LEBANESE AMERICAN UNIVERSITY

Text-based Framework for Spam Detection in Twitter

By

Bahia M. Halawi

A thesis
Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

School of Arts and Sciences

May 2017
THESIS APPROVAL FORM

Student Name: Bahia Halawi
ID #: 200903423

Thesis Title: Text-Based Framework for Spam Detection in Twitter Events

Program: Masters in Computer Science

Department: Computer Science and Mathematics

School: Arts and Sciences

The undersigned certify that they have examined the final electronic copy of this thesis and approved it in Partial Fulfillment of the requirements for the degree of:

Masters of Science in the major of Computer Science

Thesis Advisor’s Name Dr. Azzam Mourad
Thesis Co-Avisor’s Name Dr. Hadi Otork
Committee Member’s Name Dr. May Hamdan

Signature

DATE: 11/05/2017

Signature

DATE: 11/05/2017

Signature

DATE: 11/05/2017
THESIS COPYRIGHT RELEASE FORM

LEBANESE AMERICAN UNIVERSITY NON-EXCLUSIVE DISTRIBUTION LICENSE

By signing and submitting this license, you (the author(s) or copyright owner) grants the Lebanese American University (LAU) the non-exclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic formats and in any medium, including but not limited to audio or video. You agree that LAU may, without changing the content, translate the submission to any medium or format for the purpose of preservation. You also agree that LAU may keep more than one copy of this submission for purposes of security, backup and preservation. You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant LAU the rights required by this license, and that such third-party owned material is clearly identified and acknowledged within the text or content of the submission. IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN LAU, YOU REPRESENT THAT YOU HAVE FULLFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT. LAU will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

Name: Bahia Halawi
Signature: [Redacted]
Date: 18th of May, 2017
PLAGIARISM POLICY COMPLIANCE STATEMENT

I certify that:

1. I have read and understood LAU’s Plagiarism Policy.
2. I understand that failure to comply with this Policy can lead to academic and disciplinary actions against me.
3. This work is substantially my own, and to the extent that any part of this work is not my own I have indicated that by acknowledging its sources.

Name: Bahia Halawi
Signature: 
Date: 16th of May – 2017
This work is dedicated to my loving husband whose sacrificial care and outstanding support empower me day by day. I consider this work as well as any other accomplishment a modest gift to my parents who serve as my inspiration, now and always, seeking their content and satisfaction. Also, to my siblings, Hiba S, Hiba H, Abdallah, Lama and Rawan my pride, happiness and hope. Finally, I would like to dedicate this to my second family, friends and everyone who showed support or assistance throughout the way.
Acknowledgement

This thesis wouldn’t have been completed without the constant support, inspiration and knowledge of my dear supervisor, Dr. Azzam Mourad and my co-advisor Dr. Hadi Otrok. Their attitude and positive spirit were the trigger in each phase of the work. I also take the opportunity to express my gratitude for their granted efforts that helped reshape many of the ideas and eventually obtain better results. I shall thank my committee member, Dr. May Hamdan, who never failed to guide me in my undergraduate as well as my graduate studies. Finally, I shall thank Dr. Ernesto Demiani for his valuable support and contribution.
Text-based Framework for Spam Detection in Twitter

Bahia Halawi

Abstract

Due to the inevitable popularity of twitter, as well as its ability to transport messages into sparse communities, spammers tend to take twitter for granted in spreading their commercial messages. Moreover, different spammers behave in various manners. Some of them adopted behavioral approaches; others made use of content entropy while many others explored bait behaviors. Previous related works look at this problem from the perspective of studying a tweet along with its metadata, performing different statistical and profiling activities in order to infer about spam. However, these approaches do not pay attention to the limitations placed over twitter’s streaming API, minimizing user’s abilities to extracting follower and followees’ data. Also, many of the approaches violate user privacy by investigating personal data about him/her without previous consent. This thesis is dedicated to studying the relationship between tweets shared by different users, particularly, content considered as spam vs. legitimate. Moreover, we will overcome the above mentioned limitations by developing a set of Message to Message analysis approaches. First, we will deploy the cosine vector similarity and later the natural language toolkit and co-occurrence model to enhance the correctness in detection. However, due to spammer’s creativity in building organic messages, hardly looking similar to old messages, these models suffer from limitations. That is why, we elaborate the use of ontologies in detecting spam over twitter during events. Our experimental results will demonstrate the efficiency of analyzing spam content/semantic relationships over twitter through ontologies.

Keywords: Event spammers, Honey pots, Hashtags, Entropy, Semantic, Ontology.
# TABLE OF CONTENTS

## I. Introduction ................................. 1

1.1 Motivation and Problem Statement .... 1
1.2 Challenges and Problem Statement ... 6
1.3 Objectives ................................. 7
1.4 Methodology .............................. 8
  1.4.1 Message to Message Approaches for Spam Detection ... 9
  1.4.2 Message to ontology approach for spam detection ... 10
1.5 Thesis Organization ...................... 10

## II. Background and Related Works .......... 12

2.1 Introduction .............................. 12
2.2 Events on Twitter ......................... 13
2.3 Tweet Message ............................ 13
2.4 Types of Spammers ....................... 14
  2.4.1 Legal vs. Legitimate User’s Characteristics ...... 14
2.5 Similarity Notions ....................... 16
2.6 Giant Component ......................... 17
2.7 Ontology ................................. 17
2.8 Related Works ............................ 18
  2.8.1 Statistical Indicators ................. 18
  2.8.2 User Profiling ......................... 19
  2.8.3 Analysis Types ......................... 20
  2.8.4 Giant Component ....................... 22
  2.8.5 Ontology Based Approaches ............... 22
III. Framework Overview and Data Preparation .......................... 25

3.1 Introduction ................................................. 25
3.2 Approach Overview ........................................ 25
3.3 Data Preparation Module ...................................... 26
   3.3.1 Data Selection ...................................... 28
   3.3.2 Data Collection/ Extraction .......................... 28
   3.3.3 Data Processing and Cleaning ......................... 29
   3.3.4 Elimination of Stop Words ......................... 30
   3.3.5 Data Structuring .................................. 31
   3.3.6 Data Filtering .................................... 31
3.4 Data Analysis ................................................ 31
3.5 Conclusion ................................................. 33

IV. Message to Message Spam Evaluation ................................. 34

4.1 Introduction ................................................. 34
4.2 Cosine Vector Similarity Based Approach ..................... 34
4.3 NLTK Approach .............................................. 36
4.4 Co-occurrence Approach ................................... 37
4.5 Implementation and experimental results ..................... 38
4.6 Tracing the Giant Component ................................ 40
   4.6.1 Giant Component in Follower-Followees Network Model ... 40
   4.6.2 Giant Component Based on Tweet Content Network mModel ... 41
   4.6.3 Findings ........................................ 43
   4.6.4 Cosine Vector Similarity Visualizations .................. 44
   4.6.5 NLTK Visualizations .................................. 45
   4.6.6 Co-occurrence Visualizations .......................... 46
   4.6.7 Findings ........................................ 46
   4.6.8 Conclusion ........................................ 47
V.Message to Ontology Spam Evaluation .......................... 48

5.1 Introduction ............................................. 48

5.2 Approach overview ...................................... 49
  5.2.1 Article Selection .................................. 51
  5.2.2 Extraction of Ontologies ......................... 51
  5.2.3 Concept Extraction ............................... 52
  5.2.4 Tweet Evaluation Against Ontology ............ 52
  5.2.5 Tweet Filtering .................................. 53
  5.2.6 Ontology Extraction Algorithms .............. 54

5.3 Ontology Generation Module based on Text2Onto ................. 56
  5.3.1 Probabilistic Ontology Model .................... 57
  5.3.2 Data Driven Discovery ......................... 58
  5.3.3 Natural Language Processing ................. 58
  5.3.4 Implementation and experimental results ....... 59

5.4 Results using the Ontology approach .......................... 61
  5.4.1 Sports based experiment ......................... 61
  5.4.2 Technology Based Experiment .................... 64
  5.4.3 Politics Based Experiment ....................... 67
  5.4.4 Effect of Changing the Number of Articles used in Extracting
        the Ontology on Results ......................... 69
  5.4.5 Discussion ..................................... 70
  5.4.6 Conclusion .................................... 72

VI.Conclusion .................................................. 73

Bibliography .................................................... 74
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reply spammers</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Get More Followers Spammers</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>System Architecture</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>Tweet Extraction Schema</td>
<td>29</td>
</tr>
<tr>
<td>5</td>
<td>Tweet in Json format</td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>Tweet sample</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>Tweet text</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>List of tweets with common hashtag</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>Tweets Database Schema</td>
<td>32</td>
</tr>
<tr>
<td>10</td>
<td>NLTK Tree Sample</td>
<td>37</td>
</tr>
<tr>
<td>11</td>
<td>Correctness Results Collected in each experiment</td>
<td>39</td>
</tr>
<tr>
<td>12</td>
<td>Giant Component Retrieved from [18]</td>
<td>40</td>
</tr>
<tr>
<td>13</td>
<td>Giant Component for Strata Conference Tweets</td>
<td>44</td>
</tr>
<tr>
<td>14</td>
<td>Giant Component at threshold 0</td>
<td>44</td>
</tr>
<tr>
<td>15</td>
<td>Giant Component at threshold 0.7</td>
<td>44</td>
</tr>
<tr>
<td>16</td>
<td>Giant Component at threshold 0.9</td>
<td>44</td>
</tr>
<tr>
<td>17</td>
<td>Giant Component at threshold 0</td>
<td>45</td>
</tr>
<tr>
<td>18</td>
<td>Giant Component at threshold 0.5</td>
<td>45</td>
</tr>
<tr>
<td>19</td>
<td>Giant Component at threshold 0.7</td>
<td>45</td>
</tr>
<tr>
<td>20</td>
<td>Giant component at threshold 0</td>
<td>46</td>
</tr>
<tr>
<td>21</td>
<td>Giant component at threshold 0.5</td>
<td>46</td>
</tr>
<tr>
<td>22</td>
<td>Giant component at threshold 0.7</td>
<td>46</td>
</tr>
<tr>
<td>23</td>
<td>Ontology Structure</td>
<td>50</td>
</tr>
<tr>
<td>24</td>
<td>Ontology Generation Process</td>
<td>50</td>
</tr>
</tbody>
</table>
Overall Ontology Based Evaluation Model .......................... 53
Change Management High-Level Architecture ..................... 58
Rule Discovery Logic ...................................................... 59
Basketball Tweets vs. Threshold using a Soccer Ontology ...... 61
Results of Experimenting with Soccer Tweets against a Soccer Ontology 62
Results Experimenting with Spam Tweets against a Soccer Ontology . 63
Results of Experimenting with Technology Tweets against a Technology Ontology ................................................. 64
Results of Experimenting with Spam Tweets against a Technology Ontology ......................................................... 65
Results of Experimenting with Technology Event Tweets against a Technology Ontology ................................................. 65
Results of Experimenting with Technology Tweets against a Technology Ontology ......................................................... 66
Results of Experimenting with Politics Tweets against a Politics Ontology 67
Results of Experimenting with Politics Tweets against a Politics Ontology 67
Results of Experimenting with Spam Tweets against a Politics Ontology 68
Results of Experimenting with Technology Tweets against a Technology Ontology Extracted using Three Articles ............... 69
Results of Experimenting with Technology Tweets against a Technology Ontology Extracted using Five Articles ............... 69
Results of Experimenting with Technology Tweets against a Technology Ontology Extracted using Six Articles ............... 70
Chapter One

Introduction

1.1 Motivation and Problem Statement

Importance of Twitter

Social media platforms seem to be the most preferable to users from different age groups nowadays when it comes to communicating with their friends and getting in touch with them. An interesting phenomenon moreover is the use of many business owners of twitter also in marketing their services and products besides communicating with other users for different business purposes. This is treated as the professional exploitation of twitter. Twitter and other social media platforms are integrated within websites, available over tablets and mobile phones as well as web pages in an attempt to gain more popularity among users and to become easier for access and use. One of the appealing things about twitter is the fact that messages have short sizes messages have (140 characters at most), making messages simple and straight to the point. This is outstandingly essential for today’s readers, particularly youth groups, who are not attracted to long readings[24]. In addition to that, users benefit from the ability of adding a URL to these messages (after using URL shortening websites and applications). Twitter also
allows users to classify content and thus target audience based on their interests and listening desires instead of just sending around random content to everyone checking twitter. This is done by using proper hashtags that reflect the topics a user is tweeting about. Thus, twitter can allow for the generation of additional revenues as users become potential in approaching communities of interested people worldwide. It is a free marketing tool, bringing people together from all over the world and providing them with the ability to talk about their services and show it to others. Also, it grants the users optimal flexibility in talking about what they want and following as many interesting hashtags or topics as they feel like following. Besides, different participants can go around creating lists, where they actually group the people, best known for tweeting about certain topics together. This is another attempt to maximizing benefit from twitter and awaiting relevant content.

