AGT detector results for monolingual experiments in the English language in terms of precision, recall, accuracy and confusion matrix for each evaluated classifier using a 10-fold cross validation are illustrated in Table 5.3 and Table 5.4 respectively.

	RF	DT	SVM
Recall	0.999	0.999	0.993
Precision	0.999	0.999	0.998
Accuracy	0.999	0.999	0.998

Table 5.3: Precision, Recall and Accuracy for **monolingual** for **English** dataset

	Predicted		
Actual		**AGT**	**HGT**
	AGT	7748	1
	HGT	1	2250

Table 5.4: Confusion Matrix for the best model **Random Forest** for **English** dataset

As you can see, the two tree-based models performed very well, and we have only 2 misclassifications. The results of the Random Forest model (RF) and Decision Tree (DT) stand out as the best in this setting. Support Vector Machine (SVM) performs slightly worse on Recall. RF correctly classified 7748 of the 7749 AGTs in the training set. We have only 2 errors, one in false negatives, another in false positive.

The AGT detector results for monolingual experiments in the Swedish language in terms of precision, recall, accuracy and confusion matrix are illustrated in Table 5.5 and Table 5.6 respectively.

	RF	DT	SVM
Recall	0.993	0.995	0.981
Precision	0.995	0.989	0.946
Accuracy	0.999	0.998	0.949

Table 5.5: Precision, Recall and Accuracy for **monolingual** for **Swedish** dataset

	Predicted		
Actual		**AGT**	**HGT**
	AGT	9358	4
	HGT	5	633

Table 5.6: Confusion Matrix for the best model **Random Forest** for **Swedish** dataset

In Swedish language Random Forest (RF) stands out as the best algorithm once again. However, from the Table 5.6 it can be seen that nine misclassifications happened. Table 5.5 shows the result of Swedish dataset which accuracy for Swedish data set is (99.9%), with slightly worse Recall compared to the English language.

The AGT detector results for monolingual experiments in the Finnish language in terms of precision, recall, accuracy and confusion matrix are illustrated in Table 5.8 and Table 5.8Table 5.8 respectively.

	RF	DT	SVM
Recall	0.990	0.984	0.956
Precision	0.990	0.981	0.977
Accuracy	0.998	0.996	0.993

Table 5.7: Precision, Recall and Accuracy for **monolingual** for **Finnish** dataset

	Predicted		
Actual		**AGT**	**HGT**
	AGT	4464	6
	HGT	5	525

Table 5.8: Confusion Matrix for the best model **Random Forest** for **Finnish** dataset

From Table 5.8, it can be concluded that the highest misclassification number among all three languages occurred when working with the Finnish tweets. However, an accuracy of 99.8% is still very good.

5.2.1 First Experiment's Summary

	Accuracy	Precision	Recall
English	0.999	0.999	0.999
Swedish	0. 999	0.995	0.993
Finnish	0.998	0.990	0.990

Table 5.9: Precision, Recall and Accuracy for **RF** classifiers trained and tested using monolingual datasets.

The results presented in Table 5.9 are more accurate than similar Monolingual AGT detection results presented in [9][26]. The major difference is that we have used a different tool (Scikit-learn instead of Weka) and a slightly larger number of metadata properties. Apart from that we do not have an explanation as to why the Scikit-learn result is different than Weka.

All models performed very well, and the misclassifications are few. The results of Random Forest model (RF) stands out as the best in this setting.

These results show that English AGTs are rather easy to detect (accuracy 99.9 %). A more detailed study of the English dataset shows that a large portion of the English AGTs are posted by bot accounts (e.g. weather bots) that are easily identified by the device type (5, bot) used to generate them. The Swedish tweet dataset has the lowest AGT ratio (6.3%), the most significant characteristic of the Swedish AGTs is that many are posted by SMM tools (type 4) that companies/organization used to promote news published on their own websites.

