DETECTING NETFLIX SERVICE OUTAGES THROUGH ANALYSIS OF TWITTER POSTS

A Thesis

Presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Cailin Cushing

June 2012

COMMITTEE MEMBERSHIP

TITLE:	Detecting Netflix Service Outages through Analysis of Twitter Posts
AUTHOR:	Cailin Cushing
DATE SUBMITTED:	June 2012
COMMITTEE CHAIR:	Alex Dekhtyar, Ph.D.
COMMITTEE MEMBER:	Chris Clark, Ph.D.
COMMITTEE MEMBER:	Franz Kurfess, Ph.D.

Abstract

Detecting Netflix Service Outages through Analysis of Twitter Posts

Cailin Cushing

Every week there are over a billion new posts to Twitter services and many of those messages contain feedback to companies about their services. One company that has recognized this unused source of information is Netflix. That is why Netflix initiated the development of a system that will let them respond to the millions of Twitter and Netflix users that are acting as sensors and reporting all types of user visible outages. This system will enhance the feedback loop between Netflix and its customers by increasing the amount of customer feedback that is being received by Netflix and reducing the time it takes for Netflix to receive the reports and respond to them.

The goal of the SPOONS (Swift Perceptions of Online Negative Situations) system is to use Twitter posts to determine when Netflix users are reporting a problem with any of the Netflix services. This work covers a subset of the methods implemented in the SPOONS system. The volume methods detect outages through time series analysis of the volume of a subset of the tweets that contain the word "netflix". The sentiment methods first process the tweets and extract a sentiment rating which is then used to create a time series. Both time series are monitored for significant increases in volume or negative sentiment which indicates that there is currently an outage in a Netflix service.

This work contributes: the implementation and evaluation of 8 outage detection methods; 256 sentiment estimation procedures and an evaluation of each; and evaluations and discussions of the real time applicability of the system. It also provides explanations for each aspect of the implementation, evaluations,

and conclusions so future companies and researchers will be able to more quickly create detection systems that are applicable to their specific needs.

Acknowledgements

I would like to thank:

- my advisor, Alex Dekhtyar, who has closely followed every step of this project, suggested improvements to the system, and edited countless versions of this work.
- my Team: Eriq Augustine; Matt Tognetti; and Kim Paterson. I didn't know any of them before starting this project but over the past couple years we have supported each other, learned to compromise, and had awesome adventures.
- my professors Franz Kurfess and Clark Turner who pushed this project further by providing classes that have helped me expand the research and analysis that I have been able to do.
- all my classmates that have listened to the description of this project over and over again and have asked insightful questions.
- my boyfriend, Joel Scott, who consistently reminds me to just break it into smaller steps so I don't feel so overwhelmed and is always there to give me a kiss and remind me of how awesome my life is when I get stressed.
- my brainstomers and editors: Andrew LeBeau; Igor Dulkin, Andrew Harrison; and Kat Gibbs, who have listened to all the confusion in my head and been supportive in helping me sort it all out.
- my mom who has taught me optimism, energy, and happiness through her example. Without these gifts I might not have had the passion to pursue growth as much as I have.

Contents

List of Tables

List of Figures

Part 1

Introduction

Chapter 1

General Problem: Swift Perception Of Online Negative Situations

Twitter is an online social networking service that only allows its users to post 140 characters of text in one message. These posts are called tweets. According to Twitter Blog, as of March 14th 2011, Twitter users were posting approximately one billion tweets per week.[21] These relatively small and concise messages are a data mining dream. Many research groups are now developing systems that parse, categorize, or analyze sets of tweets to derive meaning from the patterns in this cloud of data. Some examples of uses that have been found for this data are tracking disease outbreaks[5], modeling earthquakes[11], and predicting stock prices[8]. Some common methods used to extract patterns are keyword searches, machine learning, sentiment analysis, and time series analysis[10].

One company that has recognized a use for this source of information is Net-

flix. Since Netflix is a service providing company that is highly judged by its reliability, being quickly aware of problems with their services is important because then they can more quickly resolve them. Currently, Netflix has 4 methods of outage detection in their services: internal monitoring systems; external synthetic transaction monitoring systems; customer service; and manual Twitter observation. However, each of these methods has its own problems. The internal monitors share a common infrastructure with the streaming system so an outage caused by an infrastructure problem will also disrupt the internal monitors used to detect it. The external synthetic transaction monitoring only runs very specific tests so it can only cover a subset of problems in a subset of the Netflix systems. Both customer service and manual Twitter observation use customer feedback, but they are slow, time consuming, and only covering a subset of the customer feedback that is being given.[13] So Netflix needs a monitoring system that is completely disjoint from their infrastructure and doesn't require manual human monitoring.

During the manual Twitter monitoring, Netflix employees found that when there is an outage in a Netflix system there are generally a significant number of tweets reporting it. For example, on March 9th, 2011, there was an outage that disrupted Netflix's service to the Nintendo Wii console. Image 1.1 shows some tweets that occurred during that time period. Not all of the tweets were reporting the outage, but many of them were.

So Netflix realized that they want a system that will automatically monitor these millions of Twitter and Netflix users who are acting as sensors and reporting all types of user visible outages and enhance the feedback loop between Netflix and its customers by increasing the amount of customer feedback that is being received by Netflix and reducing the time it takes for Netflix engineers to receive

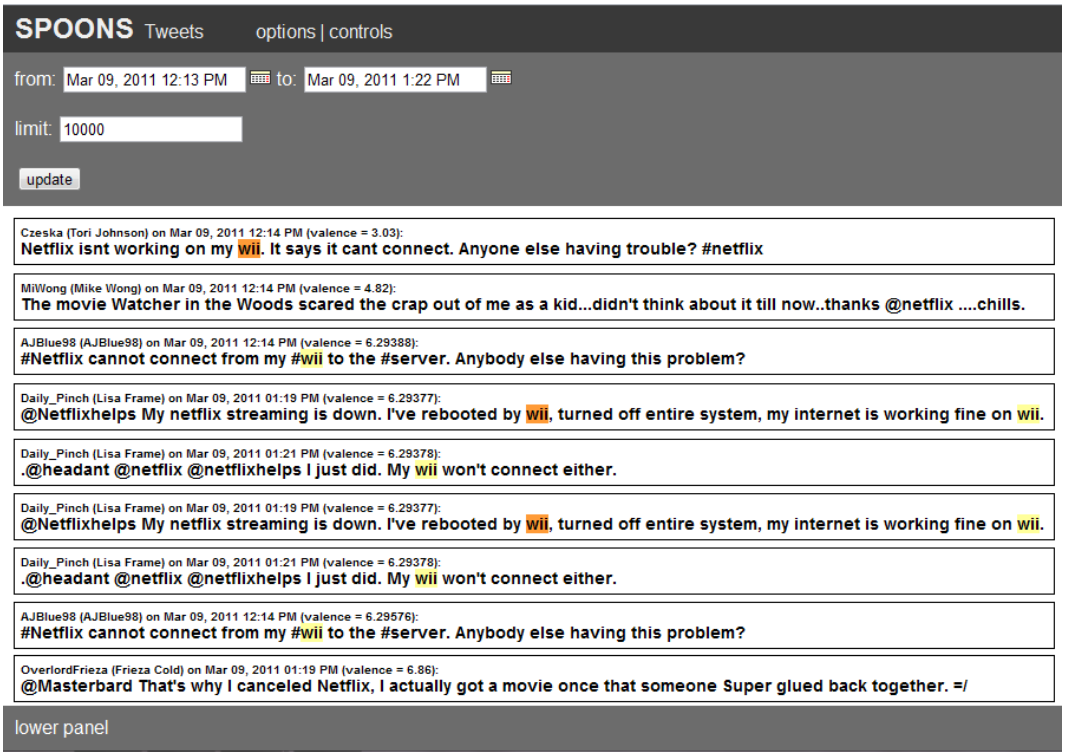

Figure 1.1: Tweets posted on March 9, 2011 during a disruption of Netflix streaming to the Nintendo Wii console.

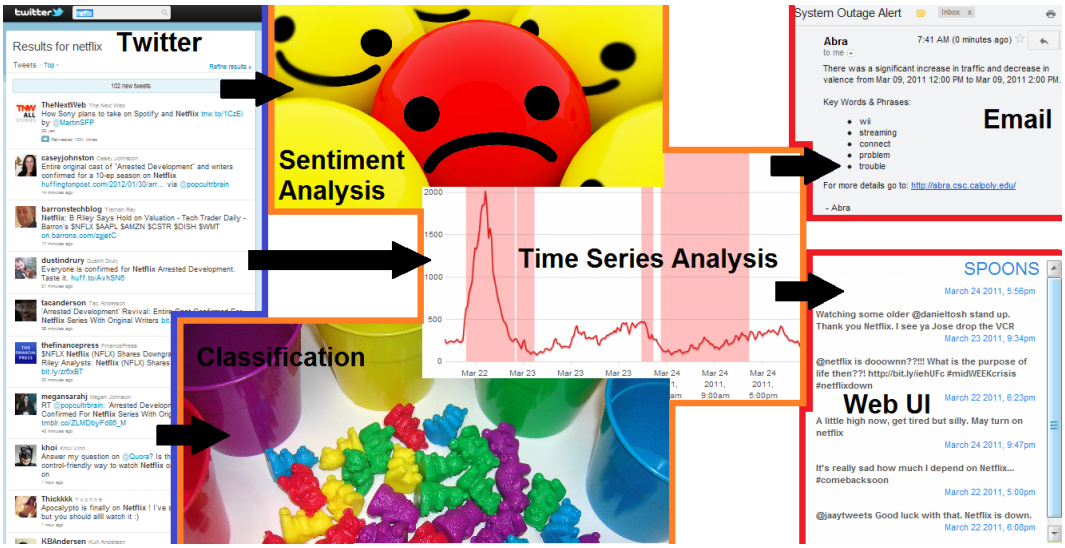

Figure 1.2: This system concept diagram shows the general flow of processing done in the SPOONS system.

the reports and respond to them.

SPOONS (Swift Perception Of Online Negative Situations) is a system that is designed to use tweets to detect outages in Netflix systems. The system supports a wide variety of detection methods that use some combination of time series analysis, classification, natural language processing, sentiment analysis, and filtering.

Image 1.2 shows how the SPOONS system can be divided into 3 main parts: input; analysis methods; and output. The inputs are tweets gathered from Twitter. Then the analysis methods use a combination of sentiment estimation, classification, and traffic volume analysis to detect when an outage is occurring[1]. The outputs of the system are: email alerts to Netflix engineers; and a web UI that displays information about the outage.

The SPOONS system is the combination of contributions from many people.