What is Spam on Social Networks?

By spam on social networks, we actually refer to the pollution induced by some users or machines, drifting content from its expected theme or notion. If a hashtag is actually discussing a certain theme, then people mentioning this hashtag on social media networks are expected to sustain a common language or interest in their publications[13]. However, for many possibly identified reasons, such as marketing, certain account holders can actually send irrelevant content, producing the so called spam. This content might look real and legitimate if diagnosed on its own. However, when analyzed within a certain scope, it becomes labeled as spam.

Why is Spam Bad?

Spam is bad and nobody really wants it. The presence of too many spammers will pollute content over twitter [22]. Users, initially deploying hashtags to enhance content classification will become annoyed of having to read irrelevant or unrelated content, relative to hashtags they are following. Spam content will decline revenues for twitter as people will use it less seriously and will switch to other platforms to connect with
others, share their ideas and market their services and products to interested people. Spam will thus allow for spreading rumors, fake news and ideas, as well as the starring of fake accounts. It is also essential to note how researchers around the globe, experimenting with twitter data and mining for insights can have less accurate results if spam is not addressed properly. The ability to eliminate the noise and tidy data is essential to make use of Twitter data across distinct domains.

**Events on Twitter**

Rather than approaching the problem of spam detection from a generic approach, we will customize our solutions and experiments in order to discover spammers in events or well identified ceremonies or hashtags. Events, nonetheless, will refer to an occasion or a topic that a group of people are discussing or tweeting about [10]. This can be noticed through a common hashtag adopted by users of this topic and its trending with time, peaking at most during or after the generation of that event.

**Tweet Message**

The tweet message body, just like the message illustrated in Figure 1, is at most 140 characters, possibly containing textual data, images, videos, URL and emoticons.

**Types of Spammers: Focus on Event Spammers**

Spammers can behave in different ways and thus be classified into groups or categories based on their approach to targeting users. Thus, it takes tailored approaches, relative to the spammer’s characteristics, to be able to identify him/her. Previous works identify many types of spam. For instance, reply spam is one technique where spammers tend to answer questions or inquiries you post by exactly the expected answer, in an attempt to loop you in an engagement, as illustrated by figure 1. Another type of spammers is the get more followers spammers. This group has tweets from users who claim to boost your audience basis. In this thesis we look at spammers from an alternative approach. To us, spammers can be real users. However, the main attribute we consider here is the
Figure 1: Reply spammers
Figure 2: Get More Followers Spammers
relevance of the content they are sharing with the hashtag they are exploiting. Typical spammer identification is part of our work yet our main focus is to trace spam content relative to event spammers.

**Legal vs. Legitimate User’s Characteristics**

There is always a margin of difference between legal and illegal advertisers. Users trying to market their products or ideas under a logical or relevant hashtag are doing their job, as long as they don’t try to force content on users. This happens when a spammer adds a hashtag that is irrelevant to your content to his tweet message, causing its discovery by users following the first topic. Also, this can clearly happen through direct messages or tweets including user mentions. That’s why, taking trending hashtags for granted to share content that is not directly related to the hashtag’s theme, makes the activity unethical.

**1.2 Challenges and Problem Statement**

Main challenges in our work exist after the limitations placed over twitter’s API, preventing streaming API’s users from accessing user’s network of followers/followees. Nonetheless, privacy concerns regarding the right to access user’s data even when it’s available, entailed a serious challenge to the selection of the approach to adopt. Moreover, additional challenges related to the filtration and cleaning of data as well as its structuring and the dealing with unlabeled tweets were additional technical challenges that made the work more complex and time consuming. In previous years, working with twitter data was much easier. Twitter API was more flexible and one could actually use simple tools like scraper wiki to extract a stream of tweets by just selecting the hashtags they wanted. These tweets would contain all the Meta data a scientist can need to perform his/her study such as the followers of that user and the accounts he actually followed along with the tweet text, Geo-location, mentions and other attributes. In 2015, twitter placed many limitations over its API. Tools like scraper wiki would
not work anymore and the process of getting data became tedious and complicated. In order to do that, in case you do not want to go to companies like Data sift and Gnip who own twitter data and are accredited of selling it, you had to access the streaming API which would allow you to traverse a small sample of tweets with limitations also on the number you can save [17]. In order to overcome this limitation through our experiments, we reverted to use an archive, which contained older tweets. We designed a script to import zipped folders from that archive, place it in a local file and extract files to arrange tweets in excel sheets. Another major challenge, which was very determinant in positioning our work, was the elimination of data attributes that provided users extracting tweets with data about followers and following users. For instance, few years back, one could get the list of followers/following users for any user. Today, this is impossible using the typical streaming API. Thus, chances of working with the follower/followees network or studying communities of users involved in a certain debate/over a hashtag became out of scope. Of course, this limitation was beneficial for us as we considered user’s data as something private and not to be disclosed without user’s confirmation or content. Yet, the problem became more challenging to us as we had to infer about spammers through content only. Here, we had few options. We could have either reverted to machine learning approaches to mimic spammer’s behavioral patterns and help detect them in future tweets or we could study the actual tokens in the tweets and collect insights about the actual semantic meanings and relationships among them. Options related to studying the URLs and mentions against predefined blacklists was out of our scope.

1.3 Objectives

In this thesis, our objective is to infer about spammers trying to exploit events over twitter. This happens when spammers realize the growth of a certain hashtag once many people start using it. In order to expand their reach, they add this hashtag, which is not actually related to the content they are sharing. In particular, if we are trying to detect spammers by using twitter’s streaming API, we can only analyze user’s content
and other characteristics, not including the list of friends or followers related to each user. Our work avoids taking this network into account, preserving user’s privacy, and focusing mainly on content spread by users in diagnosing spam content. The main corresponding objectives are:

- Detecting spam behaviors over short social network messages.
- Elaborate text-based analysis approaches for spam detection in twitter events.

## 1.4 Methodology

In this thesis, we propose a text-based framework for spam detection during twitter events. We examine a set of message to message based approaches for evaluating spam, mainly the cosine vector similarity, Natural Language toolkit and the co-occurrence approach. In order to maximize the use of these experiments, we will also develop visualizations that investigate the existence of the giant component in those network models. However, for the sake of overcoming detected limitations in these models, we finally propose and elaborate the use of ontologies for inferring about spammers. In our thesis, we will try to deploy this technique for identifying spam over twitter. For this sake, we hereby outline the thesis’ main contributions:

- Multi message to message comparison experiments for detecting spam content. In this scope, three independent experiments will be prepared to highlight their abilities in detecting spam content in a tweet message. First, the cosine vector similarity approach will be deployed, helping us in assessing the similarity in tweet’s tokens or syntax. Then, the NLTK (natural language toolkit) open source tool kit will be integrated in our code, acting as a fundamental layer to base our comparisons on, and allowing us to measure the semantic relationships between tokens as well. Later, we will develop a co-occurrence model that will track terms mentioned in the same tweet, to build up relationships across tweets based on the co-mentioning of any of the previously connected tokens.
• An attempt to tracing the giant component based on content-related relationships. A giant component can be really beneficial in discovering spammers. However, this component is usually used while modeling users and their associated circles of connections. In our case, and after collecting the results of each of the above three models, we will try to trace the giant component and check whether it provides any added value in terms of filtering out spammers.

• A message to ontology comparison approach along with a proposal for its parallelization for empowering the ontology guided comparison. In this model, we will prove experimentally, that ontologies can be beneficial in detecting spammers, even in tweet message. More or less, we will test this model against few topics and then propose a parallelization model that can further enhance our work. The main idea behind the work is to prove that specialized ontologies, in terms of topic, are a relevant asset while detecting spam, just like in emails. Throughout this thesis, different steps from selecting articles to extracting ontologies and building up a relevant dictionary to test tweets against will be emphasized. Moreover, an open source tool, text2onto, will be used to perform this. The thesis will thoroughly explain the mathematical logic behind each step along with the equations implemented across each phase. Later, the experiments associated with different topics will also be explored. Surprisingly, many findings are extracted from the experimental work which deploys ontologies. The results will shed light over these findings, mainly associated with the cardinality of needed terms while testing as well as variation of this cardinality as a function of topic.

1.4.1 Message to Message Approaches for Spam Detection

First, we will examine the cosine vector similarity approach to detecting spam, assessing its strengths and weaknesses. Similarly, we investigate the usefulness in deploying the Natural Language toolkit libraries for the same purpose. Along with these approaches, we elaborate a co-occurrence based similarity model that allows for an alternative scenario in detecting spammers.
Cosine Vector Similarity

We will exploit the cosine vector similarity approach in detecting similar terms or tokens in different tweets, in an attempt to measure the closeness among them.

NLTK

The natural language toolkit will be implemented also in order to trace semantic connections among different terms across tweets.

Co-occurrence Model

In the co-occurrence model, we track the mentioning of two or more terms together in the same tweet for the sake of associating this with weights, depending on the distinct mentions across tweets. The main intent in this approach is to generate connections among terms that are used together and later on validate the legitimacy of tweets mentioning the tracked tokens.

Giant Component Visualizations

In this set of experiments, we trace the giant component by generating network models that reference tweets instead of users. The intuition here is to investigate the existence of outliers that filter out spammers.

1.4.2 Message to ontology approach for spam detection

In the thesis, we propose an ontology based approach, for the elimination of suspicious content over twitter. To the best of our knowledge, ontology based approaches are only used for detecting spam over emails.

1.5 Thesis Organization

The remaining of this thesis is organized as follows: In chapter two, we emphasize the topic by introducing all its related concepts and background information. In the
third chapter, we will evaluate our model and explain its different components. Later in the fourth chapter, we will closely examine the Message to Message Techniques for Spam detection and then document the attempts to tracing the giant component, based on each model. In chapter five, we emphasize the message to ontology approach which we have proposed. Nonetheless, we conclude in the sixth chapter.
Chapter Two

Background and Related Works

2.1 Introduction

The problem of discovering spammers over twitter is not an easy task. Spammers with time learnt about approaches used to filter them out and are thus able to overcome these approaches and act as regular users [7]. Over and over again, new techniques are implemented for the sake of tracing spammers. Yet, with time, these techniques become known to spammers, allowing them to find escapes. This a game of cat and mouse. With more people using twitter, spammers attempt more exploitations for this platform in order to reach for more people. On the other hand, detecting these attacks, in an automated fashion and in real time, becomes a bigger challenge for twitter [9]. On a very broad scale, previous works that look at the same issue can be placed under few umbrellas. Below, we shall mention the main and most effective approaches discussed by researchers in a large segment of papers we highlight. The second chapter of this thesis is devoted to the clarification of the concepts used and mentioned in our research work. First, we will emphasize these concepts. Second, we shall make clear the importance of discovering spam over twitter and the benefits gained from achieving this. Also, we will be examining different types of spammers and explaining in details how each
one behaves and differs from others. Later on, we will be relating this work to other previous ones, revising the literature and diagnosing similarities among the works. We will also explain limitations to present the contribution made in our thesis.

### 2.2 Events on Twitter

One of the most common activities that get to be noticed over twitter are events. Different types of events that enfold a community of users or enthusiasts about a topic or trend usually filter themselves out from other users by using a hashtag or a collection of related hashtags. In this case, hashtags act as binders or filtering papers placed within the twitter streaming funnel, selecting a group of tweets and isolating them from tweets that are not related to them. Moreover, event organizers, attendees, along with multinational followers tracking this event are regularly in the notion of analyzing this hashtag and studying its circulation to assess its impact or reach [12]. The problem occurs whenever the hashtag becomes trending and twitter activists start noticing it. At this point, this hashtag becomes a target for spammers or marketers to promote their ideas or products by mentioning this hashtag. Spammers tend to spread their tweets or ideas by mentioning many trending hashtags at once or iteratively in order to get noticed and seen by the largest range of people on twitter. However, for trackers, collecting these tweets and searching for trends or insights, the data being collected becomes polluted with spam.