Detecting AGTs in the Finnish dataset turns out to be the most difficult (i.e. recall being the lowest). Whereas Swedish newspapers often use SMM tools to promote news on Twitter, it looks like that many Finnish newspapers in our sample often take a more hands-on approach and manually 'share' their newspaper web content on the Twitter account. Therefore, detecting this behavior automatically is difficult since the used device type often is of types 1 or 2 that we usually associate with HGTs, such as 'Twitter for Iphone' or 'Twitter Web Client'.

The fact that each dataset (language) has its own characteristics indicates that a classifier trained on a certain language is expected to be less accurate when used to detect AGTs in another language.

5.3 Second Experiment: Multilingual

In the second experiment, we present the results of training an AGT detector using a Multilingual training and test set. We combine all three languages datasets into one random set of multilingual data, then we train on 80 % and test on 20 %. It is worth to mention that in order to do a proper evaluation we remove test set from Training set. A proper evaluation should be based on a test set, not being used in training set as standard approach.

We select 5000 random tweets from each dataset as test set to make sure that we have all three languages in correct proportion, which includes approximately 2000 English data set, 2000 Swedish data set and 1000 Finnish data set. Table 5.10 shows the details about test dataset for separate language.

Language	Label	Count
English	HGT	1223
	AGT	978
Finnish	HGT	736
	AGT	237
Swedish	HGT	1541
	AGT	285

Table 5.10: Test data regarding separate language in Multilingual experiment

In the training phase, we use a total of 20,000 tweets in all three languages. The results in terms of accuracy, precision and recall for each evaluated classifier are shown in Table 5.11 and the Confusion Matrix for the best model (RF) in Table 5.12.

	DT	RF	SVM
Recall	0.824	0.888	0.886
Precision	0.960	0.952	0.859
Accuracy	0.937	0.953	0.922

Table 5.11: Accuracy, Precision and Recall for training phase of Multilingual experiment for the classifier model.

	Predicted		
Actual		**AGT**	**HGT**
	AGT	1332	168
	HGT	66	3434

Table 5.12: Confusion Matrix for multilingual classifier for the best model **(RF)**

In this experiment all three models DT, RF and SVM utilize the Grid Search to fine-tune algorithms in Scikit-learn to improve the performance of classifier. “Grid search is the process of performing hyper parameter tuning in order to determine the optimal values for a given model” [52]. In this study the Decision tree classifier and Random Forest were tuned by “`max_depth=20`”, “`min_samples_leaf=2`” and “`min_sample_split=4`”. The highest performance in SVM is achieved in case of taking some parameters into account by tuning factors such as “`kernel=rbf`”, “`gamma=scale`” and

“`C=90`”. In other words, SVM is performing well using a radial based function (RBF kernel), and an optimal combination of the penalty factor C with the aim of achieving the highest prediction accuracy.

Random Forest (RF) has the best results (the highest accuracy 0.953), followed by Decision Tree (0.937) and SVM (0.922). As it is shown in Confusion Matrix in Table 5.12, Random Forest classified 1332 of the 1500 AGTs in the training dataset.

Similar results for monolingual datasets (best model is Random Forest) are presented in [9][26][10]. Note that the results presented here are not as accurate as the monolingual results presented in Table 5.9. Hence, while working with multilingual data streams may have benefits for sociolinguistic research, adding new languages, and not including the actual Twitter text, come with a price.

In the coming tables (Table 5.13, Table 5.14 and Table 5.15) we indicate a separate Confusion Matrix for separate languages in this multilingual experiment. As the results show, similarly to the Monolingual results in Section 5.2, English language tweets are rather easy to classify, since we have less misclassified tweet in English dataset compared to Swedish and Finnish languages. Obviously, it could be because of higher English AGT rate in our NTS dataset.

	Predicted		
Actual		**AGT**	**HGT**
	AGT	941	37
	HGT	4	1219

Table 5.13: Confusion Matrix for **RF** for **English** language

	Predicted		
Actual		**AGT**	**HGT**
	AGT	156	81
	HGT	19	717

Table 5.14: Confusion Matrix for **RF** for **Finnish** language

	Predicted		
Actual		**AGT**	**HGT**
	AGT	235	50
	HGT	43	1498

Table 5.15: Confusion Matrix for **RF** for **Swedish** language

5.3.1 Second Experiment's Summary

In comparison to results related to the first experiment, the first thing to notice is that similar pattern of baseline experiments is presented in multilingual one, as well. We have more errors in Finnish language, slightly less errors in Swedish language and much fewer errors in English language. As we expected, we can see the same distribution in the multilingual dataset, too.