[1]The SPOONS classification methods are not described in this work

The general contributions of the work described in this thesis are:

- the implementation and evaluation of outage detection methods that monitor tweet volume over time (Chapters 5 through 8);
- several sentiment estimation procedures and an evaluation of each (Chapters 11 and 12)
- the implementation and evaluation of outage detection methods that monitor tweet sentiment over time (Chapters 13 and 14);
- and evaluations and discussions of the real time applicability of the system (Chapters 3, 8, 9, and 15).

These contributions are further defined and described in the volume analysis and sentiment analysis parts (Sections 4.3 and 10.3).

The rest of the work is organized as follows: The remainder of this part describes the SPOONS system; Part 2 describes the purpose, implementation, and results of the volume analysis methods that aim to detect outages using only a time series analysis of volumes that can be determined by filtering raw tweets based on information that is received directly from the Twitter API; and Part 3 describes the purpose, implementation, and results of the sentiment analysis methods that detect outages by looking for significant increases in negative sentiment.

1.1 Ethics of Twitter Observation

The work in this project uses content that people post on Twitter without their knowledge. This monitoring system isn't being announced to the public

because wide spread knowledge of it would increase the likelyhood of a malicious attack. This practice may lead to concerns about the level of privacy or ownership being provided to Twitter users regarding the content they post through the Twitter services. The goal of this section is to address these concerns by proving more information about the Twitter services and how the SPOONS system and this work uses the tweets.

1.1.1 Twitter Terms of Service

According to Twitter Terms of Service[22] agreement, that everyone accepts automatically by accessing or using Twitter services:

"You retain your rights to any Content you submit, post or display on or through the Services. By submitting, posting or displaying Content on or through the Services, you grant us a worldwide, non-exclusive, royalty-free license (with the right to sublicense) to use, copy, reproduce, process, adapt, modify, publish, transmit, display and distribute such Content in any and all media or distribution methods (now known or later developed)."

"This license is you authorizing us to make your Tweets available to the rest of the world and to let others do the same."

"You agree that this license includes the right for Twitter to make such Content available to other companies, organizations or individuals who partner with Twitter for the syndication, broadcast, distribution or publication of such Content on other media and services, subject to our terms and conditions for such Content use."

"We encourage and permit broad re-use of Content. The Twitter API exists to enable this."

"Such additional uses by Twitter, or other companies, organizations or individuals who partner with Twitter, may be made with no compensation paid to you with respect to the Content that you submit, post, transmit or otherwise make available through the Services."

To summarize, while Twitter users do own the content they post, by posting it through a Twitter service, they give Twitter and its partners rights to reuse it without compensation. As a user of the Twitter API, the SPOONS research group has become a partner of Twitter. So the analysis of tweets, extraction of tweet metadata, and aggregate use of that data is well within the rights of a partner of Twitter as defined by the Twitter Terms of Service.

1.1.2 Software Engineering Code of Ethics

The ACM Software Engineering Code of Ethics and Professional Practice[2] Principle 1.03 states that software engineers will, "*approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the public good.*"

Posts on Twitter are made public, therefore people who post on Twitter generally do not expect their content to remain private. However all methods currently implemented in the SPOONS system pull metadata from tweets and only use it in aggregate form. The outputs of the system don't directly link any content or data to any Twitter users. So it provides privacy to all of the authors of tweets that are contained in the SPOONS dataset.

There are some tweets quoted throughout this work. However, the authors of the tweets remain anonymous to preserve the authors' privacy.

1.2 SPOONS Requirements

Netflix has provided the following set of key requirements to be met by the SPOONS system:

Structural Independence. The outage detection system shall be structurally independent of both the software and the hardware infrastructure used by Netflix. It shall rely only on information that is publicly available and free for use. This ensures that the outage detection system stays up even when any or all Netflix servers are experiencing downtime.

Use of Amazon Web Services. Netflix is one of the largest customers of Amazon.com's cloud computing service, Amazon Web Services (AWS). AWS allows users to create new cloud machines (instances) in many regions throughout the world. The outage detection system shall be deployed on one or more AWS servers that are operationally independent of other AWS servers used by Netflix. Using a cloud solution allows the outage detection and alert system to be deployable on a global scale.

Real-Time. Netflix's streaming services run in real-time and any downtime has an immediate impact on customers. To minimize that impact, the outage detection system shall notify Netflix of detected outages as soon as possible.

Precise Outage Detection. The number of non-outage situations that raise an alert shall be minimized. While a small number of false positives detected in real-time may be acceptable, the outage detection system shall detect outages and generate alerts with as high precision as possible.

Comprehensive Outage Detection. Not all Netflix outages will generate a signal on Twitter. Those that don't may be allowed to go unnoticed by the outage detection system (as the system will have no basis for detecting them), but any outage that causes a signal on Twitter shall be detected.

User-Friendly Online UI. The outage detection and alert system shall have an easy-to-use, informative, online UI which shall provide Netflix employees with real-time information and historic data about the state of Netflix according to Twitter. The information provided shall include:

- times of outages;
- times of other anomalous events;
- current and recent Netflix-related Twitter traffic trends;
- and samples of Netflix-related tweets.

1.3 Current Status of SPOONS

This system has been worked on primarily by Cailin Cushing, Eriq Augustine, Matt Tognetti, and Kim Paterson. There have also been some course related projects that have contributed to the functionalities of the system.

Thanks to all of the people who have contributed to the it, the SPOONS system currently meets all of the requirements that have been specified. A version of the system that contains the most effective analysis methods has been deployed so Netflix engineers are receiving email alerts about outages and using the UI to track down the source of the problem.

1.4 SPOONS Limitations and Future Work

Even though the current version of some of the SPOONS methods have already been deployed at Netflix, additional challenges remain for future development:

Event Severity Evaluation. The list of outages reported by Netflix marks each outage with a severity rating of "major" or "minor". This project doesn't subdivide results into each of the severity ratings. It's possible that some of the outage events that are missed were minor outages that might not have even been visible to users. However, since these ratings aren't exclusively based on how customer facing the outage was, it is unlikely that adding this level of detail to the results would add clarity to the effectiveness of the method.

The Nature of an Outage. Netflix would like SPOONS to include information in the alert email about the nature of an outage. This information might include which hardware platforms are experiencing streaming issues, what countries or regions the outage is affecting, or perhaps just a list of key words that are showing up in the outage indicating tweets.

Malicious Tweet Attacks. Currently it is possible for a malicious Twitter user to send a large quantity of tweets "reporting an outage" and trigger false positives in the system. The only existing defense against this kind of attack is that Netflix isn't going to announce this monitoring system publicly. However, this could possibly be further avoided through the use of author profiling. The system could look at the set of a tweets that are indicating an outage and group the values by author. Then it could disclude the author with the most posts

or any authors that exceeded more than a predetermined threshold and then determine if the spike is still large enough to indicate an outage.

There are also limitations and future work specific to each type of method listed in each of the parts below.

Chapter 2

SPOONS System Architecture

The architecture of the SPOONS system[3] is shown in Figure 2.1. The gatherers use the Twitter API to collect Netflix-related tweets as they are published, and store them in a database for further processing and analysis. Once gathered, the raw tweets are run through the analysis methods. Each method contributes one or more sets of detected outages. The alert system uses the results from the analysis methods to determine if outage alerts should be generated and notifies Netflix engineers of the potential outage or service issue through email. If detailed information about the event is required, the Netflix engineers can access the systems UI through any web connected device. Through the UI, the engineers can analyze the time series, check and update the event log, and look for more information about the outage by looking though samplings of tweets.

This chapter provides a brief overview of the SPOONS system, parts of which were developed by others. The contributions to the system that are claimed by this thesis have been described broadly and are described in more detail in Parts 2 and 3.

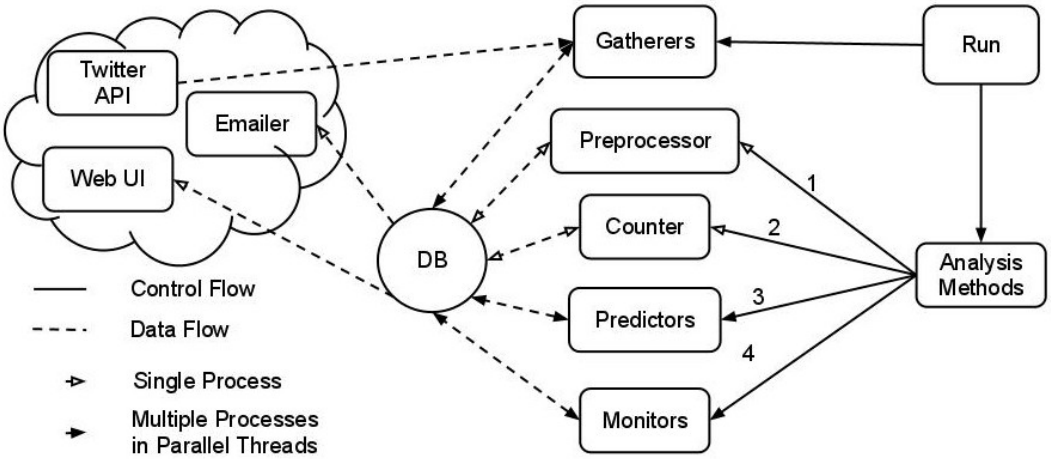

Figure 2.1: An architecture diagram for the SPOONS system.

2.1 Input

2.1.1 Gatherers

Every 2 minutes, the gatherer component queries the Twitter API for all tweets containing the word "Netflix" (in any capitalization) between the most recently collected tweet and the current time. The tweets are returned in a JSON document which is parsed and saved in the SPOONS database. An example JSON document is shown in Figure 2.2.[1] For each tweet the gatherer saves[2]:

- **Twitter ID**(id_str): a unique identification number from Twitter.
- **Published Time**(created_at): the time that the tweet was posted.
- **Content**(text): the text that was posted.
- **Language**(iso_language_code): the language that the tweet was written in.

[1] User information has been removed from this document to protect the privacy of the Twitter user.

[2] Some other information, e.g. geographical location of the tweet is also parsed and stored, but has not been used in the actual operation of the system.

```
"created_at":"Sun, 29 Jan 2012 18:20:28+0000",
"from_user":"*********",
"from_user_id":*********,
"from_user_id_str":"*********",
"from_user_name":"********",
"geo":null,
"id":163687972917620736,
"id_str":"163687972917620736",
"iso_language_code":"en",
"metadata":{"result_type":"recent"},
"profile_image_url":"http://a0.twimg.com/profile_images/
*********/profile_normal.jpg",
"profile_image_url_https":"https://si0.twimg.com/profile_images/
*********/profile_normal.jpg",
"source":"&lt;ahref=&quot;http://twitter.com/&quot;&gt;web&lt;/
a&gt;",
"text":"@Boobiewatchr i feel like i have a broken heart, netflix
left me and wont come back till i pay her!! NETFLIX IS A
Prostitute",
"to_user":"Boobiewatchr",
"to_user_id":173897014,
"to_user_id_str":"173897014",
"to_user_name":"Francois Dillinger",
"in_reply_to_status_id":163687566049161216,
"in_reply_to_status_id_str":"163687566049161216",
```


Figure 2.2: An example of the tweet JSON document returned by the Twitter API.