### 2.3 Tweet Message

A tweet message is a set of 140 characters that can contain textual data, images, videos, URLs and emoticons. Messages on twitter or tweets are characterized by their use of hashtags. These hashtags allow users to who share same interests to connect over that theme. In addition, these tweets have many characteristics and can allow for the extraction of insights and influence propagation. Some tweets are written in a regular English language and are thus easy to understand. However, some tweets contain a lot
of abbreviation making them hard to understand by machines. Tweets are expressive and freely used, making it common to write in slang and use abstract words. Each tweet can help in generating tweet related attributes including the tweet syntax, statistical indicators about the tweet and its owner as well as semantic relationships among the tweet and other relative content.

2.4 Types of Spammers

Different types of spammers are recognized relative to varying user’s behavioral patterns. Some spammers seek follow backs from other users over twitter in order to extend their circle of audience and strengthen their reach. Those users mainly ask for retweets and follow backs sometimes [16]. Sometimes they offer the same things for users who respond to their requests. Others tweet about celebrities mentioning them in all of their messages. Another sample of spammers are more dangerous and they try to attract twitter users to clicking malicious URLs or links. Spam bots are another class of accounts created in an automated fashion to transmit certain tweets- sometimes the same content- in a larger frequency. Previous works identify many types of spam. In our case, spammers are different. Although typical spammers are to be included in our identification for spammers, we will try to identify another group of spammers: hash-tag spammers[4]. It is indeed essential to understand the differences among each of the groups. Groups of spammers can include bots, users behaving like bots, users sharing ads and users sharing celebrity mentions or asking for follower gain.

2.4.1 Legal vs. Legitimate User’s Characteristics

Spammers tend to use twitter in different approaches to spread their messages and infuse advertisements about their products among twitter users. One of the approaches used by spammers is to write up their messages in a tweet and then add trending hashtags even if they do not match with the topics they are tweeting about. Another technique used by spammers is to add URLs that when clicked, redirect the users to difer-
ifferent websites, sometimes websites that are not related to the content displayed or the hashtags added in the tweet. This is very annoying for twitter users because it displays undesired content on the first hand and it brings up the risk of clicking malicious URLs. This can be very risky especially when this redirection leads to the download or execution of suspicious malware. More advanced attacks by spammers can transform the attacked twitter account into a bot that is controlled by the botnet leader in additional attacks. When tracking the different approaches spammers adopt, in targeting twitter users, one can identify a list of patterns and characteristics that are able to distinguish the legitimate users from those identified as spam. These traits are related to the profile style and information as well as the behavioral approaches traced over a certain duration of time. Legitimate twitter users tend to have a simple descriptive and informative profile layout in terms of their names, profile images and narrative biography, which typically defines their interests, and tweet topics. On the other hand, in terms of behavioral patterns, regular twitter users are noticed to be responsive meaning that once you interact with them, they are very likely to respond back to you. They are also grateful when you share their content or retweet it. These users also express interest in others not only their posts or themselves. Regular users are obviously engaged in conversations and discussions that are relevant to the context they are part of and this can be traced over their timelines. Once you get in touch with these users and engage with them, it is quite clear that real characters stand behind them. On the contrary, spammers do not reflect the same traits as those diagnosed among legitimate users. They are neither responsive nor engaging. The content they publish is somehow repetitive, exhaustive, meaning it is too consumed, and very superficial. This thesis will take for granted some of the features characterizing the behavior of spammers in order to diagnose their content over twitter. In particular, the content selected by spammers as well as the hashtags deployed in transmitting this content and reaching out for different communities through it will be of major concern.
2.5 Similarity Notions

Cosine similarity is the value given for the similarity established between two vectors, indicating the size of the angle between them. Here, the values might range from zero to one inclusive [2]. Semantically, this value is exploited in comparing the meaningful connection among two terms in the language. In other ways, assuming each vector to be a tweet (with its tokens arranged in it), the measurement of the angle between two vectors (two tweets in our case) can tell how related the content of each is against the other. Detecting the Semantic distance between terms is crucial in our work. We rely on a corpus based distributional model for this sake. Some concepts can be closer to one another as compared to other concepts. For instance elections and voting are very strongly connected because they address a related theme or idea. The estimation of how close two terms can look or be is referred to as the semantic distance. So the likeliness in terms of content or meaning is core to the value collected from this experiment. Nonetheless, this is useful in querying relationships or predictions. In our case, the distance (or rather, proximity) between results in a result list, in our case, lists of tweets, says something about the relevance among two terms or how related they are when it comes to their meanings[15]. Therefore, deploying techniques that can tell how connected two ideas are by measuring the semantic distance among the two can be insightful. In our work, we make use of the natural language toolkit (NLTK) where more than 50 lexical resource are integrated along with a large set of processing libraries used to perform a wide range of operations while working with textual data [15]. Through this toolkit, terms that seem to be witnessed together, in a particular order, are mainly associated with each other. Also, co-occurring terms are often tracked. All calculations depend on the frequencies of these appearances. NLTK in our case will be used in order to measure the semantic distances across different terms in tweets being assessed. On the other hand, Co-occurrence similarity is the value returned upon comparing two statements, reflecting how many common terms, either directly or indirectly are shared among two tweets other [19]. For instance, tracing a tuple of terms in any statement establishes a connection among them. Thus, the appearance of any of
the terms from the tuple among another term in another statement, yields to inferring the existence of a similarity among all the other terms.

2.6 Giant Component

The giant component is a structure of a graph, mainly encapsulating majority of nodes of the graph within, based on a certain similarity approach or connection [5]. The tracing of this element can help in extracting certain conclusions about the graph, especially that it leads to filtering out connected elements and non-connected ones. While working with spam, it is important to be familiar with the giant component approach to filtering out spammers. This happens because spammers are expected in many cases to look like intruders to the core network.

2.7 Ontology

This is the formal name given to the set of types, relationships, interconnections and descriptions associated with any field [6]. Ontologies are the generalized, formal and explicit description of concepts in a certain domain of discourse. Moreover, it reflects the taxonomy generated by that theme or topic, encapsulating whatever seems to be useful and relevant while expressing some logic or idea in that scope. Mining articles or pieces of natural language text for extracting ontologies requires performing eight subtasks. Successively, these are: Domain terminology extraction, discovering Concepts, deriving, learning of non-taxonomic relations, rule discovery, Ontology population, Concept hierarchy extension and Frame and event detection. Different projects are involved in automating different phases in this workflow for the sake of automating the whole process. In spam detection, ontologies can be very helpful, when used as a classification base upon checking term belongings to that ontology. For instance, an email discussing technology should not contain ads promoting drinks. The assumption behind using ontologies is that an ontology includes majority of terms used in a cluster or theme. That’s why, comparing your ideas against the topic’s ontology should allow
you to trace at least common ideas or terms with the ontology. Moreover, each theme has its own ontology that does not really change in time but can grow to include more terms or tokens if new ideas/papers/publications are created within that field. In emails, ontologies have attained a high level of correctness and usefulness when exploited for detecting spam. Just like emails, tweets do contain textual data and personal expressions. This makes it possible to study them against a linguistic theme or axiom. Just like an email, handling a certain topic is expected to contain at least few terms that are present in language corpus of the theme it discusses, tweets should similarly allow for inferring about their overall theme by detecting related terms or nouns that belong to the literature specialized in their discussed scope. Initially, it will be essential to extract the ontologies from scientific articles that cover up a large portion of nouns or terms in each domain or course. Many approaches can be used for achieving this implementation. After extracting the ontology, it will be used in comparing the tweets being streamed against it. The existence of common terms will help in identifying the likelihood of placing a tweet in the spam suspicion. Once integrated with more approaches to spam inferring, this model can achieve a very remarkable correctness.

2.8 Related Works

In this section we overview the main approaches that address the topic of detecting spammers over twitter, mainly classified into the statistical, profile, and content categories.

2.8.1 Statistical Indicators

Characteristics and statistical indicators are also another projection in this analysis style. Mainly, the work in paper [4] pays attention to user preferences traced over time, identifying it through collaborative filtering techniques. Similarly[4], presents how calculating feature values associated with users, including follower to followees, and deployment of trending hashtags, can be very helpful in telling about irregular
users. A large segment of approaches focus on extracting spam-related information through analyzing the follower-followees network associated with each user. Those papers look deeper into the network associated with each user, analyzing connections in the relative circle of friends and drawing out conclusions related to the inferred relationships. In this scope, the intention is usually to discover some spammers and then study their networks of connections with the hope of inferring more spammers through tracing suspicious relationships among the two. Sometimes, direct followers of spammers who retweet them very often are suspects of being spammers. Other times, outliers to the giant component within a certain domain or topic are highly suspected for spamming a topic or an event. The general approach is to trace the giant component, which in theory should enfold all legitimate users within, leaving spammers in a ring outside. LFUn or learning from unlabeled tweets, regarding their lists, tweet size along with other features can help a lot while achieving this. Other indicators can be inferred from geographical information associated with users such as the work presented in [4]. Major limitations to this approach are mainly related to the limitations placed over twitter’s API, which made it impossible to extract the ids or accounts of users in a user’s network of followers or followees. This problem is particular to working with twitter’s streaming API, the only option to extracting a sample of tweets, in real time and for free.

2.8.2 User Profiling

Other papers focus on discovering profile related traits or patterns that best describe the spammer’s behaviors within networks. Their target is to discover the members of these clusters who share similar portfolios or behaviors such as tweeting at the same times/frequencies/same message style/redundant tweets/etc... In [1], the main contribution is in determining deceptive double characters for one user profile. This is done by analyzing non-verbal behavior variables as a function of time such as follows and retweets. Similarly, [1] deploys similar concepts why trying to predict hyper linking behavior through a hybrid method which combines graph related attributes with addi-
tional profile information. It focuses in parallel on clustering users by activity profiles and detecting users who belong to divergent clusters. [26] is also an asset in this scope. Direct approaches to checking up the user’s portfolio include, but are not limited to, the notion of having no profile photo/biography/personal tweets or rational number of followers/followees. In addition to that, the ratio of these two is also of great indications. In the typical scenarios, a twitter user is expected to have a rational combination of people who he follows vs. the people who follow him back or else it would look like a user is following people to send them his messages regardless of whether they are friends or not. That’s why, approaches within this scope mine twitter for suspicious profile characteristics or profile-based behavioral patterns. The actual problems associated with this collection of approaches in detecting spammers rely in spammer’s supporting one another and following other spammers. In many scenarios, in order to look organic and legitimate, spammers work in groups, following one another and sharing one another’s tweets to gain credibility. This makes ratio calculations inaccurate. Similarly, it makes inferences relative to inactivity or lack of user-related attributes quite inefficient.

### 2.8.3 Analysis Types

A third direction in evaluating spam is examining the tweet and all its content. In this process, majority of works focus on tracing suspicious use of hashtags or URLs. Many patterns of generating content can be attributed to different types of spammers. [17] presents the idea of using related features, mainly the taxonomy relative to each user as well as its inferred interests in allocating spammers. In addition to that, [17] is another paper through which the statistical details related to content are deployed in deciding about legitimacy of users. Authors of this paper explain how a language model is used in assessing the results along with a tracker for divergence among different language models. For instance, some spammers ask for followers to follow back while others offer packages that range from hundreds of likes to thousands of followers and interactions. So approaches to detecting those spammers include the detection of certain phrases or terms. Similarly, some techniques look for the use of nonrelated
hashtags and sometimes the use of malicious URLs that direct tweeters to unwanted webpages. Also, many papers try to measure cosine similarity among tweets using multi-dimensional vectors. Spammers adopt to new detection techniques. For instance, detecting patterns or phrases might fail after spammers started using spinbots. Spinbots are tools used by spammers to reshape a certain phrase or idea. In other words, instead of sharing the same tweet and retweeting it by the same user or group of users, spammers tend to modify the content, appearance wise, and get the same content of messages stated differently to avoid spam detection solutions. When it comes to URLs, although some websites provide spam checkers with blacklists that can help them filter out URLs previously marked with spam, spammers tend to update their URLs making it impossible to have all malicious URLs identified. Among the works similar to this thesis, is CATS: Characterizing Automation of Twitter Spammers; it seems to present an evidently related methodology. In CAT, automation of spammers is characterized based on a combination between behavioral patterns as well as profile traits. The paper pays attention to the ratio between followers and followees as well as similarities between tweets of the same person, in an intent to discover very wide divergences or very close selection of tweets that yield to building a campaign like behavior. In addition, this paper considers the attribute of time and the number of tweets collected as a variance of unit time over the whole timeline of a user as well as the differences in time between one tweet and another. Another interesting feature is the ability of this algorithm in measuring the exploitation of trending hashtags in the selection of hashtags among different posts. The authors of CATS clarify their adoption for cosine similarity measure using multi-dimensional vectors. In this thesis, along with the cosine similarity comparison, two additional experiments will be performed on the contrary of this approach. One will make use of the NLTK language axioms while the other will extract a completely new entropy that is related to a cluster of tweets from few credible articles that discuss the same theme. In order to be able to infer about “event spammers”, this thesis will alternatively look more into the content sent by these users rather than by their relationship with one another and with other spammers. In particular, this thesis
will look deeper into the divergence between content sent by traditional users or legal ones against users who send possibly irrelevant content over the same hashtag.