The second thing to notice, based on the results shown in the tables below, is that English AGTs are rather easy to detect compared to Swedish and Finnish language.

In comparison of the baseline classifier result, which it builds model based on thirteen properties and text classifier, the share of correctly classified AGTs for two classifiers are: Baseline (99.9 %), Multilingual experiment (88.8 %). Therefore, adding languages together, and not taking the actual text into account, comes with an 11.1 % less accuracy.

Comparison with different languages in Multilingual dataset shows different things from results in below tables. Surprisingly, after fine tuning, SVM has the highest proportion of correctly identified AGTs (best Recall) in English language (98.1 %) compared to RF (96.2 %) and Decision Tree (94.2 %). However, Random Forest still has the highest Accuracy in English language. Similar to the baseline results, the Finnish dataset turns out to be the most difficult one to classify. It is also worth mentioning that Finnish language has worst Recall (55.6 %) in Decision Tree classifier among other classifiers in multilingual dataset. Swedish dataset has the lowest Precision in SVM classifier.

Random Forest	Precision	Recall	Accuracy
English	0.995	0.962	0.981
Swedish	0.845	0.824	0.949
Finnish	0.981	0.658	0.897

Table 5.16: summary of Accuracy, Precision and Recall for Multilingual classifier for separate language in **RF** algorithm by each language.

Decision Tree	Precision	Recall	Accuracy
English	0.990	0.942	0.970
Swedish	0.863	0.642	0.928
Finnish	0.910	0.556	0.878

Table 5.17: summary of Accuracy, Precision and Recall for Multilingual classifier for separate language in **DT** algorithm by each language.

Support Vector Machine	Precision	Recall	Accuracy
English	0.944	0.981	0.966
Swedish	0.647	0.733	0.895
Finnish	0.774	0.679	0.873

Table 5.18: summary of Accuracy, Precision and Recall for Multilingual classifier for separate language in **SVM** algorithm by each language.

5.4 Third Experiment: Bilingual training (Unseen dataset)

In the first experiment, we train and test individual languages taking also the actual text into account, in the second experiment we combine all three languages together, now in the last experiment we present the result for training in two languages and test using an unseen language dataset. This simulates a real-world scenario where the classifier is asked to classify a tweet written in a language that was not a part of the training set. We divided the experiments into three subsections. We start in 5.4.1 by using Swedish and Finnish in training phase and test the classifier on English language dataset. In Section 5.4.2 we use English and Finnish as training sets and test on Swedish language. Finally, Section 5.4.3 presents the last permutation using Swedish and English as training set and test in Finnish as test set.

5.4.1 Bilingual training set Swedish and Finnish

In this experiment, we use the Swedish and Finnish datasets in training phase, a total of 15000 tweets. Our training dataset include 13832 HGT and 1168 AGT. The results of applying the classifiers were trained on a dataset with two languages (Swedish and Finnish), and tested on the unseen English dataset in terms of accuracy, precision and recall and the Confusion Matrix for the best model SVM are shown in Table 5.19 and Table 5.20 respectively.

	RF	DT	SVM
Recall	0.605	0.414	0.990
Precision	0.925	0.909	0.749
Accuracy	0.903	0.859	0.923

Table 5.19: Accuracy, Precision and Recall for classifier train on Swedish and Finnish datasets and test using the unseen **English** dataset.