- **Author**(from_user): the username of the account that posted the tweet.

The Twitter API sometimes doesn't return any results for sections of the time period requested. However, the empty sections differ depending on the IP address from which the request was made. To ensure that the system is gathering all Netflix-related tweets, the gatherer runs on multiple servers, each with a separate IP address. The tweets from each server are merged in the SPOONS database using the Twitter ID to identify and eliminate duplicates.

2.2 Analysis Methods

Analysis methods are processes that analyze metadata about a subset of the tweets in the SPOONS database and determine if an outage is occurring.

First they create a time series of observed aggregate volume or sentiment values. Then they predict what the values of the time series will be during times when there isn't an outage occurring. The observed and predicted time series are compared and any time the observed traffic differs significantly from the predicted traffic the method concludes that the traffic is indicating an outage and logs an event. Each method is effectively evaluated based on how well it can create a time series that is predictable unless there is an outage event occurring.

All analysis methods run in parallel, asynchronous, unique threads so that each method can detect outages without being blocked by any of the other methods.

2.2.1 Preprocessors

Some methods require the raw tweets to first be run through one or more preprocessors before usable for outage detection purposes. The output from these preprocessors is then used as input of a counter.

2.2.2 Counters

Counters break tweets stored in the SPOONS database into time frames based on the publication time of each post. At present, SPOONS aggregates Netflix-related Twitter traffic into 30 minute time frames with 15 minute shifts. Time frames start on the hour and every 15 minutes after that; they start at :00, :15, :30 and :45 minutes respectively. This is shown in Figure 2.3. Therefore, a single day's worth of traffic is represented by about 96 time frame values, with each tweet contributing to two time frames. The overla allows SPOONS to achieve some built-in smoothing of the traffic, while still maintaining sensitivity to sudden

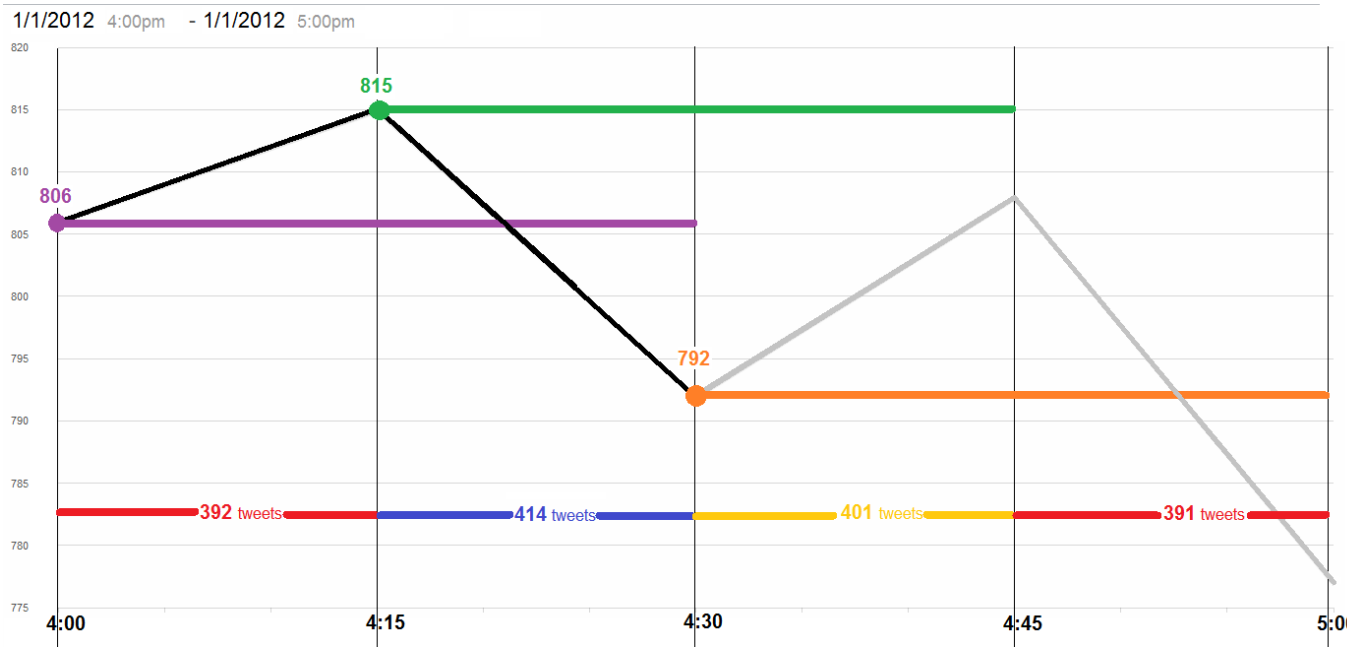

Figure 2.3: A demonstration of the spacing between time frames and how they overlap.

changes in the traffic pattern. Even though the time frames are 15 minutes long, they are updated with the newest batch of tweets from the gatherers every 2 minutes. This means that outages can be detected within about 2 minutes of the outage tweets reaching a spiking level.

The counters are the components that distinctly define the analysis methods. The subset of the tweets and type of metadata that the counter aggregates defines what the analysis method is observing. In general, there are three types of time periods that are considered when determining what the analysis method should observe:

- **Outage Time Periods**: times during an outage event. Service outages are when there is currently a member visable problem in a Netflix service. These are the events that the system is trying to detect.

- **Media Time Periods**: times during a media event. Media events are time periods when some Netflix-related news is released and highly discussed, e.g. information about quarterly earnings reports, new products/services announcements, or profiles of key Netflix personnel in the media.
- **Normal Time Periods**: times not during an outage or media event.

During media time periods, the metadata can often reflect observe values that are more similar to the ones seen during outage events than during normal time periods. This can be caused by a large number of posts about a news story or strongly negative posts in reaction to the media event. To reduce the number of false postive alerts caused by these events, some of the methods attempt to remove media tweets from the set of observed tweets by placing limitations on the tweets that the counter aggregates.

Counters store their output in the SPOONS database so it can be used by predictors, monitors, and the user interface.

2.2.3 Predictors

The key to many of the outage detection methods described in this work and employed in SPOONS is accurate estimation of normal Netflix-related traffic volume. The normal traffic definition determined by the predictors is used by the monitors to detect when the current traffic is anomalous. SPOONS implements two types of predictors: trend and model.

2.2.3.1 Model Predictors

Model predictors create a model of normal traffic that is used to predict future traffic behavior. These models are extracted through time series analysis of volume and sentiment values. These predictions are then compared to the actual volume/sentiment values and the standard deviation between the actual and predicted values is computed and maintained. Chapter 6 evaluates each of the model predictors and determines which one to use in the evaluation of the analysis methods. The following predictors are implemented in the SPOONS system.

Day Offset. Visual inspection of Netflix-related traffic has led to the discovery of a consistent daily volume cycle. In the absence of traffic anomalies, Netflix-related Twitter traffic tends to show a clear and repeatable 24-hour pattern. The day offset predictor naively predicts the traffic volume for a given time fame to be the same as the traffic volume for the same time frame of the previous day (i.e. 24 hours prior). This shift is shown in Figure 2.4.

Week Offset. The week offset predictor uses the same concept as the day offset predictor, but targets a weekly pattern. The traffic tends to show patterns that differ in amplitude depending on the day of the week. The week offset predicts the traffic volume for a time frame to be the same as the actual traffic observed during the same time frame one week earlier (i.e. 168 hours prior). This shift is shown in Figure 2.5.

Weekly Average. The week offset predictor performs poorly during time frames when anomalous traffic was observed in the prior week, replicating the anony-

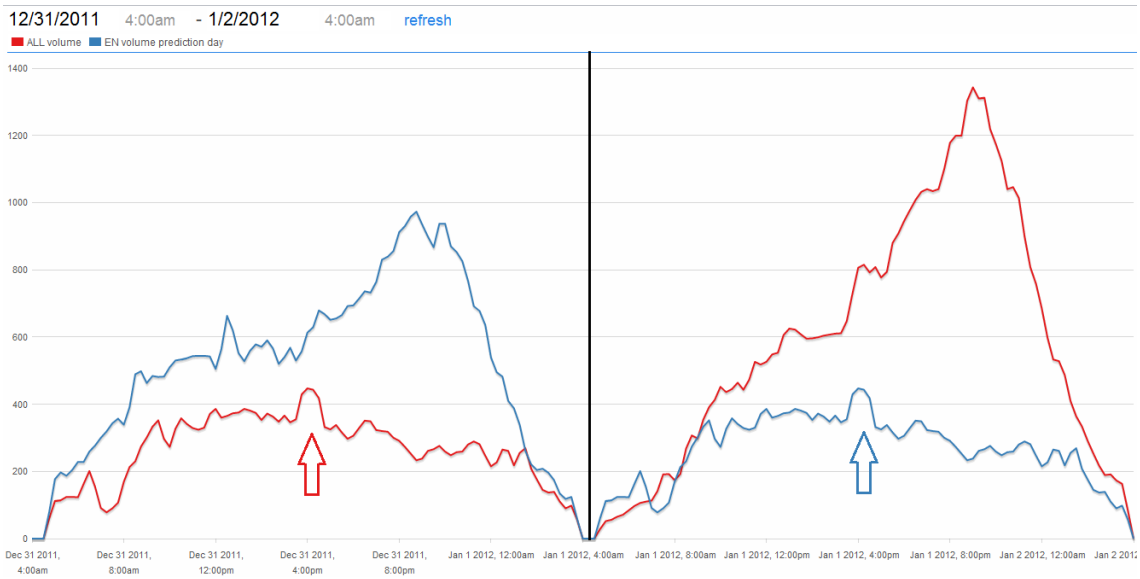

Figure 2.4: The actual (red) and day offset predicted (blue) time series for two days of the total volume dataset.