### 2.8.4 Giant Component

Giant component based approaches to inferring about spam are similar in the sense that they attempt to trace clusters of related users/ connected among one another, either through mapping their direct relationships or translating their content sharing/re-sharing activities. In [23] Community analysis is performed over different social graphs. The ideas is to help in detecting profiles with high ranks to a certain criminal network. Another related work in this scope is in [19]. In [19], additional attributes, inferred about central nodes, such as their names/locations and descriptions are also evaluated. In this notion, having the user’s network of friends becomes very crucial to working with this type of approaches. Otherwise, relying on content circulation becomes very superficial and fails to support on actual spam detection. The main failure in relying on this scope is due to inability to access data needed to build up those networks. In particular, twitter API made it impossible to collect this information by using the twitter streaming API. Nonetheless, to the best of our knowledge, no previous works, on tracing giant components over networks mapping out tweet content instead of follower/followees.

### 2.8.5 Ontology Based Approaches

During the past few years, the number of email users increased dramatically, leading to the tracing of enormous numbers of spam emails. As spammers always try to uncover a way to avoid existing filters, new filters require to be developed to catch spam. Furthermore, content-based Spam detection frameworks are receiving a significant amount of attention by academic researchers and industrial practitioners. Ontologies, among those content-based approaches, allow for machine-understandable semantics of data. That’s why, they have been adopted as one of the most efficient techniques in detecting those spammers. Implementations of ontology based approaches range among papers.
Ontologies are dynamic, depending on the topic or the user, allowing for real-time adjustments with changing environment for the spam diagnosis. Nonetheless, the idea behind the exploitation of this approach is to trace outliers to an ontology of a certain theme, suggesting a malicious or suspicious behavior behind that email, most likely stating that it is a spam attempt. To the best of our knowledge, papers and approaches focus on the detection of spammers in emails and none of them targets short messages as those found on social media platforms such as twitter. In our work, we are interested in testing the efficiency of using ontologies in the detection of spam in twitter.

In the following, we present few approaches targeting spam detection in emails. In [25], discussing the design of a system that uses ontologies to model features that are extracted from a user’s profile. The features are given to machine learning classifiers – J48 and Naive Bayes – that learn a user centric model of Good Spam or Bad Spam. As a result, the system puts a message into a user’s inbox if its contents are relevant to his interests. The USpam is evaluated on ENRON Spam datasets; and the results of experiments reveal that false alarms are reduced by 10% to 30 compared with existing prior art without compromising the detection accuracy. Moreover, USpam, is a two-level hierarchical Spam detection system. The framework consists of five modules:

1. Message decomposition module (MDM)
2. Syntactical safeness module (SSM)
3. Semantic annotation module (SAM)
4. Ontology construction module (OCM)
5. Adaptive filtering module (AFM)

Level 1 discriminates, with the help of MDM and SSM modules, between a Ham and Spam messages. If the Level 1 marks a message as Spam, it is presented to Level 2 classification stage that classifies it as a Good Spam or Bad Spam with the help of OCM, SAM and AFM modules. Consequently, USpam output consists of 3 classes: Ham, Good Spam and Bad Spam. In [25], two levels of ontology spam filters were
implemented: a first level global ontology filter and a second level user-customized ontology filter. The use of the global ontology filter showed about 91% of spam filtered, which is comparable with other methods. The user-customized ontology filter was created based on the specific user’s background as well as the filtering mechanism used in the global ontology filter creation. The main contributions of the paper are (1) to introduce an ontology-based multilevel filtering technique that uses both a global ontology and an individual filter for each user to increase spam filtering accuracy and (2) to create a spam filter in the form of ontology, which is user-customized, scalable, and modularized, so that it can be embedded to many other systems for better performance. Weka(http://www.cs.waikato.ac.nz/ml/weka/) was used in the system. Weka is an open source software package, which has been implemented in object-oriented Java class hierarchy. Weka provides powerful machine algorithms and classification algorithms for data mining tasks. Also, it provides association rules, clustering algorithms, and regression. The initial step was to gather a good dataset on which the decision tree will be based. This data should include the characteristics of spam email as well as the legitimate email. Also the attributes and the values for each type of email must be such that the decision tree based on the training data will not be biased. Other remarkable works deploying ontologies for inferring about spammers includes [3] and [28].

2.8.6 Conclusion

In this chapter, we have elaborated on the challenges met while trying to discover spammers in general. Moreover, we have analyzed the scope of working with events over twitter in order to identify the event spammers we are targeting in our detection activities along with the differences among other types of spammers. For that sake, many definitions and examples were also presented to make the process clearer. In parallel, many mathematical notations were introduced and explained before technically referring to them.
Chapter Three

Framework Overview and Data Preparation

3.1 Introduction

In this chapter, we outline first our approach and describe its intuition. Later, we emphasize its different components and explain how they were developed to conclude with their contribution to detecting spammers over twitter.

3.2 Approach Overview

After examining different papers that discuss the problem of detecting spammers in twitter, we realized the existence of different approaches to inferring those spammers. Some of the papers approached the topic from a behavioral perspective, analyzing how spammers tweet and interact with one another, whereas another group of the papers emphasized content related attributes to drive out specific insights related to suspicious users. However, those approaches did not consider the changes taking place at the twitter API level, specifically those related to user privacy and the collection of personal
follower networks. Therefore, in thesis, we overcome those limitations by elaborating a text based framework for spam detection in twitter events. Nonetheless, our approach embeds a data preparation process where we deploy different data driven techniques from the data pipeline to extract the raw data from the online archive and prepare the datasets needed in testing. Once tweets are ready, we run the second phase which is a series of experiments to inspect the validity of the evaluation modules we deployed. First, the cosine vector similarity approach is used and we demonstrate its strength in detecting the mentioning of the same tokens or terms across different tweets, reflecting this through higher similarity rates. However, this approach suffers in tracing organic tweets within the same cluster and that is why we deal with this through the deployment of the natural language toolkit which connects terms that belong to the same semantic category or topic tree. Also, we further developed this approach by bringing together co-occurring terms and tracking their frequencies for the sake of using them as a white list that legitimizes suspicious tweets. After collecting the results in each of the three modules, we maximized their use through tracing the giant component associated with each. Yet, overall accuracy was below our expectations and that’s why we reverted to designing an ontology based evaluation model. 3 illustrates the overall architecture behind the model, following the Data Pipeline in every step.

3.3 Data Preparation Module

This is the unit responsible for setting up the needed resources in order to be able to complete the intended experiments. In this section, we will get across distinct sub- phases while preparing our data and resources. While refining the resources in our model, we have braced up the need for scaling the system to handle big data and thus code is optimized for confronting this demand.
Figure 3: System Architecture
3.3.1 Data Selection

With the limitations placed over twitter’s API, and since our objective is to detect twitter hashtag spammers, we have investigated alternative approaches for collecting free tweets. For this sake, we will use an online archive for tweets that date back to a random collection of events and trends. These data sets are raw and unstructured. The time frame to which the data sets belong to ranges from 05-2013 to 08-2013. Nonetheless, the files are compressed in the following links: [29],[30],[31],[32].

3.3.2 Data Collection/ Extraction

After probing the data source and ascertaining the means to hoarding them, we had to automate the extraction of this data. In this phase, we were concerned with the systematic collection of tweets that we can study and experiment with. Data collection was the nucleus to our model which would enable us later on to answer relevant questions, evaluate outcomes and make predictions about future probabilities and trends. In our case, tweets were the target and we intended to scrape them from a pre-identified archive page. In the first phase of the framework, a script we prepared will automatically extract the zip file from archive website and iteratively unzip the tweet files.

We downloaded 4 files from archive.org that contain tweets (the ”Spritzer” version).

Download rate: ranged from 300 KB/s (using wget) to about 4 MB/s (using aria2c).

Figure 4 depicts the architecture used in the extraction of data. For scalability issues, we have prepared a pagination process that works by counting pages and dividing them into processes. In case we have a cluster of machines, we can distribute this work even more and allow for parallelism in importing data and mapping it into the connected database. At this point in time, we obtained the tweets to actually work with. However, these tweets still require some cleaning and structuring before we start using them in our experiments. We deployed multithreading techniques in reading files and deployed multiprocessors in extracting the tweet files. Once extracted, the tweet files are in raw Json format. They require structuring in tabular formats, cleaning from redundancies, filtering according to content and hashtags we want to work with. Figure 6 is a sample
of tweets in JSON format.

### 3.3.3 Data Processing and Cleaning

In Twitter, each user can select the language of preference, through which the overall settings and display of Twitter will appear. However, a user with Arabic settings, can still send English tweets. Therefore, we cannot rely on the preference settings, provided through the data we have extracted as an attribute, in inferring about user’s tweet language. As a result, we had to design a script to extract only English tweets.
bystudyingthelanguageofthetweetandidentifyingthehighestsimilaritybetweenits
tokensandlanguageaxioms. Inadditiontothat,wehavepaidattentiontoeliminating
duplicatetweetsbycheckingtheirkeysandmaintainatmostoneinstanceofit.

At this stage, we have unique English tweets. Here is how the tweet looks like at
this level. There are many columns, each carrying a certain attribute or description
ofthetweetincludingbutnotlimitedtothegеographiccoordinates,timeoutoftweeting,
followerandfolloweesnumbersaswellaslinkstotheprofilepictureoftheuser.

Figure 9 illustrates how the tweet database is structured and organized.

3.3.4 Elimination of Stop Words

by studying the language of the tweet and identifying the highest similarity between its
tokens and language axioms. In addition to that, we have paid attention to eliminating
duplicate tweets by checking their keys and maintain at most one instance of it.

At this stage, we have unique English tweets. Here is how the tweet looks like at
this level. There are many columns, each carrying a certain attribute or description
of the tweet including but not limited to the geographic coordinates, time of tweeting,
follower and followees numbers as well as links to the profile picture of the user.

Figure 9 illustrates how the tweet database is structured and organized.

3.3.4 Elimination of Stop Words

Just like any language, the English language has a lot of stop words that pollute
the tweet text, in particular, when trying to analyze it along with its metadata. These stop
words are common words, found in any language and having no significant meaning
when presented solely [14]. Moreover, removing them from tweets can allow us to
experiment with the important words alone. In our experiment, first of all, for all tweets we will remove all the stop words and the symbol from the token as well as the # from the token and a token that contains the ‘http’ link. Then we will create a vector from each tweet. Examples of stop words: the, a, an, above, across, before, for, an, nor, but, or, yet, and so). Our list of stop words is minimal with only determiners (tend to mark nouns where a determiner usually will be followed by a noun) or determiners with prepositions (express temporal or spatial relations) or just coordinating conjunctions (connect words, phrases, and clauses) depending on the needs of the application [14].

3.3.5 Data Structuring

In structuring our data sets, we have examined all the attributes that accompany a tweet, imported each one into a distinct column in our excel sheets and organized the columns according to relevance to our work and make the experimental part easier and neater.