	Predicted		
Actual		**AGT**	**HGT**
	AGT	2229	22
	HGT	744	7005

Table 5.20: Confusion Matrix for **(SVM)**

We have used the same fine-tuning of the parameters (Grid Search) as for the multilingual experiment in bilingual experiments. The interesting point is that Support Vector Machine has the best result (accuracy 92.3 %) followed by Random Forest (90.3 %) and Decision Tree (85.9 %). Moreover, SVM with 38.5 % difference from Random Forest (60.5 %) has the highest recall (99.0 %), although it has slightly less precision compared to Random Forest. This is corroborated after comparing the Confusion Matrix for RF and SVM, it is found that SVM has less misclassified tweets compared to RF. This indicates over-fitting, so that the complex models produced by RF and DT are far too specialized on the training languages Swedish and Finnish, and that the SVM model focusing on only the essential information better adapts to handling a new language. Although it might not take all special cases into account, it still performs better.

5.4.2 Bilingual training set English and Finnish

In this training phase, we use the English and Finnish datasets, a total of 15000 tweets. Our Training dataset includes 12219 HGT and 2781 AGT. The results of applying the classifiers were trained on a dataset with two languages (English and Finnish), and test on the unseen Swedish dataset in terms of accuracy, precision and recall and the Confusion Matrix for the best model Random Forest are shown in Table 5.21and Table 5.22 respectively.

	RF	DT	SVM
Recall	0.711	0.688	0.744
Precision	0.518	0.244	0.228
Accuracy	0.939	0.742	0.823

Table 5.21: Accuracy, Precision and Recall for classifier train on English and Finnish datasets and test using the unseen **Swedish** dataset.

	Predicted		
Actual		**AGT**	**HGT**
	AGT	454	184
	HGT	422	8940

Table 5.22: Confusion Matrix for (**SVM)**

As it is shown in Table 5.21, we have similar results as in subsection 5.4.2. That is, the SVM with 3.3 % difference from Random Forest (71.1 %) has the highest Recall (74.4 %). However, the Random Forest algorithm has the higher accuracy (93.9 %) to detect AGTs compared to SVM (82.3 %) and Decision Tree (74.2 %). Since Recall is the most important metric to detect AGTs in our experiments, Support Vector Machine algorithm occupies the best position compared to the other two algorithms. Although, once again it might not take all special cases into account.

5.4.3 Bilingual training set English and Swedish

In this training phase, we use the English and Swedish datasets, a total of 20000 tweets. Our Training dataset includes 17111 HGT and 2889 AGT. The results of applying the classifiers were trained on a dataset with two languages (English and Swedish), and test on an unseen Finnish dataset in terms of accuracy, precision and recall and the Confusion Matrix for the best model Support Vector Machine are shown in Table 5.23 and Table 5.24 respectively.

	RF	DT	SVM
Recall	0.620	0.549	0.658
Precision	0.746	0.765	0.511
Accuracy	0.937	0.934	0.897

Table 5.23: Accuracy, Precision and Recall for classifier train on English and Finnish datasets and test using the unseen **Finnish** dataset.

	Predicted		
Actual		**AGT**	**HGT**
	AGT	329	201
	HGT	112	4358

Table 5.24: Confusion Matrix for (**SVM)**

Similar results are presented here once again, the SVM algorithm (65.8 %), with almost 4 % difference compared to Random Forest (62.0 %), has the highest Recall. However, the Random Forest algorithm has the best accuracy (93.7 %) to detect AGTs compared to SVM (89.7 %) and Decision Tree (93.4 %). It seems that still SVM algorithm occupies the best position compared to the other two algorithms.

5.4.4 Third Experiment's Summary

The main purpose of the Bilingual experiments has been to evaluate how a language independent AGT detector reacts when trying to classify a tweet in a new unseen language, a tweet written in a language that is not a part of the training data. This is a crucial step since, in practice, in a real-world scenario, we cannot expect the AGT detector to have been trained on all possible languages.

Support Vector Machine	Precision	Recall	Accuracy
(Train: SW, FI) Test: **English**	0.749	0.990	0.923
(Train: EN, FI) Test: **Swedish**	0.228	0.744	0.823
(Train: EN, SW) Test: **Finnish**	0.511	0.658	0.897

Table 5.25: Accuracy, Precision and Recall for best model ***(SVM)*** *results on each permutation on an unseen language*

We explored three different combinations by training the classifier using two languages tweets and evaluated the approach using a third unseen language. We learned that the idea of language independent Twitter bot is not as straightforward as we wanted it to be. It should be taken into account that English data is rather strange compared to Finnish and Swedish data, they have a much higher rate of AGTs which is not common on Swedish and Finnish.