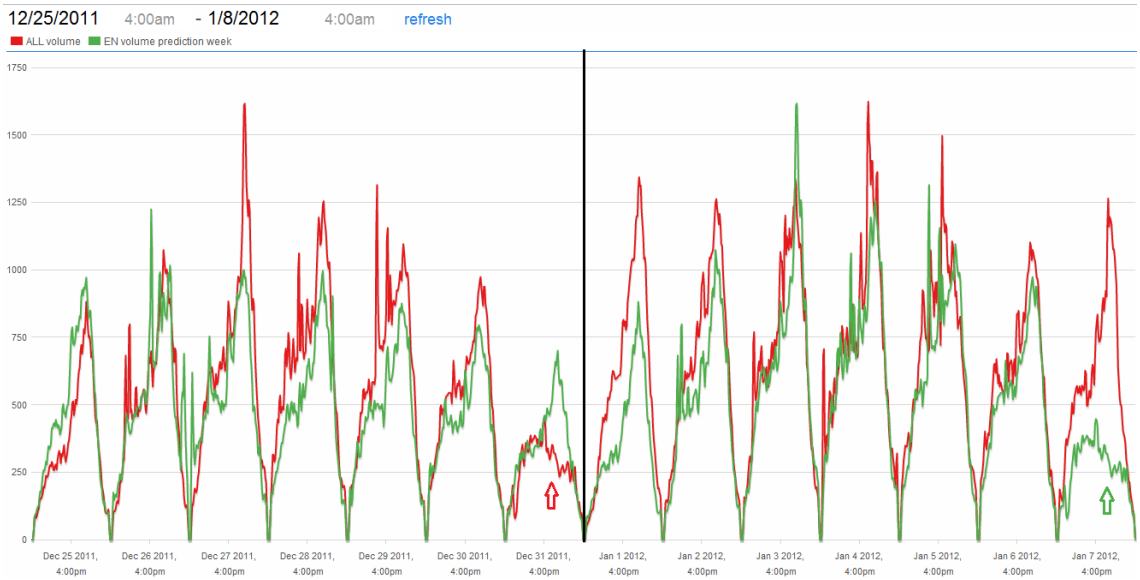

Figure 2.5: The actual (red) and week offset predicted (green) time series for two days of the total volume dataset.

mous behavior in its predictions. The weekly average predictor corrects this weakness by taking the mean of all previous values on the same time frame over the previously observed weeks. Using the mean as a prediction mitigates the effects of anomalous traffic. The set of weekly average values is calculated using the formula:

$$P(t) = \frac{\sum_{i=1}^{n} V(t-(i*W))}{\sum_{i=1}^{n}(i)}$$

Here, P(t) is the traffic volume prediction at time t, V(t) is the actual observed traffic at time t, and n is the total number of weeks used in the predictor, and W is the ordinal number of the week with 1 being the earliest week.

Weighted Weekly Average. The weighted weekly average predictor accounts for trends that change over time, such as the overall growth of the number of tweets. It uses the same scheme as the weekly average predictor, but weighs more recent weeks higher than less recent ones according to the following formula:

$$P(t) = \frac{\sum_{i=1}^{n} (n-i+1)*V(t-(i*W))}{\sum_{i=1}^{n}(i)}$$

Here, P(t) is the traffic volume prediction at time t, V(t) is the actual observed traffic at time t, n is the total number of weeks used in the predictor, and W is the ordinal number of the week with 1 being the earliest week.

Outlier Elimination. The outlier elimination model removes outlier volume values from the weighted average computation. This method detects outliers by comparing the difference between the estimated and observed traffic volume with the standard deviation of the estimate for that frame. There are two types of outlying values that are removed: holes and spikes.

Holes, periods with a volume of 0, are caused by a problem with the Twitter

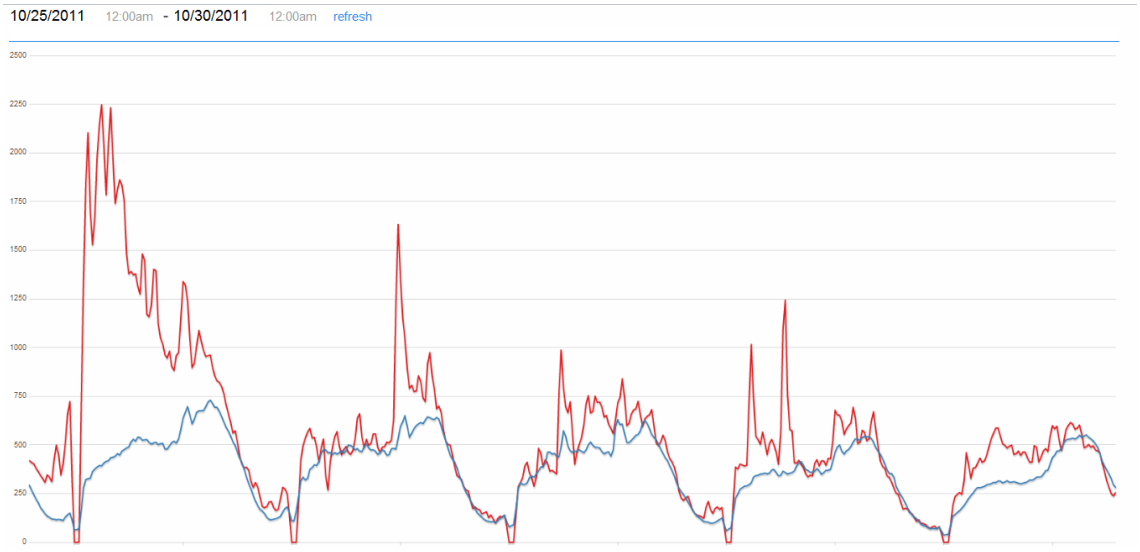

Figure 2.6: The actual (red) and model predictor (blue) time series for five days of the total volume dataset.

API and don't actually reflect a lack of twitter posts about Netflix. When the predictor encounters a hole, the predicted value is set to the current model value and standard deviation is not updated.

Spikes occur when the volume of a period is more than 2 standard deviations higher than the current model value. A model standard deviation value that includes all values is tracked to determine the standard deviation that defines a spike in the prediction. However, for the standard deviation calculation that is used for monitoring and calculating weighted average, spike values are are replaced by the current model value.

2.2.3.2 Trend Predictor

The trend predictor calculates an adjustment for each traffic volume estimate based on previous values.

Exponential Smoothing. Single Exponential Smoothing[15] constructs the smoothed traffic volume prediction S by weighting the more recent previous value A_{t-1} and previous actual values A_{t-n} based on how long ago they occurred with a predetermined smoothing factor α. The following equation is used for $t > 1$ and $0 < \alpha < 1$:

$$S_t = \alpha A_{t-1} + (1 - \alpha) S_{t-1}$$

The most recent previous value A_{t-1} is given a weight of α. Then the remaining weight is split between values before t-1 with the same formula.

Double Exponential Smoothing[16] extends Single Exponential Smoothing by taking into account the trend of the previous values b_t. For $t > 1$, $0 < \alpha < 1$, and $0 < \gamma < 1$;

$$S_t = \alpha A_{t-1} + (1 - \alpha)(S_{t-1} + b_{t-1})$$

$$b_t = \gamma(S_t - S_{t-1}) + (1 - \gamma) b_{t-1}$$

The trend predictor calculates smooth and trend values using Double Exponential Smoothing with $\alpha = 0.25$ and $\gamma = 0.25$ and then those values are used in the trend monitor to detect sudden spikes in traffic. See Appendix A.2 for more information about Exponential Smoothing and how the weighting constants α and γ were chosen.

2.2.4 Monitors

The goal of the monitors is to create a log of the events that are being indicated by the methods. Each method has one or more monitors. For each time period, a monitor compares the actual value to the predicted value and determines if there is an outage. This work uses two of the monitors implemented in SPOONS:

Figure 2.7: The actual (red) and trend predictor (blue) time series for five days of the total volume dataset.

model and trend.

2.2.4.1 Model Monitor.

The model monitor detects events based on the difference between the model value from a model predictor and the actual value from the analyzer. If the difference for a frame exceeds the standard deviation calculated by the model predictor by more than the allowed threshold then the time frame is treated as indicating an outage. The standard deviation threshold can be tuned any number that is a multiple of 0.05 between 0.25 and 4.[3]

2.2.4.2 Trend Monitor.

The trend monitor detects events based on an actual value exceeding the estimated value created by the trend predictor (Section 2.2.3.2) by more than

[3]Monitor tuning is descibed in Section 3.3.1.

the allowed threshold. This is determined using the equation:

$$ActualVal_t >= SmoothVal_{t-1} + ThresholdMultiplier * TrendVal_{t-1}$$

The threshold multiplier can be tuned any number that is a multiple of 0.05 between 1 and 10.[3] This monitor was inspired by the way Lechvenko et al.[9] determine outages.

2.3 Output

2.3.1 Alert Generation

At the end of each time frame, each monitor determines whether or not the Netflix-related Twitter traffic during that frame signifies an outage event. If the monitor reaches this conclusion, it triggers an alert and contacts Netflix engineers with a brief email specifying the time of the alert and the reasons why the alert was raised. From there, the SPOONS UI extracts this data and plots it on the traffic time line.

2.3.2 User Interface

The ultimate goal of the user interface is to provide an always-accessible platform for quick analysis of outage signals and other anomalous tweet behavior. Every element in the UI provides some piece of useful quickly-parsable information to the user.

Figure 2.8 shows a screen shot of the current user interface. The main screen of the UI accepts a user specified time period and displays information about that time period using three components: a time series chart that can display

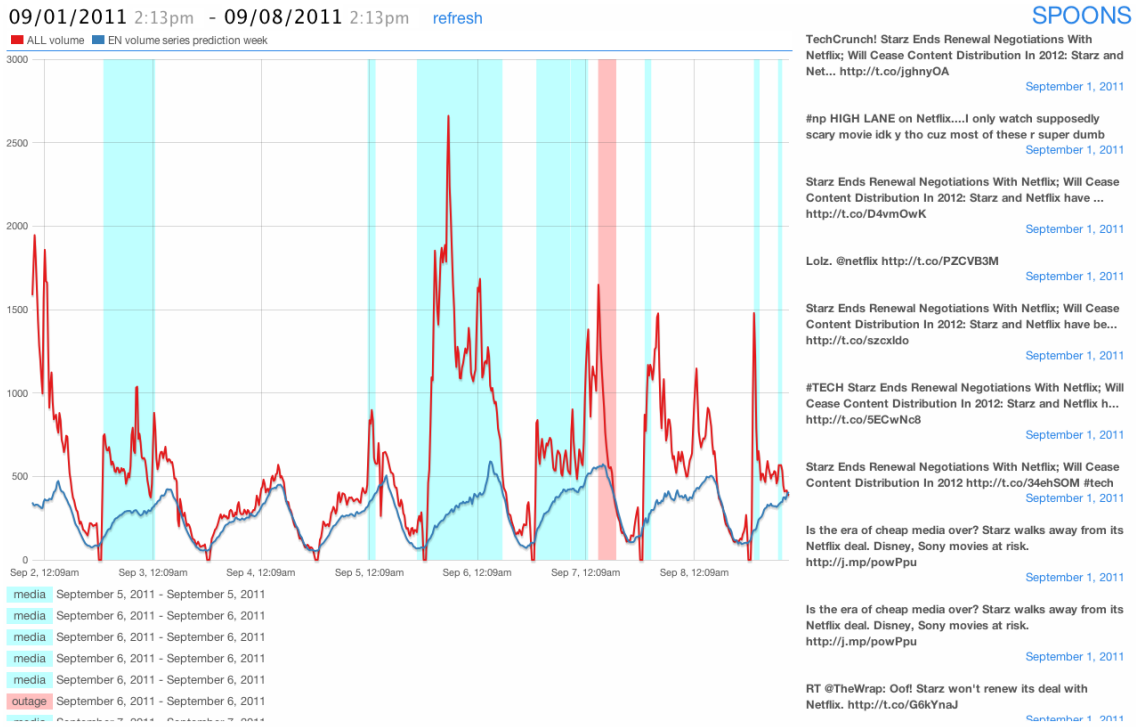

Figure 2.8: The user interface depicting multiple time series (predicted traffic vs. actual traffic), media and outage events, and a list of relevant tweets for a given time period.

any of the time series that are stored in the SPOONS database; an event log which displays times for both outage and media events; and a sampling of stored tweets.