3.3.6 Data Filtering

Tweets collected contain different topics and different themes as they are generated by random users, under different hashtags and different time zones. We have a timeline of tweets that arranges tweets by the timing of their posting. Furthermore, our intent is to perform our experiments based on events so we relied on hashtags to filter each set of tweets discussing a topic. That’s why, at this point in time, we have prepared the data sets that contain tweets, each group clustered together according to the hashtags they mention, and having the attributes of each tweet assigned to it. Below is a sample of tweets sharing the same hashtag with the attributes of each tweet also appended. Besides, the tweets are archived in a database with the following schema:

3.4 Data Analysis

In analyzing twitter content, one can rely on different aspects of the message in comparing it against other messages. Message to message approaches can call for comparing
Figure 8: List of tweets with common hashtag

Figure 9: Tweets Database Schema
similarities among tokens of a tweet, semantic connections or occurrences of words and their associated co-occurrences. On the other hand, tracing the giant component, which is the accumulation of most connected nodes of users in the central part of the network, can filter out a layer of users, not belonging to this giant component, and thus not considered as a part of this trend or hashtag. Therefore, those users are suspected of being hashtag spammers or event intruders. The technique of relying on ontologies to compare tweet messages against can be another alternative to approaching the problem of inferring about spam. In this scope, each theme or topic, having its own ontology, can be traversed to track the appearances of tweet tokens in that ontology and legitimize its content. Nonetheless, the giant component tracing is another key approach that is eminent in our analysis phase.

3.5 Conclusion

In this chapter, the data driven framework has been introduced and each of its steps emphasized. In each segment of the data pipeline, we have explained the technical steps to fulfillment. Moreover, samples of the outcome at each level were added to clarify the transition across different phases in the data preparation module, which is the core of this chapter’s work.
Chapter Four

Message to Message Spam Evaluation

4.1 Introduction

In this work, we elaborate three different message to message approaches to estimate the usefulness and efficiency of each method in pointing out spam content. We have started with the cosine vector similarity approach and due to particular limitations in tracing spam through this approach, we proposed the use of NLTK approach, tested it also and outlined the results. Limitations in both approaches suggested the adoption of the co-occurrence model, which we have implemented and tested also in an attempt to treat the witnessed limitations. Below, we will discuss in details, the work associated with each approach, its implementation, results and limitations.

4.2 Cosine Vector Similarity Based Approach

Mathematically, the cosine similarity is the value indicating the size of the angle formed between two vectors. In our case, each tweet is a dimension or a vector and the calculated similarity or angle formed between these two vectors (tweets) corresponds to the frequency of shared terms among the tweets [11]. Cosine similarity then gives a
useful measure of how similar these two tweets are likely to be in terms of their subject matter. Moreover, suppose we have a tweet with the token “sky” appearing 2 times and another document with the word “sky” appearing 1 time. The Euclidean distance between them will be higher but the angle will still be small because they are pointing to the same direction, which is what matters when we are comparing tweets [11].

The cosine of two non-zero vectors can be derived by using the Euclidean dot product formula:

\[ a \cdot b = \|a\| \|b\| \cos \theta \]  

(4.1)

Given two vectors of attributes, A and B, the cosine similarity, \( \cos \theta \), is represented using a dot product and magnitude as

\[
\cos (\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}
\]

(4.2)

where \( A_i \) and \( B_i \) are components of vector A and B respectively. So the cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them [11]. In our approach, first, we define the set of tweets we are testing which are the collection of tweets mentioning the same hashtag. After that, we instantiate a vectorizer and transform our tweets into a matrix. Once we have the matrix for all tweets, we can calculate the Cosine Similarity between the first tweets with each of the other tweets of the set. The first row of the matrix is the Cosine Similarity between the first tweet with all tweets in the set. Note that the first value of the array is 1 because it is the Cosine Similarity between the first tweets with itself. Also we note that the presence of similar words among tweets yields to a better score when measuring their similarities.

**Example**

Considering the #2016 hashtag as a binder, and analyzing the 3 tweets we get: Tweet1 terms are summer, Cup, European, and year. Tweet2 terms are year, company, and apple. Tweet3 terms are football, teams, European, and Cup. Tweet1 is similar to
Tweet 2 by 0.3. Tweet1 is similar to Tweet 3 by 0.7. The idea is to look at tweets with relatively lower similarities, as compared to other tweets of the cluster, as spam.

### 4.3 NLTK Approach

To overcome the limitations encountered while working with the cosine vector similarity approach, in particular the problem of expressing the same idea using different terms, we decided to evaluate the Natural Language Processing tool kit for English, written in python [21]. The idea is that this tool kit which contains topic trees, can serve in measuring the semantic distances between tweets’ tokens and not only the exact occurrences of the same tokens across different tweets. This is attributed to its ability in referencing back words to their original lexical stem [21]. This is helpful in evaluating the similarities among terms, with respect to meaning and not only the exact occurrence of the term. In our approach, we map tweets discussing the same hashtag into a shared matrix and compare each tweet in that matrix against other tweets in the matrix itself. Nonetheless, instead of comparing the syntax among each tweet, we trace back each token to its origin in the English corpus, measuring the similarity according to the relativity among the main nodes in the English axiom or dictionary. The answer thus reflects how the tokens being compared are pointing to related topics or ideas.

Here is an outline to the analysis segment that can be covered using NLTK:

1. Word Similarities
2. Twitter hashtags
3. Mentions

In our approach, we have calculated the distance between words based on the distance between the words nodes in the tree related to its original parent. For instance, for car and Mountain Bike the closest parent is machine. The calculation occurs after tracing the paths to reach the original parent node. Two directions can be adopted in this scope. One can ignore hashtags or measure the divergence between the tweet and the hashtags being used in the tweet content. In our scope, we do not consider hashtags in our study.
Findings

4.4 Co-occurrence Approach

In order to overcome the problem of organic tweets, we will deploy a third technique that aims at expanding the circle of terms we examine while checking for similarity. Throughout experimenting in the previous two approaches, we realized that partial responsibility behind the low accuracy in detecting spam is because users are creative and can discuss new topics, while still tweeting to the same hashtag. The idea here is to track the tokens that get to be mentioned together in the same tweet in order to infer about the co-occurrence of these terms and relate the tokens mentioned in cross-occurring tokens. We save tuples of words occurring together and give grades for major tokens (high frequency) and lower grades to minor tokens (relatively lower frequencies). Tokens that have not appeared before get a grade of zero until new tweets with the same tokens appear. After collecting the dominant nouns and the co-occurrences of 15% of the size of the cluster, we get for each co-occurring noun a number or weight that is associated with it based on how many times these two terms appeared both distinctly in tweets mentioning the hashtag we are working with. To calculate the similarity between a new tweet and the current graph of co-occurrences, we will collect the nouns of the new tweet. Then we will check if the nouns contain any dominant noun which in
its turn contains any co-occurring noun. The assumption is that majority of new tweets mentioning the same hashtag will be talking about connected stuff or related ones, at least related to the dominant nouns and/or the co-occurring nouns. In this model, the result is based on two factors. The number of co-occurring nouns and the weights and their sum.

\[
Result = \frac{C_{occ_i}}{\sum C_{occ_i} + 1} \times f_i
\]  

(4.3)

So a tweet’s ratio is proportional to its pair’s weight. Each token will be associated with its frequency, which is incremented whenever mentioned again.

### 4.5 Implementation and experimental results

In order to fulfill the mentioned experiments, we have used an HP computer with the following specifications:

- Intel Core i5 2.3Ghz
- 8GB Ram
- 5400 rpm hard disk

Cosine vector similarity is a quick and simple approach to finding similar tweets or tokens within a tweet. It is also a very helpful indicator when it comes to detecting similarity based on exact instances across different tweets. However, it is clear that this is not enough to inferring about tweet spam. People can be really creative over twitter, generating different ideas through different tokens and approaches. Also, the same token can be expressed in different terms or words. In parallel, when it comes to organic tweets, or tweets that are relevant to the topic but with no previous mentions in older tweets, the approach becomes a complete failure in returning an approximate value that suggests the likelihood of spam. This suggests the use of an alternative approach to make the evaluation more accurate. Another limitation is that we are assuming that more retweets mean more credibility. However, spammers can be retweeting one another and this sets up a failure in the approach. Although the results obtained
while experimenting with the NLTK approach were relatively better than those collected through cosine similarity approach, the overall accuracy still seems unacceptable. Adopting the NLTK approach supports in recognizing similar ideas, expressed through different semantics. This is an enhancement as compared to the cosine similarity model. Moreover, the evaluation enhances the overall similarity since users are free to express their ideas using different words. Yet once again, we realize that this approach cannot serve solely in the elimination of spam. The generation of new ideas is not tolerated by the approach. Nonetheless, the same assumption that claims that the more frequent an idea circulating in the cluster the more credible it is has not been destroyed. Spammers can still be tweeting the same ideas, retweeting one another, and using spinbots to express the same content differently, without being treated as spammers. Along with these issues, this process is time consuming and will take a lot as we have 80,000 word nodes to start with and by calculating the relationships between each word with all other words it may reach $[80,000]^2$ or 6,400,000,000 relations. The co-occurrence model is helpful in making sense of tokens related to each other, even if they are not semantically connected or mentioned together in the same tweet. Moreover, the appearance of two tokens together will help in creating a relationship between two tokens and thus reducing chances of falsely labelling an organic tweet, discussing relevant content which does not relate semantically or redundantly to previous tweets. As illustrated in Figure 11, we realize that the correctness achieved, based on the content of the tweet alone, is very low. At first, experimenting with cosine vector similarity

![Figure 11: Correctness Results Collected in each experiment](image-url)
yields to around 25% correctness in optimal scenarios. Attempts to enhance the results through co-occurrence model, brings up a slight enhancement, keeping the margin of error really large. Similarly, the co-occurrence model introduced more improvements to the work and this was translated through a 10 percent increase in result efficiency as compared with cosine vector similarity. Yet, we consider the results obtained at this point below our expectations. This is why we will propose an alternative approach to overcome the limitations discovered in content-based evaluation methods.

4.6 Tracing the Giant Component

In this section, we will demonstrate the attempts to tracing the giant component based on two different models: the follower-followee network and later the content based network.

4.6.1 Giant Component in Follower-Followees Network Model

Working with the traditional list of friends and followers extracted from a user’s profile information over twitter, one perform a lot of experiments that support in deciding
about spam among this content. Tracing the giant component and diagnosing outliers is on top of the possible missions. Moreover, this model can give a magnified decipherment for the network topology as well as to the sub-networks and clusters, including those suspected for spam, in a network graph. Here, the giant component is the largest set of nodes (in this case users), that are directly connected to one another, either by a direct follower/followees relationship or by having previously retweeted the tweets of one another. In this scope, among the regular findings are nodes (users) with high connections or followees or nodes that get highly retweeted. In both cases, this indicates the existence of an influencer user. It is quite important to detect those influencers since they are remarkable figures in their communities or the group of people following their fields or interesting topics. Thus, being retweeted by those noteworthy figures or having mutual follower and followees connections with them reduces chances of being suspected as spam. Nonetheless, the closer a user is to those figures, the more he/she are core to this community. On the contrary, being very far from these figures and not frequently retweeted, portends that you are not producing any input that is of high interest to people in that community and thus chances of tweeting about a completely different, or spamming the aforementioned hashtag or topic, are higher. This type of partitioning usually focuses on the group of people that are completely extrovert in that community. They analyze their connections with other community members and their closeness to stars to be able to assay their weights in the network. In this example, for the sake of benchmarking our proposed experiments later on, we will primitively with demonstrating the adequacy of using the giant component approach in detecting spammers over twitter. Moreover, the exhibition will utilize the giant component technique over the list of followers and followees of people tweeting with the strata conference hashtag.13 is the demonstration of the final outcome to this experiment.

### 4.6.2 Giant Component Based on Tweet Content Network mModel

In our analysis, we are interested in analyzing the closeness among the tweets, as in content, shared by users deploying the same hashtag and thus composing a community.
We will try to catch tweets that contain content that does not relate to the other content existing in the same community. In this notion, we will also try to discover whether a giant component exists in this network of tweet content. Users whose content is not central to other user’s content (following or active on that hashtag), as well as users who are not connected by no means to any other user in the cluster of users are simply irrelevant to the cluster being scrutinized most of the time. In order to be able to infer about these “event spammers”, this thesis will alternatively look more into the content sent by these users rather than by their relationship with one another and with other spammers. This shall be apprehended through the overt divergence between content sent by traditional users or legal ones against users who send possibly irrelevant content over the same hashtag. In this section, our intention is to authenticate the existence of the giant component in the content based network graph model. Besides, we shall ratify the obtained peripheral tweets, if the giant component does exist, that will more likely mean they are exterior or not related to the main topic of the cluster. In the below experiments, we will try to discover whether a real giant component exists in a content-based network model. Each section shall render the procured network assembling a giant component. After that, we shall criticize each figure and explain its end result.