The results also show surprisingly that the SVM model outperformed other models when applied to an unseen language, although with lower precision. This is a surprise since it was repeatedly performing worse than Random Forest in previous experiments. In all three combinations, SVM with almost 3 % difference compared to Random Forest, turns out to be quite good at detecting AGTs. It shows good results (99.0 % recall) when it comes to test unseen English dataset with only 0.3 % difference with monolingual English results (99.3 % recall) in Table 5.3.

It should be noted that the final experiment from a machine learning perspective is rather odd since we evaluate the approach by using a test set distinctly different (another language) than the training data. Hence, ordinary machine learning strategies based on a scenario where test and training data are taken from the same sample do not really apply. This indicates that both Random Forest and Decision Trees overfit with respect to the languages used in the training, whereas SVM seems to better adapt to this new scenario. Similar result was presented in Lundberg et al. [26] when they reported better results with a smaller under fitted model, in their case with a smaller decision tree, when they are testing using an unseen language.

As it is shown in Table 5.25, The SVM recall varies between 0.658 (unseen Finnish) and 0.744 (unseen Swedish) and 0.990 (unseen English). For instance, in the worst-case scenario (unseen Finnish), in a set of 5000 tweets it correctly identifies 329 (out of 530) AGTs, and it also classifies misclassifies 112 HGTs as AGTs (see Table 5.24). It indicates that the error rate is still too high for us to recommend this to be used in real world application. The percentage of AGTs that is identified is too low, and also, the percentage of HGTs that is classified as AGTs is too high. So, that we might remove a number of actual

HGTs is probably not that problematic since we have so many tweets, the major problem is that we are not removing all AGTs and that the resulting dataset is still polluted with AGTs that might skew any subsequent linguistic analysis.

In conclusion, it will far from filter out all the AGTs from their datasets (e.g. [26]), it will only remove approximately 60-70% of AGTs, but still better than nothing. However, it depends on exactly what type of experiments subsequent analysis we are interested in. It might consider as good enough because we are removing most of the AGTs.

6. Conclusion

This is the concluding chapter and is divided into two subsections. The conclusion of this thesis is present in Section 6.1 whereas Section 6.2 is dedicated to recommendations for future work.

6.1 Conclusion

In this study our goal has been to build a language independent classifier for detecting auto generated tweets (AGTs) written in any language. Our approach is to use machine learning to identify AGTs in the NTS data stream (as it explained in Section 4.2) using only thirteen properties (see Section 4.6) that can be computed from the metadata that comes with every tweet. We divided our experiment into three parts.

In the first part, the monolingual case, we train and test our model based on the same language. In this experiment, in addition to the thirteen properties, we also take the actual twitter text into account when we classify a tweet. We consider the first experiment as a baseline case to which other approaches, not using Twitter text, will be compared. The text based classifier performs very well with only 2 misclassified tweets in English data set with highest accuracy of 99.9 %. With respect to objective 2 in monolingual classifier, we observed that the results illustrated that Random Forest performs slightly better than other models. Detecting AGTs in English data set with 0.999 recall turns out to be rather easy, then followed by Swedish data set with 0.993 and Finnish data set with 0.990 as the most difficult one.

As the main purpose of this thesis is detecting AGTs written in any language, we continued our second experiment by combining all three languages together to verify the accuracy of mixing languages to classifier. With respect to objective 2 in multilingual classifier we observed that, Random Forest has once again the best results with 0.953 accuracy, followed by Decision Tree with 0.937 and SVM with 0.922. Although less precise than the monolingual case, the results follow the same pattern; we have more errors in Finnish language, slightly less errors in Swedish language and much fewer errors in English language.

With respect to Research Question 2, in comparison of the baseline classifier result, which it builds model based on thirteen properties and text classifier, the share of correctly classified AGTs for two classifiers are: Baseline (99.9 %), Multilingual experiment (88.8 %). Therefore, combining languages comes with an 11.1 % less accuracy.