Time Series Chart. The time series chart provides a graphical representation of the time series information stored in the SPOONS database such as actual values for each time period, expected value predictions, detected outage events, and reported media and outage events. This allows Netflix engineers to choose what time series data to display and then quickly scan for anomalous behavior and detect unusual tweet patterns. All of the events reported by researchers or Netflix engineers are color-coded by type (blue for media events, red for outages) and overlaid onto the chart to provide additional contextual information.

Event Log. The event log displays all events within the currently selected time range. Information available for each event includes type (media, outage, etc.), confirmation-status (e.g. whether Netflix engineers confirmed an outage), duration, start and end times, severity, and any user-supplied explanatory notes. The event log also accepts and stores new information about events, which means Netflix engineers can report new events, confirm detected events, and supply relevant notes.

Tweet List. Whenever an event is detected. Netflix engineers need to figure out the underlying cause of the alert. The chart functions as time range control for the tweet list. Through a simple click-and-drag gesture, users are able to narrow the range from which the tweet list pulls its tweets. Selecting an event in the event log will also narrow the time that the list pulls tweets from by randomly selecting tweets within the event time period. As the engineer scans

through outage indicating times, they are able to get a better idea of the general concepts that are being expressed during the possible outage time period.

Chapter 3

Detection Evaluation

The detection evaluation is an evaluation of how well detected events correspond to actual outage events reported by Netflix. This evaluation uses the same data set for both configuration and evaluation. So this is an ideal evaluation that determines how well the methods can create a time series that has higher values during outage times and lower values during normal times. This chapter describes the detection evaluation procedure, the results of the detection evaluations will be shown in Chapters 7 and 14.

3.1 Description of the Data Set

Detection evaluations are configured and evaluated using data from the time periods shown in Table 3.1. The evaluation time period starts after the beginning of the tweet collection time period because the set of reported events from Netflix starts in March. The two months of tweet data before the evaluation period begins are used to allow the predictors to create smooth consistent time series that represents normal traffic patterns.

Time Period	Begin	End
Tweet Collection	January 20, 2011	January 20, 2012
Evaluation	March 14, 2011	January 20, 2012

Table 3.1: These time periods describe the time periods that the tweets and events used in this work occurred during.

Tweet Collection. The SPOONS system has the entire collection of tweets that were posted on Twitter during the tweet collection time period and contain the word "netflix".

List of Outages. Netflix provided a list of 196 outage events that occurred during the evaluation time period, including the approximate start and end time of each event. These are the reported events that define the outages that the methods are trying to detect.

3.2 Evaluation Metrics

How effective a method is at outage detection is measured using the following metrics:

- **Precision**: the percent of the alerts generated by a method that occurred during an outage event.
- **Recall**: the percent of the reported events that were caught.

Recall reports how many of the reported outages the method is able to detect while precision reports how well the method limits its alerts to only outage times. Each of these metrics can be calculated in two ways:

- **Intersection**: considers whether the detected events overlap with the reported events. Any intersection between a reported outage time period and a detected outage time period is considered a true positive; any reported outage event that doesn't overlap with any of the a detected outage events is a false negative; any detected outage that has no intersection with the events reported by Netflix is a false positive.

- **Minute**: considers how much of the detected events overlap with the reported events. Any minute that occurs during the reported time period for an outage and the time period of a detected outage is considered a true positive; any minute that occurs during a reported outage event, but not during any detected outage events is a false negative; any minute that occurs during a detected outage, but not during any reported outages is a false positive.

3.2.1 Method Ranking

The goal of the recall metric is to evaluate a set of detected events generated by an analysis method and determine:

> If each of those events had been detected in real time, during the time period of the evaluation, and had sent Netflix engineers alerts when they started, what percent of the reported events would they have been alerted to?

The recall metric using the intersection calculation determines this by reporting the percent of the detected events overlap with the reported events; the percent of the reported events that have an alert during the time of the outage.

However, the intersection calculation has a major weakness: as the length of events increases, both of the metrics that the system is being evaluated on

increase. For example, if a monitor reports one event starting at the beginning of time and ending at the end of time, then all reported outages will intesect with that event and that event will intesect with at least one reported outage resulting in 100% precision and recall. So evaluating the analysis method with metrics only calculated using the intersection calculation encourages the methods to make fewer, longer events.

To compensate for this, the minutely calculation is used to calulate the precision metric which then acts as a limiting factor. Netflix has specified a precision of about 0.5 to be a usable amount of false positive noise. The precision metric using the minutely calculation reports the percent of minutes during detected events that overlap with reported events. So to ensure that the events created overlap with reported outages for at least as much time as they don't, any methods that aren't able to achieve a minutely precision of at least 0.5 are disqualified for use.

Any methods with a minutely precision above 0.5 are ranked by their intersection recall and considered for use.

3.3 Evaluation Procedure

The detection evaluation has two parts: metric calculation and statistical significance comparison. First all the methods are evaluated using the metric calculation and monitor tuning procedures described below. Then the metric results are compared across methods to determine the strengths and weakness of each method and determine how statistically significant the differences in their results are.

3.3.1 Monitor Tuning

Monitor tuning is used to determine the best set of parameters to use in each of the monitors for each of the methods. This parameter configuration is defined by three parameters:

- **Threshold**: the value that determines how large of a spike indicates an outage. Thesholds are described for each of the monitors in Section 2.2.4.
- **Alert Resistance**: the number of frames that must indicate an outage before an alert is generated.
- **Recovery Resistance**: the number of frames that must indicate normal traffic before an outage event is closed.

Each method has a unique parameter configuration for each monitor. These parameters are configured by a process that iterates over a range of values for each parameter and finds the configuration that produces the highest ranking.

To determine the ideal configuration for the methods, they are tuned using the minutely evaluation which best evaluates whether the detected outages fit closely to the reported outages. Then events detected by the monitor are evaluated using the intersection evaluation to report how well the system detects outages.

This evaluation tunes the monitors using all of the events in the evaluation time period and then evaluates the same events using that configuration. This does not simulate how well these methods would have done in real-time during these months. Instead it measures the methods under the most ideal cases. Section 8 will evaluate the effects of running some of the methods in a real-time simulation.

3.4 Statistical Significance

In this work, the chi square test[1] is used to evaluate if there is a statistically significant difference between the results of two methods. The null hypothesis[1] is "Outage detection recall is not affected by the difference between these methods".

This calculation is based on the observed values table. The rows of the table are the methods that are being evaluated, the columns are the true positive and false negative intersection values from a method's best confusion matrix.

The following formula is used to calculate the chi square value of the table where O is the observed value and E is the expected value.

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

$$E = \frac{\sum (O_{row}) * \sum (O_{column})}{\sum (O_{table})}$$

The calculated chi square value can then be used to look up the p (probability) value in the chi square distribution table shown in Figure 3.1. The degrees of freedom value for a chi square value is $(m-1)*(n-1)$ where m and n are the dimensions of the observed values table. For the calculations in this evaluation procedure, these tables are 2x2 so the evaluation has 1 degree of freedom. In general, p values of 0.05 and less are considered statistically significant enough to reject the null hypothesis.[1]

Therefore, a p value of 0.05 or less on a table that contains two methods rejects the null hypothesis that says that outage detection recall is not affected by the difference between those methods. Which means that the difference in recall results between the two methods was significant, so one method is significantly better. In reverse, a p value greater than 0.05 means that the two methods being compared detected outages with about the same recall, so neither one is

Degrees of Freedom *(df)*	**Probability** (p)										
	0.95	0.90	0.80	0.70	0.50	0.30	0.20	0.10	0.05	0.01	0.001
1	0.004	0.02	0.06	0.15	0.46	1.07	1.64	2.71	3.84	6.64	10.83
2	0.10	0.21	0.45	0.71	1.39	2.41	3.22	4.60	5.99	9.21	13.82
3	0.35	0.58	1.01	1.42	2.37	3.66	4.64	6.25	7.82	11.34	16.27
4	0.71	1.06	1.65	2.20	3.36	4.88	5.99	7.78	9.49	13.28	18.47
5	1.14	1.61	2.34	3.00	4.35	6.06	7.29	9.24	11.07	15.09	20.52
6	1.63	2.20	3.07	3.83	5.35	7.23	8.56	10.64	12.59	16.81	22.46
7	2.17	2.83	3.82	4.67	6.35	8.38	9.80	12.02	14.07	18.48	24.32
8	2.73	3.49	4.59	5.53	7.34	9.52	11.03	13.36	15.51	20.09	26.12
9	3.32	4.17	5.38	6.39	8.34	10.66	12.24	14.68	16.92	21.67	27.88
10	3.94	4.86	6.18	7.27	9.34	11.78	13.44	15.99	18.31	23.21	29.59
	Nonsignificant								Significant		

Figure 3.1: The chi square distribution table[1] that is used to determine the p value that corresponds to a calculated chi square value.

significantly better.

Part 2

Volume Analysis

Chapter 4

Specific Problem: Time Series Analysis of Raw Tweet Filtering

4.1 Problem Definition

This part covers a subset of the analysis methods implemented in the SPOONS system that specifically aim to detect outages using only time series analysis of Netflix-related tweet volumes that can be determined by filtering raw tweets based on information that is received directly from the Twitter API. The goal of the filtered volume methods is to enhance the real-time responsiveness of the system by pulling a volume measurement for a time period from the database as soon as as that period ends. This means that the results of these volume measurements are not blocked by any preprocessors.

4.2 Survey and Analysis of Related Research

4.2.1 Keyword Tweet Filtering

Levchenko et al.[9] implemented a system that uses tweets to detect outages in several widely used web services such as Amazon, Gmail, Google, PayPal, Netflix, Youtube, Facebook, Wikipedia, and Flickr. They say that one advantage of using tweets over more direct service monitoring systems is that the Twitter users are acting as millions of sensors who have a large breadth and flexibility of in the definition of an outage.

Their system uses two predicates to determine if a tweet is reporting an outage: For any $x \epsilon X$, where X is a predetermined set of service names,

Let *IsDown* = {A tweet contains "x is down"}. (e.g "Facebook is down.")