**Implementation**

In pursuance the above mentioned course, we scraped tweets discussing the hashtag (#Strataconf), one of the most famous events that happen each year. The file obtained contains 4000 tweets mentioning this hashtag and we tried to map out users into nodes and edges representing follows or content shared among one another. We used Markov clustering to filter out users and segregate them based on their relative connections among each other. A user following another will make the two connected through an edge, giving a larger weight (reflected through node size) to the user being followed. A user retweeting another will lead to the instantiation of an edge as well, giving a larger weight to the user being retweeted. The junction formed between two users is the number of shared followers or followees established among them. Nonetheless, the
alignment of users depends on this established closeness, as well as to their centrality to other members of the network. The referenced similarity is evaluated on a peer to peer basis, namely between each two nodes. More to the giant component visual, it is pivotal to keep in mind that the size of the node will grow as the tweet node becomes more important to the network it entraps. So if the tweet is connected to many other tweets, its size will be larger than a tweet that is only connected to one other tweet. Aside from this, colors in our case represent communities. Each group of linked users will provoke the unification through a color. In the above experiment, the internal circular unit represented our giant component. It encapsulates a large portion of the network, suggesting that people included in the giant component are related or tweeting about related points/topics. Nodes with relatively larger sizes belong to users with high influence when it comes to this event or topic. Also, the colors represents communities, depending on the connections traced among people belonging to each. On the other hand, in order to fulfill the second round of experiments, after filtering the twitter archive for a collection of tweets discussing one hashtag, nouns of these tweets are also extracted. The intent is to measure the level of similarity between any tweet and another tweet in that cluster. Vectors containing tweet nouns are compared against vectors containing dominant nouns (tweeted most frequently/based on the cosine vector similarity or natural language processing or high co-occurrence values). The aspiration here is to utilize the results generated by each of the three models in drawing the giant component based on the content network model in contrast with the previously achieved follower-followees network model.

### 4.6.3 Findings

Tracing of the giant component was fortunate, steering to the isolation of large group of spammers outside the giant component. The evaluation of the results and a closer look to these spam suspects proves that almost all of them are properly distinguished. Thus, the model is undeniably decisive in the elimination of spammers, if we base our analysis on the follower-followees network model.
4.6.4 Cosine Vector Similarity Visualizations

We realize that majority of values obtained, while experimenting with the cosine vector similarity, do not exceed 0.3, meaning it mandates the existence of 3 tokens out of ten, to be present in the dictionary of accepted terms. At a threshold 0, we have a single component as all tweets achieve at least a zero percent similarity by default. Nonetheless, clustering of tweets starts as the threshold grows larger. The value of the cosine vector similarity associated with each tweet is increasing with the tweets that contain redundant terms in the cluster. Moreover, we notice that similarities are very low, especially with short tweets that contain very little text. As the threshold
moves to 0.7, the distance between the nodes grows and we start to witness a new alignment, based on the token intersection across different tweets. The sparser the nodes, the less relative they are to the central or dominant terms in the cluster. However, the classification is weak and little inferences are collected. Increasing the threshold further will add to the clustering of the graph, generating three segregated clusters, each including within it a group of tweets discussing similar tokens.

4.6.5 NLTK Visualizations

Analyzing the graphs collected based on the NLTK model yields to the figures 17, 18 and 19. At threshold zero, the component is central to the network, including majority of nodes in it. Moreover, we do not realize the existence of any spammers. As the threshold grows to 0.5, clustering starts occurring. We realize the separation of few nodes and the alignment of tweets around each. This reflects a high intersection among the tokens being published in the tweets arranged around one another. Increasing the threshold more, until it becomes 0.7, more filtration is encountered, placing a group of tweets outside the central component. Thus, in practice, those are identified as spam tweets. The correctness however, of this clustering is not very high as majority of these tweets treated as spam are legitimate.
4.6.6 Co-occurrence Visualizations

Figures 20, 21 and 22 illustrate the experiments based on the co-occurrence model. In fact, figure 20 shows how the component looks whole at a threshold of zero, containing all the elements in the central node. At this point in time, all nodes evidently satisfy a default minimal threshold of zero. As the threshold increases to 0.5, few clusters are now witnessed, one at the center and a smaller one at the periphery. Spam in this case is still not clear, however, based on content similarity, we realize that tweets are being collected closer to one another. In the third figure 22, a main component in the middle is traced, along with a peripheral layer, theoretically considered as spammers. However, a large number of those tweets, whose number is relatively larger than the giant component itself, are no actual spam.

4.6.7 Findings

The obtained interactive visualizations show that accurate and neat filtration can be over follower/followees data. However, the results are not as good when trying to detect the existence of the giant component in a content based network. Attempts to tracing the giant component when the network model is relative to tweet content fail to produce a decisive giant component. Furthermore, we notice that a large number of the tweets labeled as spam are not actual spammer. Thus, the false positive rates are higher than tolerated.
4.6.8 Conclusion

In this chapter, a set of Message to message techniques in identifying spammers through giant component tracing were presented. First, the results were based on the cosine vector similarity, then we relied on the Natural Language toolkit and later on the co-occurrence model. The details behind the implementation of each of the experiments as well as the giant component tracing were recognized. Moreover, the giant component visualizations were added and analyzed to highlight the strengths and weaknesses witnessed. As a result, we were able to identify the real problems and cases where each of the approaches fail to yield accurate spam detection.
Chapter Five

Message to Ontology Spam Evaluation

5.1 Introduction

Inferring about spammers through their content is not an easy task, especially with the existence of many obstacles such as text fragments, URLs and slang phrases. Nonetheless, after speculating the cosine vector similarity approach, NLTK based approach as well as the co-occurrence based model, we realized that all three suffer from major limitations. Organic content is the most evident one which causes the different approaches to fail in pointing out spam. As people have to deal with twitter’s limited text size, users are in the notion of using shortened terms or abbreviations. Moreover, the use of slang words as well spelling mistakes encountered makes the job even harder. This makes it harder to extract inferences from content of tweets. Tweets with non-meaningful hashtags are an additional challenge. Finally, a major challenge is short tweets and the hardness in understanding context from a little text. Failure in properly classifying content between spam and legitimate leads to failure in tracing the giant component. Referring to results obtained from all three models (cosine similarity, NLTK similarity and Co-occurrence based similarity) leads to the composition of false giant component illustrations. Many of the filtered out nodes are actually core to the topic being dis-
cussed. On the contrary, a significant number of nodes structured in the middle of the giant component are not relevant to the topic and would more likely be considered as spam to that specific notion or community. After performing a set of message to message experiments in an attempt to differentiating between spam content and legitimate ones, along with the attempts to tracing the giant component in each of the experiments, we shall now propose an alternative approach to achieving this goal. On the other hand, it is evident that ontologies are among the most efficient techniques deployed in detecting spam in emails. However, to the best of our knowledge, those approaches did not use ontologies with spam detection in twitter. In this regard, we propose a new model that relies mainly on comparing tweet messages against ontologies in order to infer about spam. The idea is that ontologies extracted about a certain theme or topic should contain a large segment of terms that cover this topic. These terms are what people who discuss sub-topics in this theme use or mention. Nonetheless, by generating a dictionary of such terms and traversing them token by token, we seek to spot the same terms in the tweet text. Optimally, the witnessing of one or more terms in the tweet text among dictionary terms, gives the tweet more credibility and less likelihood of being categorized as spam. Differences among topics and hashtags are an additional expectation from the framework. Nonetheless, we aim to run the experiments over different thresholds to infer about the most ideal number of tokens needed to satisfy the set hypothesis.

5.2 Approach overview

An ontology is an implicit representation for knowledge through a set of concepts and relationships that assist in understanding this field. An example is illustrated in Figure 23. Ontologies have turned into a trending semantics driven modeling technique, in a data driven world, allowing the understanding of a particular course of interest [8]. Moreover, annotating texts semantically provides machines with the ability to process any piece of text and perform a wide range of text based applications. Additional difficulties are faced when a corpus changes, demanding modifying the ontology generated.
Logically, ontology generation models adopt a common direction, mainly designating the following steps: (1) Domain terminology extraction, (2) discovering Concepts deriving, (3) learning of non-taxonomic relations, (4) rule discovery, (5) ontology population, (6) concept hierarchy extension and (7) frame and event detection. Figure 24 illustrates the architecture of our proposed approach. Ontology generation tools act as a container for the algorithms mentioned above. Thus, they allow the generation of an ontology based on any theme or genre. For this sake, it is mandatory first to start with a set of textual files such as articles or journals. These tools mandate having these files in a clean format, meaning images, videos, figures, tables, etc…, should all be removed to allow the tool to identify different elements across the text. After wisely selecting a set of articles that typically reflect the overall ideas or terms discussed in that notion, these articles are cleaned from any figure and are then inserted into the tool. At this stage, users can interact and select the intended objectives such extracting concepts. The tool being used systematically performs the steps described above, in order to extract and
populate the ontology along with all the relationships between the corpus elements. After collecting the list of concepts in that ontology, we transform it into an array of terms that acts as a dictionary and is used in comparing the tweets against. This takes place in the evaluation phase, where tweets are modeled as sentences composed of tokens. So the terms from the tweet are traversed and compared against the terms from the dictionary, yielding to an evaluation indicator that suggests the likelihood of being a spam tweet to that particular topic.

5.2.1 Article Selection

In order to be able to extract the ontologies associated with each theme we are working with, we shall feed the ontology generation platform with credible articles that reflect the most used terms, hence composing this theme’s taxonomy. We wisely selected few articles from scientific journals with minimal intersections for the sake of covering the widest range of terms or ideas. As we are working with limited resources, we only used few articles to achieve this mission. The more articles we send to any ontology generation tool, the more time and resources it will be needed while compiling the files. In this model, we shall be selecting a group of credible articles, discussing the same topic or theme, cleaning those articles and feeding them into the ontology generation tool. Moreover, this tool will be utilized for the sake of extracting concepts and later on generating a dictionary, specialized with each topic. This dictionary, which contains the extracted concepts, will serve as our white list, traversed in real-time, when processing a new tweet, in order to tell how relative the content of this tweet is to the examined topic.

5.2.2 Extraction of Ontologies

After selecting the articles, we send those discussing the same theme or topic to our ontology generation module. In this step, the objective is to extract the taxonomy and that requires following the two trajectories explained above: First the linguistic procedures and later on, adjustments through the statistical ones. Algorithms are executed to
identify all inheritance relationships as well as the general rules and regulations among different semantic figures in the ontology. At this point, we have the list of attributes being generated. We also obtained the concepts and entities to be used in the comparison later on. These concepts are the main terms or key nouns that can be found in any textual piece that addresses a topic or a subtopic.

5.2.3 Concept Extraction

The ontology generation module used in extracting ontologies, generates the list of concepts from articles belonging to different themes. Politics, soccer, and technology were the main topics used in the experiments presented in this section. We made 3 independent runs for the text2onto, each time feeding it with articles that belong to the relative domain. So the generation of a politics ontology required running text2onto with politics articles. The generated ontology, which contains the concepts we are relying on, is to be used as the test benchmark against tweets discussing a politics hashtag.

5.2.4 Tweet Evaluation Against Ontology

The presented ontology framework is indeed the key element in the architecture being deployed. It acts as the evaluation benchmark to our datasets and returns the output needed for measuring effectiveness. To achieve this task, the ontology based algorithm relies on the collected hash set of concepts, upon extracting tokens from articles. Nonetheless, the validation model is responsible for getting necessary results. In our platform, communication among different phases across the evaluation model is done automatically. We have achieved that using scripts designed to structure input and output so that results can be inserted into various segments. This keeps our work light weighted, especially that scripts are written for parallel processing and computing. Therefore, one of the privileges of our system is that each section is implemented independently, meaning it supports reuse of code and logic in any other evaluation process. Due to computational limitations, we will use few articles about each topic. For this
purpose, we will use the software text2onto where we feed it few articles and extract the associated ontology.