The main goal is to develop a classifier that is, in principle, language independent since it does not use the actual Twitter text but only relies on language and country independent metadata that are available in each tweet. Therefore, the application should work on any language no matter if they are used in the training phase. In our last experiment we present results which were trained on a dataset with two languages and applied on a third unseen language/dataset. The results of three permutation on the three most frequently used languages of NTS shows significantly worse results when trying to classify tweets from a new unseen language.

Regarding our objective 3, we observed that the difference between monolingual and bilingual classifiers is noteworthy. The share of correctly classified AGTs for bilingual training and test on an unseen language for the different language are: English: 99.9 %, Swedish: 74.4 %, Finnish: 65.8 % which compared to monolingual experiment that train and test in the same languages are: English: 99.9 %, Swedish 99.9 % and Finnish 99.8 %. Finnish language with 34.1 % difference is the most significant one.

Hence, the idea of language independent Twitter bot detection is not as straightforward as we expected to be. As is discussed in subsection 5.4.4, this AGT detector probably is not sufficiently accurate for cross-disciplinary research projects that would like to filter out AGTs from datasets when only identifying 60-70 % of AGTs.

Another thing to notice is that fine-tuning the machine learning hyperparameters has a great impact in all results. SVM performed very well once we fine-tuned it. The fine-tuned model of SVM outperformed the other models in the final experiment and was slightly better than the tree-based models in detecting AGTs. In other words, when we are applying it to a new language, it is less over-fitting and we get better results. However, when it comes to tree-based model, they are in some way better when training and testing in the same language.

It is also worth mentioning that, by considering the fact that the test set is qualitatively different from the training set is not the standard scenario in machine learning, SVM is better to adapt to this scenario than Random Forest and Decision tree.

Overall, all three experiment results indicate that English AGTs are rather easy to detect and Finnish dataset turns out to be the most difficult one. The Swedish dataset with the lowest AGT ratio (6.3 %) and fewer AGTs generated by pure bot accounts, has slightly less errors.

6.2 Future work

Future studies would benefit from adding 2-3 more languages to dataset. If we had more time, we could have done it for the Persian language. Apart from the mentioned one, future research can address why Scikit-learn has quite higher results than Weka. Also, the experiments can be extended to better understand more about each language's individual behavior.

The results show that a fine-tuned SVM model all of a sudden had better results when applied to an unseen language. This finding indicates over-fitting with respect to the languages used in the training data. The fact that the test set is qualitatively different from the training set is not a standard scenario in machine learning and needs to be addressed in future studies.

7. References

[1] D. G. Campbell, *Egypt Unsh@ckled - Using Social Media to@#:) the System: how 140 Characters Can Remove a Dictator in 18 Days*. United Kingdom: Cambria Books, 2011.

[2] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment," *Int. AAAI Conf. Web Soc. Washington, DC. Media*, pp. 178–185, 2010.

[3] A. Gay Avello, P. T. Metaxas, and E. Mustafaraj, "Limits of electoral predictions using Twitter," *Int. AAAI Conf. Barcelona, Catalonia, Spain. Weblogs Soc. Media*, pp. 490–493, 2011.

[4] J. N. Sutton, L. Palen, and I. Shklovski, "Backchannels on the front lines: Emergency uses of social media in the 2007 Southern California Wild res," *Int. Conf. Inf. Syst. Cris. Response Manag.*, San Diego, CA, USA. 2008.

[5] T. Sakaki, O. Makoto, and Y. Matso, "Earthquake shakers Twitter users: real-time event detection by social sensors," *IEEE Proc. 19th Int. Conf. world wide web. North Carolina, USA*, pp. 851–860, 2010.

[6] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *J. Comput. Sci.*, vol. 2, pp. 1–8, 2011.

[7] T. Scheffler, "A German Twitter Snapshot," *Proc. Lr. Germany*, pp. 2284–2289, 2014.

[8] J. Lundberg, M. Laitinen, M. Levin, and A. Lakaw, "Creating the Nordic Tweet Stream: A real-time monitor corpus of big and rich language data," *J. Univers. Comput. Sci.*, vol. 23, pp. 1038–1056, 2018.