Let *Fail* = {A tweet contains "#xfail", or it contains "#x" and "#fail"}. (e.g "My movie won't stream! #Netflix #fail")

A sequential outline of the method they implemented that describes how these predicates are used to detected outages in the list of services defined by X is shown below.

1. Pull tweets from Twitter API.
2. Filter out all tweets that don't pass the*IsDown* predicate.
3. Split the remaining tweets into time periods and count the volume for each time period.
4. Apply Exponential Smoothing to the time series and store the expected smooth and trend values for each period.

5. Detect events by comparing the expected values with the actual values.

6. Limit the number of false positives by requiring the occurrence of a tweet to pass the

   emphFail predicate within 60 minutes of the detected start time.

Metrics and Evaluation. Levchenko et al. analyzed the results of several different predicate options by observing the most common phrases used in tweets that were posted during an outage. They also did trial and error experiments to determine which Exponential Smoothing parameters would result in the detection of the most outages. The system was run and evaluated using over 1,556 entities and tweets that occurred during 2009. The *IsDown* predicate alone detected 5358 events, however when the *Fail* predicate was added the number of events were reduced to 894 events. Manual inspection determined that the *Fail* predicate reduced the number of false positives. There were three evaluation procedures run on the results:

- (a) a comparison of the events detected by their system to a list of 8 outages that was compiled using Google News articles;
- (b) an analysis of the top 50 detected events;
- and (c) an analysis of 50 randomly selected detected events.

The metrics used in these evaluations are recall, precision, and time to detection. The results of these evaluations are shown in Table 4.1.

Contributions. The contributions of the work done by Levchenko et al. are:

- a demonstration that even simple techniques can identify important events;

Evaluation	Recall	Precision	Time to Detection
(a) 8 Known Outages	1.00	X	10 to 50 minutes
(b) Top 50 Detected Events	X	0.96	X
(c) Random 50 Detected Events	X	0.70	X

Table 4.1: The results of the evaluations done by Levchenko et al.[9]

- two predicates that are commonly found in tweets about service outages;
- and a method for detecting a spike in a volume time series.

Analysis of Solution and Application to SPOONS. Levchenko et al. were only able to validate a subset of their detected events because a full precision and recall validation would have required a list of outages during 2009 for every company they were monitoring. So while the events they were able to verify indicate that the system can detect outages, the full effectiveness of their method is still largely unknown.

However, the SPOONS system is able to fully evaluate the effectiveness of its methods because Netflix consistently updates it with a full list of outages. So to evaluate the effectiveness of this method in relation to the other methods in the SPOONS system, the *IsDown* predicate and Exponential Smoothing spike detection are integrated into the SPOONS system as the keyword volume analysis method and the trend monitor.

4.3 Contributions of this Part

The contributions being claimed in this part of the work are:

- the comparison between 4 types of volume filtering;

- and 4 analysis methods that are acceptable for use by Netflix engineers.

Chapter 5

Volume Analysis Methods

The methods presented in this section analyze the volume of a subset of the tweets in the SPOONS database over time. The four volume analysis methods described in this section are the total volume analysis method, English volume analysis method, keyword volume analysis method, and linkless volume analysis method. Figure 5.1 shows how the data sets for each of the methods overlap and supports the tier descriptions below.

5.1 Preprocessor

The volume analysis methods don't use a preprocessor. This simplifies their process and decreases their runtimes.

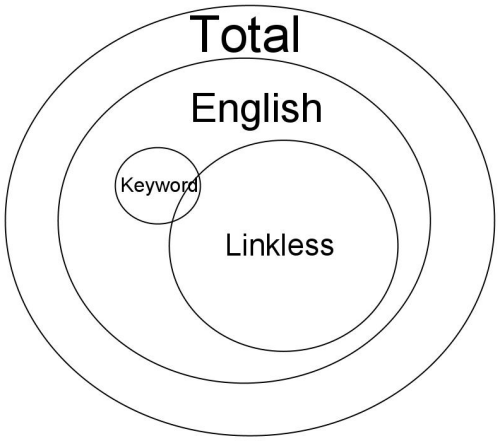

Figure 5.1: A venn diagram showing the overlaps between the data sets for each of the filtered volume methods.

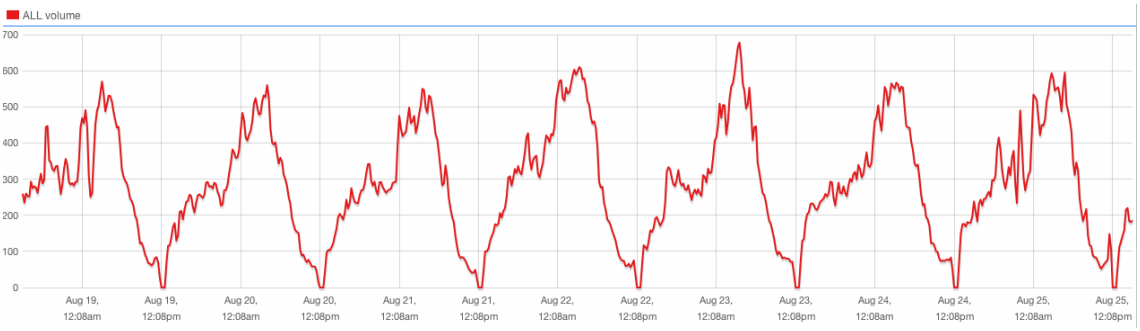

Figure 5.2: A 7 day period (August 18 - August 25) with no media events or serious outage.

5.2 Counter

5.2.1 Tier 1: Unfiltered Volume

5.2.1.1 Total Volume Filtering

The total volume analysis method uses the entire data set of tweets that the SPOONS system pulls from Twitter. Therefore the volume values in the total volume time series are a count of the total number of tweets posted on Twitter

during the time period and contained the word "Netflix".

This time series follows a fairly regular pattern when there aren't any Netflix related events occurring. The pattern mostly repeats daily, but at times does contain some weekly trends. Figure 5.2 depicts one week of normal traffic. As seen from the figure, the traffic reaches a daily low at 2am, slowly rises to an initial peak at 4pm, and a second peak at 7pm as East and West Coast customers arrive home from work (all times PST).

5.2.2 Tier 2: Language Filtered Volume

5.2.2.1 English Volume Filtering

The English volume analysis method uses the subset of the tweets in the total volume data set that are in English. The language of a tweet is determined using the language value returned by the Twitter API. Since all of the content filtered methods are restricted to only English tweets, this method enables the results of the methods to be compared to the base line of total English volume.

5.2.3 Tier 3: Content Filtered Volume

The content filtering methods described below will filter out tweets from the English volume data set based on attributes of their unedited contents. The keyword filtering section describes a white listing filtering method that defines attributes of a tweet's content that should be included in the analyzed data set. The linkless volume section describes a black listing filtering method that defines attributes of a tweet's content that should not be included in the analyzed data set.

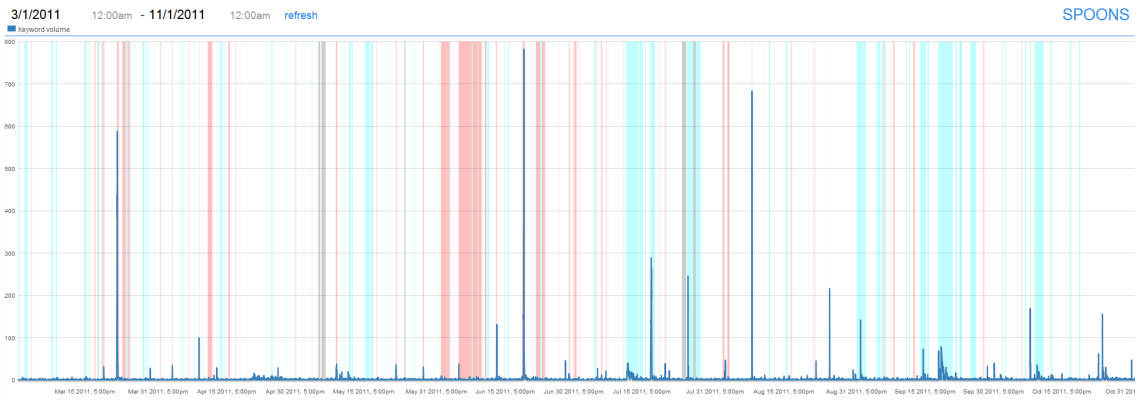

Figure 5.3: Volume of Netflix-related Twitter traffic containing the phrase "is down" between January and November of 2011.

5.2.3.1 Keyword Volume Filtering

The keyword volume analysis method calculates the volume of the subset of the English volume data set that contain the phrase "is down", similar to the *IsDown* predicate defined by Levchenko et al[9]. Figure 5.3 is a graph of the traffic pattern of all of the tweets in the keyword volume data set from January until November of 2011.

An analysis run in November of 2011 showed that "is down" occurs in 3% of the tweets posted during an outage event, but in less than 1% of the tweets posted during a media event. Therefore, in general, the outage events will spike 3 times higher than media events.

5.2.3.2 Linkless Volume Filtering

The linkless volume analysis method calculates the volume of the subset of the English volume set of tweets that do not contain a URL. This method assumes that the presence of a URL indicates that the tweet is about a media story and is not a tweet reporting unknown outages. This method reduces the number of

Figure 5.4: Linkless traffic (red) and total Netflix-related Twitter traffic (green) between August 25, 2011 and September 3, 2011

false positives triggered by spikes in traffic caused by people posting about media stories.

Graph 5.4 displays an example of a time period where the linkless method only reduced an outage by less than 10% but reduced a media event practically to the level of normal traffic.

An analysis run in November of 2011 showed that in the English data set, URLs occur in 46% of the tweets posted during a media event, but in only 21% of the tweets posted during an outage event. Therefore, in general, this method will reduce media spikes by about half, while only diminishing outage spikes by about a fifth.

5.3 Predictors and Monitors

All of the volume analysis methods use both the model and trend predictors and monitors.

Chapter 6

Prediction Evaluation

In this chapter, each of the volume analysis methods is evaluated on how well they reveal normal traffic patterns. Since the two main causes of traffic spikes in this data set are outage and media events, this part also evaluates how well the methods differentiate between the two types of events.

6.1 Evaluation Procedure

This procedure evaluates methods based on their ability to follow normal and media traffic and diverge from outage traffic. To evaluate this metric, the mean square error (MSE) between what the predictor determines the volume should be and what the volume actual is during event type time periods is calculated. The evaluation metrics for these pattern identifications are:

- **Outage Deviation**: $\frac{MSE(ActualOutage,PredictedOutage)}{MSE(ActualNormal,PredictedNormal)}$
- **Media Deviation**: $\frac{MSE(ActualMedia,PredictedMedia)}{MSE(ActualNormal,PredictedNormal)}$

Where $MSE(A,B) = \frac{\sum_{i=1}^{n}(A_i - B_i)^2}{n}$, n is the number of values in each of the sets, $Actual[EventType]$ represents the subset of the time series of volumes created by a counter that occur during the event type time periods; and $Predicted[EventType]$ represents the subset of the time series o volumes created by a predictor that occur during the event type time period.