5.2.5 Tweet Filtering

Hashtags have a conspicuous effect on the results and their variations. In the examples executed above, hashtags are disregarded from the evaluation. We assumed that hashtags can actually be taken for granted to diverge the outcome of the evaluation and avoid being noticed as spam. In the validation approach adopted by the ontology based algorithm, we seek terms or nouns in the tweet that are mentioned in the ontology list of concepts. A spammer can exploit this by tweeting about an irrelevant topic and just mentioning one or more valid hashtags. In this way, the spam content is equivocal to the algorithm, as it is able to overcome it by achieving the minimal relevancy required by containing that term or hashtag. As we are interested in measuring content relevance against the overall content related to that theme, we disregard hashtags. Therefore, tweets composed of hashtags only are treated as a spam to prevent users infusing meaningful hashtags from overcoming the algorithm even if their tweet is not within the theme’s scope. In other cases, users of this algorithm might be interested
in measuring the frequency of tweets to gather insights regarding various matters like supporting sports teams in a certain zones or assessing voters from different locations. Such scenarios entail accepting hashtags and weighing their presence in a tweet to take it into consideration while comparing against the ontology concepts. Even if the tweet contains no actual content or new messages, tweeting a certain hashtag is a must for cumulatively summing up users’ counts and mentions. Therefore, inclusion of hashtags in the evaluation process becomes relative to the scenario of use. Sentiment analysis projects that aim to reflect opinions and accurate emotions should completely overlook tweets made up of hashtags while number evaluations for human activity prediction such as political support can be inferred through such tweets.

5.2.6 Ontology Extraction Algorithms

Each algorithm used in the ontology generation module is used to generate or assist in the generation of a particular modeling primitive [8]. It is also important to note that these tools contain private libraries which produce declarative primitives, thus providing extensibility, flexibility and translation ability. Modeling primitives are the following: Concepts (class), concept inheritance (subclass of), concept instantiation (instance of), properties or relations (relation), domain and range restrictions (domain/ range), mereological relations, and equivalence.

Concepts (CLASS)

Concepts or classes are an assessment for the relevance of a certain term with respect to the corpus in question. In order to perform this, three logical phases are implemented:
- a) Relative Term Frequency (RTF)
- b) TFIDF (Term Frequency Inverted Document Frequency).
- c) Entropy and C-value/NC-value method
Concept Inheritance (SUBCLASS-OF)

In this phase, we made use of the hypernym structure of Word Net and Hearst patterns as well as linguistic heuristics to trace sub-class of relationships.

Concept Instantiation (INSTANCE-OF)

In this similarity based approach, the algorithm goes around extracting context vectors for instances and concepts from the text collection and assigning instances to the concept corresponding to the vector with the highest similarity with respect to their own vector. Mere logical relations are among the relations examined in the implementation. In particular, JAPE expressions for the purpose of discovering mereological (part-of) relations are exploited. Jape expressions are rules and regulations, relative to a particular language. This is done through an algorithm that counts the occurrences of sequences that reflect a part-of relation between any two terms.

Mereological Relations

Part of relations are the main focus here in the exploration. The algorithm here counts the occurrences of certain patterns that assist in the identification of the part of relation among terms. After that, the probability of collecting this value is also validated. Word Net is also used for comparing the results and highlighting major differences.

General relations

In order to extract relations across textual data, subcategories as well frequencies and arguments related to transitive, intransitive and complement sentence structures are emphasized.

Equivalence Relationships

The intuition that terms or concepts are equivalent to the extent to which they share similar syntactic contexts is deployed in featuring equivalence relationships. The algorithm thus mainly focuses on contextual features derived from the language axioms. Values
generated are later on used as the probability for the equivalence of the concepts in question.

5.3 Ontology Generation Module based on Text2Onto

Building ontologies is a complicated job that requires a lot of time and resources, especially from large amounts of textual data. Many tools can be used for performing the following task such as TextToOnto, ASIUM, OntoLearn, OntoLT and Mo’k. Although most of the mentioned tools share a large common ground, we will use text2onto because it helps us in combating a set of problems, these tools suffer from, mainly the flexibility in collecting modeling primitives instead of just representing knowledge bases semantically. In this part of the text, we will discuss the technical experiment we performed while generating the ontology through text2onto tool. Text2onto is an open source key technology for semantics-driven modeling, mainly supporting users in order to construct ontologies from a given set of (textual) data. Text2onto allows the generation and designing of an ontology from textual data. The two key features that motivated us to use Text2onto are respectively the: probabilistic model through which ontologies are attached along the generated results, and data driven discovery technique which tracks changes in attained corpus and maps the discovered variations automatically into the probabilistic ontology model in an incremental manner rather than doing it from scratch. These changes can be easily noticed and analyzed over time. While trying to extracting an ontology, different tools tend to adopt either the machine learning techniques or linguistic ones. On the contrary, Text2Onto uses both where the developers of the tool deploy different modeling primitives rather than in a concrete knowledge representation language. Moreover, the interaction with the tool is very negligible, as contrasted with other tools and linguists or machine learning specialists. In theory, as users are typically the ones who are most familiar with the domain, user interaction should be a central part of the system architecture. A controller is core to the
text2onto architecture, supporting in the relative initialization of different algorithms, which are responsible for processing data, learning orders and applying the probabilistic model. Each algorithm passes through three execution phases: The notification process where changes are tracked and then the computation phase were witnessed changes are mapped to the generated knowledge and finally the result generation phase where the corpus gets finalized and the probabilistic model updated.

### 5.3.1 Probabilistic Ontology Model

The probabilistic ontology model is one of the core notions differentiating text2onto among other ontology generation tools. Furthermore, it consists of a set of modeling primitives, regardless of the ontology representation language being used such as OWL, RDFS and F-Logic. Probabilistic Ontology Model acts as a bag containing learnt elements. Here, probabilities are deployed in order to enhance results, allowing a more precise decision on the inclusion or exclusion of a certain object. In text2Onto, a modeling primitive library is deployed in order to allow for defining new primitives in a declarative fashion. As a result, knowledge is easy to describe and represent. These modeling primitives allow for the translation of any type of knowledge needed. Each sentence is associated with a probability relative to its entities. The statement can exist with a probability that is calculated based on the following formula:

\[
P(S^{(m)}; \theta) = \frac{\exp(\theta^T f(e^{(m)}, t^{(m)}))}{\sum_e \exp(\theta^T f(e, t^{(m)}))}
\]  

(5.1)

\(P(S^{(m)}; \theta)\) is the probability for each sentence, \(\theta\) is the log likelihood of a corpus \(D\) in this ontology, \(s\) is sentence represented as a parse tree and \(t\) is a unary pattern. Here \(e^{(m)} = (e^{(m1)}, ..., e^{(mn)})\) is a vector of entities. Different entities here are looked at as a categorical random variable which has domain as all the noun phrases (PNPs and CNPs) in the corpus. Through the probabilistic ontology model, the results of the system are associated with the relative probabilities. This is a collection of instantiated modeling primitives which are independent of a concrete ontology representation language.
5.3.2 Data Driven Discovery

The main objective in this phase is to actually build up implicit specifications by analyzing the ontology variations across data. Initially, three different approaches to discovering changes can be outlined: (i) structure-driven (ii) usage driven and (iii) data driven. Data driven is the main approach to discovery through text2onto and is highly connected to the underlying data or text. So changes are expected once modification to texts occur. Moreover, change strategies are also tracked, helping in measuring influence across that ontology. This takes place prior to formally mapping out knowledge diagnosed about concepts, instances, and relations as well as knowledge about how these aspects change as depicted in figure 26. Implicit mandatory points are calculated here, allowing for bottom up modifications in behaviors and respectively in the model used for discovery. This model is specifically crucial for tracing all changes and modifications taking place and mapping this into the whole mathematical system being calculated. Figure 27 illustrates a sample of how the logic functions.

5.3.3 Natural Language Processing

Text2Onto deploys both machine learning techniques along with linguistic processing approaches while performing the ontology extraction activity. Nonetheless, Text2onto exploits the Gate framework (https://gate.ac.uk/), extending its flexibility when it comes to running new linguistic algorithms along with the ability to annotating the results through regular expressions. Before initially running any algorithm, Text2onto pro-
cesses files, tokenizing them and separating sentences from one another. Later, the POS tagger places the terms in the suitable category. In parallel, a morphological analyzer is used to lemmatize the text and after that stemmer is used to stem them respectively. At the stage, the textual material becomes ready to be used. So a Jape transducer is responsible for matching patterns across different ontology learning algorithms.

5.3.4 Implementation and experimental results

In order to fulfill the mentioned experiments, we have used an HP computer with the following specifications:

- Intel Core i5 2.3Ghz
- 8GB Ram
- 5400 rpm hard disk

The objectives of the experiments we conducted are:

1. Evaluate the performance of the main algorithm implemented in evaluating spam tweets against relevant ones (filter out irrelevant tweets).
2. Compare the correctness of the results upon changing the similarity threshold which is core to the algorithm.

3. Infer about the existence of a relationship between the theme of the topic and the threshold selected for the comparison.

<table>
<thead>
<tr>
<th>Name of Tweet set</th>
<th>Theme</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basketball</td>
<td>Sports</td>
<td>253</td>
</tr>
<tr>
<td>Tech Event</td>
<td>Technology</td>
<td>254</td>
</tr>
<tr>
<td>Politics</td>
<td>Politics</td>
<td>297</td>
</tr>
<tr>
<td>Politics 1</td>
<td>Politics</td>
<td>253</td>
</tr>
<tr>
<td>Politics 2</td>
<td>Politics</td>
<td>297</td>
</tr>
<tr>
<td>Soccer</td>
<td>Sports</td>
<td>115</td>
</tr>
<tr>
<td>Strata Event 1</td>
<td>Technology</td>
<td>392</td>
</tr>
<tr>
<td>Strata Event 2</td>
<td>Technology</td>
<td>385</td>
</tr>
<tr>
<td>Spam</td>
<td>Spam tweets</td>
<td>35</td>
</tr>
</tbody>
</table>

Table 5.1: Datasets Used in the Experiments

In order to assess the behavior of the ontology algorithm over different topics and relative to varied threshold values, we conduct a set of independent experiments. The figure below outlines the different data sets used for this purpose, including the size of each. The threshold value represents the minimal similarity accepted in validating the legitimacy of a tweet. Each tweet is cleaned against stop words and irrelevant terms are disregarded from its context. Then, tweets are iteratively tokenized. A threshold here represents the percentage of words/tokens needed minimally to accept a tweet into the legitimate category. A 0.1 threshold for instance, mandates the existence of 10% of the tweet tokens among the words in the respective ontology being used for the comparison. As the threshold increases, more tokens become required for accepting the tweets into the legitimate (i.e. not spam) category. Six different thresholds are used in testing, ranging from 0.05 to 0.5. To evaluate the effect of selecting the right threshold relative to the topic being evaluated and the tweets being processed, the same sets of tweets will be used in the execution. Changing the threshold is compulsory for tracing changes in evaluation patterns. The displayed results will imply the benefits and drawbacks for adopting each threshold. Thus, users of the algorithm will have the ability to decide on the best threshold relevant to the scenario and flexibility in tolerating spam. Strictness
in detecting spam tweets will lead to compromises at the level of false positives being detected as well and therefore over all correctness.

Table 5.2: Threshold Values Used in Testing

<table>
<thead>
<tr>
<th>Thresholds used</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
</tr>
<tr>
<td>0.1</td>
</tr>
<tr>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>0.4</td>
</tr>
</tbody>
</table>

5.4 Results using the Ontology approach

5.4.1 Sports based experiment

Evaluating a Set of Basketball Tweets against a Soccer Ontology

Fig 28 results of experimenting with basketball tweets against a soccer ontology In figure 28, the performance of the ontology based approach is evaluated against a random group of tweets discussing random NBA basketball games. Six different thresholds (0.05, 0.1, 0.2, 0.3, 0.4, and 0.5) are used for experimenting, in order to measure the impact of modifying the expected similarity. By analyzing the above results, we notice that a lower threshold (0.05 and 0.1) yield to more accuracy in terms of detecting spam content among tweets. In particular, the results are relatively better when it comes to false positives that seem to increase as the threshold increases also. Another observa-
Figure 29: Results of Experimenting with Soccer Tweets against a Soccer Ontology is that after the third threshold (0.2), the results seem to converge. Although result accuracy decrease, specifically more false positives being traced, the results among the final three thresholds are exactly the same. After the third threshold (0.2), the algorithm returns a spam answer for majority of tweets being executed.