[9] J. Lundberg, J. Nordqvist, and A. Matosevic, "On-the-fly Detection of Autogenerated Tweets," 2018.

[10] A. You *et al.*, "Detecting Automation of Twitter Accounts :Are you a Human, Bot or Cyborg?," *IEEE Comput. Soc.*, vol. 9, no. X, pp. 1–14, 2012.

[11] C. M. Zhang and V. Paxson, "Detecting and Analyzing Automated Activity on Twitter," 2010.

[12] N. Chavoshi, H. Hamooni, and A. Abdullah, "Identifying correlated bots in Twitter," *Conf. Soc. Informatics*, Seattle, pp. 14–21, 2016.

[13] F. Morstatter, L. Wu, T. Nazer, K. M. Carely, and H. Liu, "A new approach to bot detection: striking the balance between precision and recall," *Int. Conf. Adv. Soc. Networks Anal.(ASONAM) Min.*, pp. 533–540, 2016.

[14] V. S. Subrahmanian and et al, "The DARPA Twitter Bot Challenge," *IEEE Comput. Soc. Comput.*, vol. 49, pp. 38–46, 2016.

[15] C. Grier, "@spam: the underground on 140 characters or less," *Proceeding 17th ACM Conf. Comput. Commun. Secur. Chicago,*

Illinois, pp. 27–37, 2010.

[16] A. A. Amleshwaram, “CATS:Characterizing automation of Twitter spammers,” *IEEE Fifth Int. Conf. Commun. Syst. Networks(COMSNETS)*, pp. 1–10, 2013.

[17] F. Benevenuto, “Detecting spammers on Twitter,” *Collab. Electron. Messag. anti-abuse spam Conf.*, vol. 6, p. 12, 2010.

[18] J. Martinez-Romo and L. Araujo, “Detecting malicious tweets in trending topics using a statistical analysis of language,” *Expert Syst. with Appl. Elsevier*, pp. 2992–3000, 2012.

[19] A. Barbaresi, “Collection and indexation of Tweets with a geographical focus,” *Tenth Int. Conf. Lang. Resour. Eval. (LREC), Proceeding 4th Work. Challenges Manag. Lang. Corpora*, pp. 24–27.

[20] S. Gabriel, “The 2014 #YearOnTwitter,” 2014. [Online]. Available: https://blog.twitter.com/2014/the-2014-yearontwitter. [Accessed: 09-Jan-2017].

[21] “The Twitter Rules. Tweet.,” 2015. [Online]. Available: https://support.twitter.com/articles/%0D18311-the-twitter-rules.

[22] M. Mowbray, “The Twittering Machine,” *In: WEBIST(2)*, pp. 299–304, 2010.

[23] K. Thomas, C. Grier†, V. Paxson, and D. Song, “Suspended Accounts in Retrospect: An Analysis of Twitter Spam,” *ACM SIGOPS Conf. Internet Meas. Conf.*, pp. 243–258, 2011.

[24] Yazdan Boshmaf et al, “The socialbot network: when bots socialize for fame and money,” *Proc. 27th Annu. Comput. Secur. Appl. Conf. Orlando, FL, USA, ACM*, pp. 93–102, 2011.

[25] Yazdan Boshmaf et al, “Key challenges in defending against malicious socialbots,” *Proc. 5th USENIX Conf. Large-Scale Exploit. Emergent Threat. USENIX Assoc.*, pp. 12–12, 2012.

[26] J. Lundberg, J. Nordqvist, and M. Laitinen, “Towards a language independent Twitter bot detector.”

[27] “Scikit-learn.” [Online]. Available: http://scikit-learn.org.

[28] “Tensorflow.” [Online]. Available: https://www.tensorflow.org/.

[29] J. Scotland, “Exploring the Philosophical Underpinnings of Research: Relating Ontology and Epistemology to the Methodology and Methods of the Scientific, Interpretive, and Critical Research Paradigms,” *English Lang. Teach.*, vol. 5, no. 9, 2012.