6.1.1 Predictor Evaluation

There are 5 model predictors described in Section 2.2.3.1: Day Offset, Week Offset, Weekly Average, Weighted Weekly Average, and Outlier Removal. However, to simplify the evaluation of the analysis methods, all of the model predictors were evaluated based on their ability to predict normal traffic patterns in the total volume data set. From that the best model predictor was chosen to be used in the rest of the analysis method evaluations and just called "the model predictor."

This evaluation was run in November of 2011 using a sets of tweets, outage events, and media events between January and November of 2011. Table 6.1 shows the prediction evaluation of all of the model predictors. This evaluation shows that all of the predictors are able to differentiate between normal and other types of traffic by at least two orders of magnitude. However, since the outlier removal model predictor most closely followed the normal traffic pattern, it is the model predictor used in the rest of the evaluations and is referred to as "the model predictor" from now on.

Model Predictors	Outage Deviation
Outlier Removal	**3.80**
Weighted Weekly Avg	3.79
Weekly Avg	3.32
Day Offset	2.63
Week Offset	2.23

Table 6.1: The results of the model predictor prediction evaluation.

Method	Outage Deviation	Media Deviation
Keyword	83.19	18.80
Linkless	4.33	40.33
English	3.83	76.18
Total	3.80	57.99

Table 6.2: The results of the filtered volume analysis method prediction evaluation

6.2 Results

Table 6.2 show the results of the prediction evaluation run on each of the filtered volume analysis methods. It shows that every method is able to recognize that the MSE between expected traffic and outage traffic is at least 3 times larger than the MSE between expected traffic and normal traffic.

By applying tier 2 filtering, English language filtering, the outage deviation is increased, however the media deviation is also increased. This means that the English volume method will be more likely to detect when an event is happening, but not better at identifying what type of event it is. The English volume method probably follows a clearer pattern of normal traffic than the total volume method because the non-English tweets are posted more sporadically which adds inconsistent noise to the normal total volume traffic signal. It doesn't increase the differentiation between media and outage times because nothing about a tweet being in English indicates what the tweet is about.

Both of the tier 3 filtering methods, keyword volume and linkless volume, decrease the magnitude of difference between media and normal time periods while increasing the difference between outage and normal time periods. Therefore they not only decrease the likelihood of a media event causing a false positive, but also increase the likelihood of an outage event standing out and triggering a true positive.

All of these initial results indicate that each of the filtering methods is able to detect outages more effectively than the previous tier.

Chapter 7

Detection Evaluation

This detection evaluation evaluates how well the events detected by the volume analysis methods correspond to actual outage events reported by Netflix.

7.1 Evaluation Procedure

Since the detection evaluation is the main evaluation used to compare and rank all SPOONS analysis methods, the description of the evaluation procedure was shown as part of the Introduction in Chapter 3.

7.2 Results

Table 7.1 shows the best threshold and metrics determined by the detection evaluation run with each of the monitors on each of the filtered volume analysis methods. The best results for each of the methods are highlighted. See Appendix C for the complete set of result information.

Method	Monitor	Threshold	Minutely Precision	Intersection Recall
Linkless	**Trend**	**3.60**	**0.525**	**0.408**
Keyword	**Trend**	**5.00**	**0.532**	**0.347**
English	**Trend**	**5.15**	**0.504**	**0.342**
Total	**Trend**	**6.75**	**0.523**	**0.260**
Keyword	Model	1.50	0.501	0.112
Total	Model	-	-	-
Eng	Model	-	-	-
Linkless	Model	-	-	-

Table 7.1: The results of the filtered volume analysis method detection evaluation.

Method	Outages Alerted	Outages without Alerts
English	67	128
Total	51	145

Table 7.2: The observed values for comparing the total and English volume analysis methods.

All four of the methods were able to achieve the required precision when they used the trend monitor. The keyword analysis method was also able to get a precision greater than 0.5 with the model monitor, but had a higher recall with the trend monitor.

These results indicate that each additional level of filtering increased the outage detection recall. So increasing the filtering applied to the set of tweets being observed increases the difference between normal and outage traffic patterns.

To determine if the difference between the results from different levels of

Method	Outages Alerted	Outages without Alerts
Keyword	68	116
English	67	128

Table 7.3: The observed values for comparing the English and keyword volume analysis methods.

Method	Outages Alerted	Outages without Alerts
Linkless	80	129
English	67	128

Table 7.4: The observed values for comparing the English and linkless volume analysis methods.

Method	Enhanced Method	Chi Square Value	p Value
(1)Total	(2)English	3.10	$0.05 < p < 0.10$
(2)English	(3)Keyword	0.01	$0.90 < p < 0.95$
(2)English	(3)Linkless	1.83	$0.10 < p < 0.20$

Table 7.5: The chi square results for the filtered volume analysis methods.

filtering is actually significant, the chi square test is run on the the recall values from the confusion matrices that were used to calculate each of the methods' best metrics. These matrix values are shown in Tables: 7.2, 7.4, and 7.3 and the resulting chi square values are shown in Table 7.5.

The p value for the comparison between the total and English methods is between 0.05 and 0.10 which is right on the boundary between significant and not significant. One reason that the English volume analysis method would perform better than the total volume analysis method is that the non-English tweets are less common then the English tweets and from a larger set of time zones. So they probably don't affect the general pattern of the total time series and just add noise when they do appear.

The keyword volume analysis method only caught one more outage than the English volume analysis method which means that it is more than 90% likely that the difference in effectiveness is just a coincidence. This means that while the time series for the two methods are very different and it's even likely that they caught different outages, they are both practically equally effective at detecting

outages.

The linkless volume analysis method had an even higher recall than the keyword volume analysis method, but it's improvement over the English volume analysis method is also not statistically significant. However, the linkless volume analysis method was able to detect 13 more outages than the English volume analysis method, which means even though it's not above the $p = 0.05$ threshold for being significant there's still at least an 80% probability that it is a better method.

Chapter 8

Real Time Detection Evaluation

As described in chapter 3, the detection evaluation procedure uses the same data set for both configuration and evaluation. So it is an ideal evaluation that determines how well the methods can differentiate between outage and normal/media time periods, i.e. create a time series that has higher volume or sentiment values during outage times and lower volume or sentimet values during normal times. This section describes an evaluation that limits the configuration tuning to only be run on data that would have been available if the system were running in real time.

This section is only a preliminary investigation into the effects of real time evaluation of the results; the results of this evaluation do not reflect the best that the system could have done in real time. There hasn't been enough research into different real time tuning options to determine what the best procedure would be. This evaluation determines how well the system would have done in real time if it were incrementally updated using the current tuning process (as decribed in Section 3.3.1) on all previous data and only previous data.

8.1 Evaluation Procedure

This evaluation procedure is the same as the ideal detection evaluation procedure except now each month is evaluated using a configuration that was tuned on the set of results that occurred before the start of that month. So for example, at the beginning of month 5, all monitors use a configuration that was created by tuning them using data from months 1-4.

The first month of predictions and events aren't included in the evaluation because they are a used to create the initial configuration.

8.2 Results

Table 8.1 shows a comparison of ideal and real time detection results for each of the volume methods. None of the methods were able to achieve the required precision when they were restricted to only using information that would have been available in real time. This indicates that while the ideal evaluations are good for comparing the relative effectiveness between methods, they don't accurately reflect how well the methods would have done if they were running during that time.

It is possible that with enough data, a constant ideal threshold will emerge. However, there are a large number of volatile factors that go into determining that threshold: number of Netflix users; number of twitter users; Netflix media events; and Netflix outage severity and frequency. Therefore, it seems unlikely that a static threshold will be the best indicator over time.

Method	Monitor	Tuning	Minutely Precision	Intersection Recall
Linkless	Trend	Ideal Real Time	0.525 0.454	0.408 -
Keyword	Trend	Ideal Real Time	0.532 0.294	0.347 -
English	Trend	Ideal Real Time	0.504 0.222	0.342 -
Total	Trend	Ideal Real Time	0.523 0.211	0.260 -

Table 8.1: The results of the real time method detection evaluation.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

9.1.1 Filtered Volume Analysis Methods

The filtered volume analysis methods analyze the volume of a subset of the tweets in the SPOONS database over time without any preprocessing.

Tier 1 Filtering. The total volume analysis method demonstrated the simplest form of volume based outage detection, the analysis of the total volume of tweets about Netflix. It was not as effective in detecting outages as the higher tiers of filtering, but did serve as a baseline to show the unique contribution of the English volume analysis method.

Tier 2 Filtering. The English volume analysis method only analyzes the volume of tweets about Netflix that are in English. The addition of English filtering caused the most significant change in outage detection effectiveness. This in-

dicates that the non-English tweets are creating noise in the total volume time series.

Tier 3 Filtering. Content filtering further narrows down the subset of tweets being analyzed by looking for characteristics of the tweets' contents. The keyword volume analysis method tries to identify outage indicating tweets by looking for the phrase "is down". The linkless volume analysis method tries to remove media indicating tweets by only observing tweets that don't contain links. The addition of content filtering did increase the recall in both the keyword and linkless volume analysis method, but not statisically significantly. However the linkless volume analysis method results do indicate that it is likely that the addition of linkless filtering is an improvement over the English volume analysis method.

9.1.2 Comparison to Research

Levchenko et al.'s[9] *IsDown* predicate and Exponential Smoothing spike monitoring are integrated into the SPOONS system as the keyword volume method and the trend monitor. However, the keyword volume method does not use Levchenko et al.'s *Fail* predicate to limit false positives because the precision of the method did not need to be improved and the addition of the *Fail* predicate would likely reduce the recall.

The keyword volume method using the trend monitor is able to achieve the required precision and seemed to slightly improved upon the results of the filtering tier below it. However those results are able to be met and likley improved by observing the linkless data set of tweets instead. Since the system implemented by Levchenko et al. monitors outages in several Internet services, it makes sense

that it would limit the tweets that it pulls from Twitter. However, since the methods in SPOONS are currently only monitoring Netflix, it is able to achieve higher metrics by analyzing a larger subset of the volume of tweets that are about Netflix.

9.1.3 Real Time Evaluation

The detection evaluation procedure used to compare the analysis methods' uses the same data set for both configuration and evaluation, so it evaluates the most ideal results that the methods could produce in real time. The real time evaluation limits the configuration tuning to only be run on data that would have been available if the system were running in real time.