Evaluating a Set of Soccer Tweets against a Soccer Ontology

Figure 29 above demonstrates the results of the same experiment repeated over a different data set. Tweets of this data set discuss a soccer game and conversations for tweeters about it. Observing the scores more thoroughly, we realize that the lower a threshold, the more accurate are the results. At a threshold of 0.05, the percentage of correctly evaluated tweets exceeds 62% whereas the rate decreases to 40% at a threshold of 0.1. Larger thresholds yield to relatively lower correctness results (around 33%). Moreover, the results after the third threshold (0.2) remain the same but with a high false positive rate (67%).

Evaluating a Set of Spam Tweets against a Soccer Ontology

In order to measure the effect of changing the threshold on spam recognition only, we repeat the execution of the same ontology based approach for the same six thresholds. We observe that at a lower threshold (0.05 and 0.1), although results accuracy seem better in the overall example, recognizing spam only is less efficient than detecting it at a relatively higher threshold (0.3 and above). As the threshold seems to increase, the
false negatives become nonexistent and the detection of spam tweets becomes complete

**Discussion**

While experimenting with different sets of sports tweets against a soccer ontology, we realize that the ontology based approach is powerful in detecting organic tweets and treating it as legitimate. Moreover, we notice that, the lower a threshold, the more flexible is the detection, meaning that the algorithm allows in more suspected tweets for the sake of not labeling organic tweets as spam. At the lower thresholds we witness higher overall efficiency in segmenting different types of tweets but minimal accuracy in correctly recognizing spam tweets. As the threshold increases, we can trace the trade off, overall correctness eventually decreasing but effectiveness in realizing various spam patterns increasing to become ideal. The optimal threshold therefore lies somewhere in between, most probably 0.2, where false positive rate is acceptable and spam detection is accurate. More experiments will be needed however, on different topics, to validate these findings. Another conclusion is that ontology based approaches preserve a minimal ability in detecting spam tweets, even when tweets do not belong to the same theme which the ontology being used for the comparison discusses. The third conclusion in this scope is the fact that results are being stable for all high thresholds. In particular, the last three thresholds (0.3, 0.4, and 0.5) have completely diverging results. This becomes of large importance when handling large data sets as we can reduce the effort being exerted in checking for similarity. Rather than checking for 50%
similarity to accept a tweet, it is feasible to check for 40 and even 30% similarity.

5.4.2 Technology Based Experiment

Evaluating a Set of Technology Tweets against a Technology Ontology

Figure 31 depicts the results obtained when running the ontology based algorithm against a random group of Technology tweets discussing some technology trends and events. Again, the same thresholds (0.05, 0.1, 0.2, 0.3, 0.4, and 0.5) are used in implementing the results, giving us the ability to compare among the efficiencies of each. By having a closer look at the results, we realize that lower thresholds (0.05 and 0.1) are better in detecting spam content, compared to higher thresholds. False positives obviously increase with the increase in threshold. Moreover, results obtained at the third threshold (0.2) and above (0.3, 0.4 and 0.5) are almost the same. On the other hand, accuracy declines after the third threshold as more false positives are noticed. Just like the previous model where we compared spam tweets against the soccer ontology, after the third threshold (0.2) the results are exactly the same, the algorithm returning a spam answer for almost all tweets being checked.
Efficiency in Recognizing Spam Tweets

Figure 32 illustrates the variation of result efficiency in terms of identifying spam tweets. When running the set of spam tweets, which contains different types and patterns of spam, changing the threshold shapes the efficiencies observed different. The ontology-based approach is executed against the same six thresholds used in the rest of the experiments. Results reveal that lower thresholds (0.05 and 0.1), miss some spam tweets while higher thresholds (0.2 and above) reflect a more accurate result in terms of recognizing all spam tweets. At the false negatives level, a higher threshold has a better impact on the accuracy of handling this type.

Evaluating a Set of Technology Event Tweets against a Technology Ontology

Figure 33 is another independent execution of the ontology-based approach using a set of tweets that discuss strata conference against a technology ontology. When the
threshold is small (0.05), the detection accuracy exceeds 57% over a set of almost 400
tweets. As the threshold increases, the accuracy decreases gradually to reach 30% after
the third threshold (0.2). Results after that converge and the accuracy is very low for
all three thresholds (0.3, 0.4, and 0.5).

Evaluating a Set of Technology Event Tweets against a Technology Ontology

In figure 34, another similar execution of technology related tweets are run for the same
thresholds and results match with the previous experiments. The lower a threshold, the
higher a correctness rate and the increase in threshold yields to higher false positive
rates. Stability of results is achieved after a 0.2 threshold.

Discussion

Just like the findings of the sports based experiments, results for this experiment reas-
sure the conclusion. Lower thresholds are better in overall assessment but less accu-
rate in tracing spam tweets. However, larger thresholds become strict, labeling organic
tweets as spam tweets and thus overall results decline. The compromise among both
suggests using an in between threshold, and getting lenient depending on the theme and
nature of tweets. In parallel, eliminating the need for checking for 50% similarity by
checking for 30% only is also verified again in these experiments, particularly, as the
experiments prove that results among the three thresholds converge completely.
5.4.3 Politics Based Experiment

Evaluating a Set of Politics Tweets against a Politics Ontology

In figure 35, a politics ontology is used to test a group of tweets that discuss different election topics. The smaller thresholds (0.05 and 0.1) have higher correctness rates with smaller rates of false positive tweets being labeled. On the contrary, the false positive rates increase as the threshold increases. After the third threshold (0.2), we notice that the results look the same for the false positives and correct tweets tested.

Evaluating a Set of Politics Tweets against a Politics Ontology

Figure 36 depicts the results of experimenting using a politics ontology. The correctness of spam evaluation exceeds 70% at a threshold of 0.05 and decreases to 56% at a threshold of 0.1. As the threshold increases, the rate continues to decrease until it maintains its stability at a threshold of 0.2. After the threshold of 0.2(0.2: 0.5) the
Figure 37: Results of Experimenting with Spam Tweets against a Politics Ontology correctness rate remains around 30%.

**Evaluating a Set of Spam Tweets against a Politics Ontology**

The above figure reflects the results collected upon testing the ontology approach against a group of spam tweets that contain different patterns used by spammers. Low thresholds (between 0.05 and 0.1) have less accuracy when differentiating between spam tweets and legitimate ones. Optimal results are obtained after the third threshold (0.2) where all the spam tweets get collected. At lower thresholds, accuracy seems to increase as the threshold increases also.

**Conclusion**

Experiments run with politics tweets are very insightful. They reassure the previous two conclusions, in terms of selecting the best thresholds, which seems to be 0.2. This threshold has the most logical tradeoff among the six tested thresholds. It provides us with the perfect ability to recognizing spam while not getting too strict in falsely labeling organic tweets as spam. Moreover, as conversations in politics tweets look the closet to regular sentences (as compared to technology and sports), the overall accuracy of the politics experiment are the highest. Again, results prove that we do not have to check for at least 50% similarity between the tweet and the ontology being used and this can reduce our effort in checking actually for 30% similarity only across the last three thresholds that carry the same results over all tweet themes.
5.4.4 Effect of Changing the Number of Articles used in Extracting the Ontology on Results

The figures above (38-40-40) illustrate the variation of result accuracy as we manipulate the size of the ontology being used for extracting the ontology and eventually using the extracted concepts for comparing against tweet tokens. As we have explained before, the comparison we perform is against a set of concepts that belong to an ontology. This ontology is extracted by traversing a set of credible articles that discuss a certain theme or topic. In the first experiment, 3 articles are used for the extraction of concepts, correctness rate starts around 130, 150 and 180 for a threshold of 0.05 for 3, 5, and 6 ar-
Figure 40: Results of Experimenting with Technology Tweets against a Technology Ontology Extracted using Six Articles respectively. Result accuracy continues to decrease over all three experiments as we increase the threshold. In the experiment whose ontology was extracted through 3 articles, the rates are less accurate relative to the other two experiments over all threshold. We do witness an enhancement in results as the number of articles increases.

### 5.4.5 Discussion

Results illustrate that the larger the number of articles being used for extracting ontology concepts, eventually used in the comparison, the better are the results in terms of accuracy. The experiments executed are indeed insightful. Lots of insights are generated from the independent examples and from the comparison of different examples together.

**Threshold Selection and Adaptation**

Lower thresholds help in achieving better false positive rates, as compared to larger thresholds. However, this takes place at the price of accuracy in detecting actual spam. That’s why, the most acceptable results occur somewhere in between, bringing up a compromise at the level of accuracy.
Comparison between Different Themes

Different themes of topics produce varied results in terms of accuracy of spam detection. For instance, sports related topics contain a lot of slang, abbreviations and misleading terms. Tweets in this scope are also shorter than tweets in alternative topics. That’s why, tracing spam content in these tweets is quite impossible, even with the ontology based approach. On the contrary, politics tweets have a better structuring. Some of them are even complete sentences. Also, the formality sensed in those tweets help in writing longer tweets to complete the sentence. This makes it more relevant to the ontology based approach while examining the tweets content. Therefore, topics can play a role in helping throughout the evaluation phase.

Effect of Using a Larger Ontology

The ontology in our case is acting as white list of dictionary of acceptable terms. Nonetheless, this dictionary includes basically terms frequently mentioned in a group of articles or discussions relative to a topic. That’s why, the larger the ontology we have, the larger is the dictionary designated. This is helpful in having more terms to compare the tweet tokens against and will thus make the work more accurate.

Few are as Good as Many

After a certain threshold (mainly 30% similarity rate), few terms become as good as many terms while deciding on legitimacy of tweets. We noticed this during the experiments, as results across thresholds 0.3 and above converge. Of course, this becomes of big importance when the data sets being tested get larger. By adopting the smaller threshold (0.3 instead of 0.5 for instance) we reduce a big part of the overhead and collect results at a faster pace. A lower cardinality, indicating the need for a lower overlap is in this case satisfactory for detecting a real organic tweeting style or content.
5.4.6 Conclusion

In chapter 5, the ontology based approach was mainly elaborated, as well the set of mathematical formulations and semantic-based algorithms used in the extraction of the ontology. In this chapter, we emphasized the importance of working with ontologies for spam detection. Also, we have contrasted the results collected and compared them with the previous models, highlighting the ontology based framework’s ability in inferring about spam.
Chapter Six

Conclusion

The thesis examined the topic of detecting spammers over twitter by primarily focusing on the content they share. In this scope, we first addressed the use of cosine vector similarity to infer about repetitive content. Later, we deployed the Natural Language toolkit in measuring content similarity, to bring out semantic connections among different tweets, as users can express similar ideas through different terms. After that, we implemented a co-occurrence model that tracks the tokens being mentioned together in the same tweet to associate them with similarities against tweets containing any of the co-mentioned tokens. In addition to that, in the work presented, we have traced the giant component for the follower followees’ network model and attempted to trace the same element in a content based network model. The results were solely generated for cosine vector similarity experiments, NLTK experiments and co-occurrence similarities. Another cardinal component in our work is the design of an ontology based approach for the sake of evaluating spam content. In this model, we will extract ontologies from credible articles and base our assessment on dictionaries generated from the ontology concepts. In the sequel, here are the main contributions:

- A group of message to message models for tracing spam content among a larger group of tweets discussing a common hashtag.
• An attempt to tracing the giant component based on content-related relationships.

• A message to ontology comparison approach along with a proposal for its parallelization for empowering the ontology guided comparison. Results collected reveal the strength and weakness of each of the models. Moreover, we are able to prove that ontology based approaches are promising and more efficient, while examining content only, on deciding on the legitimacy of tweets. Experiments completed reveal that the optimal cases in the first three models are not as good as the ontology based one. In addition to that, relative to the ontology model, interesting findings were attained. Mainly, implying that more articles used while extracting the ontology will yield to enhanced accuracy. Furthermore, we notice that few terms can be as good as many terms, above a certain threshold, when deciding on the similarity between tweet tokens and ontology concepts. Finally, we proposed that parallelism can help deploying this model over larger data sets and over distributed clusters in execution.
Bibliography