[30] J. Brownlee, “A Gentle Introduction to Scikit-Learn,” 2014. .

[31] “NLTK” [Online]. Available: http://www.nltk.org/.

[32] J. Lundberg, M. Laitinen, M. Levin, and A. Lakaw, “Creating the Nordic Tweet Stream: A real-time monitor corpus of big and rich language data,” *J. Univers. Comput. Sci.*, vol. 23, pp. 1038–1056, 2018.

[33] M. Laitinen, J. Lundberg, M. Levin, and R. Martins, “The Nordic Tweet Stream: A Dynamic Real-Time Monitor Corpus of Big and Rich Language Data,” in *Proceeding of Digital Humanities in the Nordic*

Countries 3rd Conference, Helsinki, Finland, March 7-9, 2018.

[34] A. L. M. Laitinen, J. Lundberg, M. Levin, "Revisiting weak ties: using present-day social media data in variationist studies," *Explor. Futur. Paths Hist. Socioling. / [ed] Tanja Säily, Minna Palander-Collin, Arja Nurmi, Anita Auer, Amsterdam John Benjamins Publ. Co.*, pp. 303–325, 2017.

[35] J. L. M. Laitinen, "ELF and social networks: Evidence from a thirdgeneration ELF corpus," *Anna Mauranen Svetlana Vetchinnikova (eds.), Lang. Chang. Impact English as a Ling. Fr. Cambridge Cambridge Univ. Press*, vol. Forthcomin, 2018.

[36] R. Wang, B., Zubiaga, A., Liakata, M., Procter, "Making the most of tweet-inherent features for social spam detection on twitter," 2015.

[37] E. K. S. B. and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, Inc, 2009.

[38] C. S. and B. Ribeiro, "The importance of stop word removal on recall values in text categorization," IEEEXplore, *Neural Networks, Proc. Int. Jt. Conf.*, vol. 3, pp. 1661–1666.

[39] L. A. Juan Martinez-Romo, "Detecting malicious tweets in trending topics using a statistical analysis of language," *Expert Syst. with Appl. An Int. J.*

[40] R. T. Larson, Keri, Watson, "The Impact of Natural Language Processing Based Textual Analysis of Social Media Interaction on Decision Making," *Proc. 21st Eur. Conf. Inf. Syst.*

[41] K. Lee, B. D. Eoff, and J. Caverlee, "Seven Months with the Devils: A Long-Term study of Content Polluters on Twitter," *AAAI Conf. Weblogs Soc. Media*, Barcelona, Spain, vol. 34, pp. 76–80, 2011.

[42] S. Marsland., *Machine learning: an algorithmic perspective*. 2014.

[43] M. Mohri, A. Rostamizadeh, and A. Talwalkar, "Foundations of machine learning," *MIT Press*, 2012.

[44] M. McCord and M. Chuah, "Spam detection on twitter using traditional classifiers," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics, Banff, Canada)*, vol. 6906 LNCS, pp. 175–186, 2011.

[45] L. Breiman, "Random forests. Machine Learning," *2001*, pp. 5–32.

[46] J. R. Quinlan, "C4.5: Programs for machine learning," *Morgan Kaufman Publ.*, 1993.

[47] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Proc. 2007 Conf. Emerg. Artif. Intell. Appl. Comput. Eng. Real Word AI Syst. with Appl. eHealth, HCI, Inf. Retr. Pervasive Technol.*, p. 3/24, 2007.

[48] C. Cortes and V. Vladimir, "Support-vector networks. Machine Learning," pp. 273–297, 1995.

[49] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," *Proc. fifth Annu. Work. Comput. Learn.*

theory, COLT '92, ACM, pp. 144–152, 1992.

[50] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," *Springer New 700 York, New York, NY*, 2009.

[51] A. H. Wang, "Detecting spam bots in online social networking sites: a machine learning approach," *Data Appl. Secur. Priv. XXIV. Springer*, pp. 335–342, 2010.

[52] Krishni, "An introduction to Grid Search," 2019. [Online]. Available: https://medium.com/datadriveninvestor/an-introduction-to-grid-search-ff57adcc0998.