The real time evaluation indicated that the methods are not as effective at detecting outages in real time as they are in the ideal evaluations. This is not shocking , but really points out that while the SPOONS analysis methods do work in real time, the tunning procedure that defines outage traffic is not fully optimized to be the most effective in real time. So there is definitely room for improvement in the real time system that isn't accounted for in the detection evaluation described in Section 3.

9.2 Limitations and Future Work

Meme Spikes. This work focuses on differentiating between media and outage traffic spikes. However, there is a third, much rarer, type of traffic spike: meme spikes. Meme spikes occur when someone posts a tweet that's not about a media story or service outage, but is funny or interesting and then a large number of

people start re-tweeting it. An example of a meme that appeared in the data set and caused a noticeable spike was a series of posts that contained the phrase, "If I'm ever a ghost, I hope the person I haunt has Netflix." This phrase was posted 3,621 times within 7 hours, that's about 500 tweets per hour about ghost desires. None of the methods in SPOONS directly attempt to remove these meme spikes because over the entire data set only 5 meme trends caused noticeable spikes which isn't a significant number of possible false positives to be worth trying to eliminate.

Keyword Analysis Improvements. Matsuo et al.[12] created a procedure for algorithmically determining the keywords in a document. This procedure could be run on a "document" consisting of tweets that were posted during outage times and updated every time Netflix reported official outage events in the system. This would allow for a larger number of keywords that would dynamically update to new ways of reporting outages over time.

Real Time Tuning Evaluations. The real time tuning evaluation done in this work was designed to show the difference in metrics between the ideal and real time evaluations. Another evaluation that could be done would be to determine a good length for the dynamic tuning periods and how many of the previous time periods to include in the configuration tuning. The goal of that evaluation would be to find a period length and number of periods that would smooth out any extreme outage thresholds but still account for changing trends.

Part 3

Sentiment Analysis

Chapter 10

Specific Problem: Detecting Outages Through Sentiment Analysis

10.1 Problem Definition

This part covers a subset of the analysis methods implemented in the SPOONS system that detect outages by looking for significant increases in negative sentiment. The theory is that when there is an outage in a Netflix service then the Netflix users will post tweets with negative sentiments to express their displeasure. This problem also includes the evaluation of what combination of sentiment estimation options produce the best sentiment estimations.

10.2 Survey and Analysis of Related Research

10.2.1 Twitter Sentiment Compared to Public Polls

The goal of the sentiment analysis method developed by O'Connor et al.[17] was to find a correlation between the overall sentiment on Twitter and the results of public polling. To determine the sentiment of the tweets, they mapped each tweet's words to a polarity (positive or negative) and then tracked the aggregate sentiment over time. A sequential outline of the method they created is shown below:

1. **Word Rating.** Positive and negative words are defined by by the subjectivity lexicon from OpinionFinder.
2. **Tweet Rating.** A message is defined as positive if it contains any positive word, and negative if it contains any negative word.
3. **Tweet Metadata Aggregation.** The sentiment of a time period is the ratio of positive tweets to negative tweets.
4. **Time Series Analysis.** Moving weighted average was used to smooth the time series of sentiment over time.

Metrics and Evaluation. This sentiment determination system was evaluated through a comparison to public opinion surveys. The evaluation was done using 3 topics: consumer confidence; presidential approval; and elections. The tweet and survey data sets for each topic consisted of:

- **Consumer Confidence**: how optimistic the public feels about the health of the economy and their personal finances.

  - Tweets: English tweets containing the keywords: economy; job; and jobs.
  - Surveys: The Index of Consumer Sentiment (ICS) from the Reuters/University of Michigan Surveys of Consumers and The Gallup Organizations Economic Confidence Index.

- **Presidential Approval**: how the public feels about President Barack Obama.
  - Tweets: English tweets containing the keyword: Obama.
  - Surveys: Gallup's daily tracking poll for the presidential job approval rating for Barack Obama over the course of 2009.

- **Elections**: how the public feels about the 2008 U.S. presidential election candidates.
  - Tweets: English tweets containing the keywords: Obama and McCain.
  - Surveys: A set of tracking polls during the 2008 U.S. presidential election cycle.

The metric used for evaluation was the cross-correlation between their calculated time series and plots made from the polling results. O'Connor et al. give multiple correlation values for some of the topics to show the effect that the smoothing time period has on the results.

Contributions. The main contribution of the work done by O'Connor et al. is the demonstration that even basic sentiment analysis can be used to determine public opinion on specific topics.

Analysis of Solution and Application to SPOONS. The polling system doesn't use highly advanced or complicated methods for sentiment analysis or time series creation, but their methods were effective enough to see the correlation results that they were looking for.

The system created by O'Connor et al. is similar to the SPOONS system because they both use English tweets that contain a topic identifying keyword and they both use lexicon word sentiment rating and weighted averaging.

10.2.2 Keyword Identification

Matsuo et al.[12] created a procedure for algorithmically determining the keywords in a document. Keywords are words that describe the main topics and purpose of the document. Their method is outlined sequentially below:

1. Break the document into sentences, remove stop words, stem each remaining word, and create a mapping of unique word (term) to sentence.

2. Determine the most frequent terms, up to 30% of total terms.

3. Cluster frequent terms or create co-occurrence matrix that calculates the co-occurrence frequency for every term to frequent term combination.

4. Calculate expected probability of frequent terms by dividing the sum of the total number of terms in sentences where g appears by the total number of terms in the document

   $p_g = n_g / N_{Total}$

5. Test each term's co-occurrence pattern with frequent terms to determine how much each term is biased by the frequent terms. This is done using the

chi square test where the null hypothesis is that "the occurrence of frequent terms G is independent from occurrence of term w."

In general,

$\chi^2 = \sum \frac{(O-E)^2}{E}$

Here, the observed value is the frequency with which term w appears with frequent term g. The expected value is what the frequency of w appearing with g would be if their occurrences were independent; the unconditional probability of a frequent term g multiplied by the total number of co-occurrence of term w and frequent terms G.

$O = freq(w, g)$

$E = p_g n_w$

So, for all unconditional probabilities of frequent terms $g \epsilon G$

$\chi^2 = \sum \frac{(freq(w,g)-p_g n_w)^2}{p_g n_w}$

Metrics and Evaluation. To evaluate the effectiveness of this keyword identification method, the keyword identification process was run on 20 technical artificial intelligence papers, and determined the top 15 words for each paper. Then the words from all the papers were combined and shuffled, and the authors of the papers were asked to select from the list terms which they think are important in their paper and choose 5 or more terms which they thought were indispensable.

The accuracy of their keyword identification was compared to 3 other methods: tf, tfidf, and KeyGraph and evaluated using the metrics:

- **Precision**: the number of keywords identified by the process that were also

Evaluation	Precision	Coverage	Frequency Index
Matsuo et al.	0.51	0.62	11.5
tfidf	0.55	0.61	18.1
tf	0.53	0.48	28.6
KeyGraph	0.42	0.44	17.3

Table 10.1: The results of the evaluation done by Matsuo et al.

identified by authors divided by the total number of keywords identified by the process.

- **Coverage**: the number of keywords identified by the authors that were also identified by the process divided by the total number of keywords identified by the authors.
- **Frequency Index**: the average frequency of the top 15 terms identified by the process.

The results of the evaluation are shown in Table 10.1. Their method was able to achieve comparable precision results with a slightly higher coverage. The average frequency of the words they identified was lower than the rest of the methods which indicates that the increase in coverage comes from the ability to detect keywords that don't occur frequently.

Contributions. The main contribution of the work done by Matsuo et al. is the creation of a method that systematically determines the keywords in a document.

Analysis of Solution and Application to SPOONS. This method of identifying keywords is built into the SPOONS system as a contextual valence shifting option in the sentiment processor. The keywords identified are given a weight that emphasizes their sentiment rating relative to their ranked chi square values.

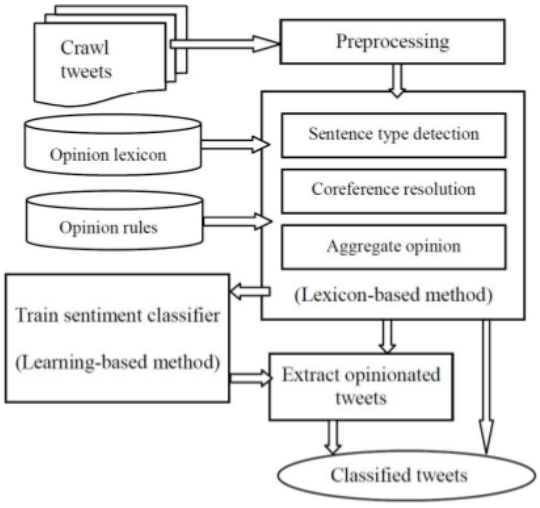

Figure 10.1: A digram of the sentiment analysis method developed by Zhang et al.

10.2.3 Combining Methods for Twitter Sentiment Analysis

One method of sentiment analysis of tweets is to use a lexicon of opinion words to identify the sentiment of tweets by the words that they have in common with the lexicon. The problem with this is that many strong opinion words used in tweets (such as "lovveeee" or ":)") aren't in traditional lexicons. Machine learning can be used to identify these more dynamic words, however traditional machine learning requires a large amount of manual tweet labeling and needs to be redone for each new application. Zhang et al.[23] attempt to eliminate the weaknesses of using lexicon or machine learning based sentiment analysis of tweets by combining the two methods. The architecture diagram of the method they developed is shown in Figure 10.1 and outlined sequentially below:

1. Preprocessing: retweet removal, abbreviation extension, link removal, sentence segmentation, and parts of speech tagging.

2. Employ an augmented lexicon based method to identify the sentiment of individual words.

3. Identify tweets that contain opinion words as opinion tweets.

4. Extract some additional opinionated indicating words. Use the chi square test to compare the expected and actual co-occurrences of words with known opinion words and mark words that are connected to known words of an opinion type as also indicating that opinion.

5. Use new opinion words to identify more opinion tweets.

6. Train a sentiment classifier with the identified opinion tweets.

This method assumes that all of the words within a tweet express the the same sentiment and that is also the sentiment of the tweet.

Metrics and Evaluation. Zhang et al. compare 3 versions of their method (each with a different combination of features) to 2 other high end sentiment analysis methods. The 5 methods being compared are:

- **ME**: a state-of-the-art learning-based method used by the website Twitter Sentiment, which uses Maximum Entropy as the supervised learning algorithm. The API of the sentiment classifier is publicly available.

- **FBS**: a lexicon-based method proposed for feature-based sentiment analysis.

- **AFBS**: the lexicon-based method described in #2 of the outline above, without the final SVM sentiment classifier.

Entity	ME	FBS	AFBS	LLS	LMS
Obama	0.756	0.878	0.868	0.880	0.888
Harry Potter	0.764	0.862	0.880	0.902	0.910
Tangled	0.630	0.794	0.818	0.720	0.882
iPad	0.628	0.642	0.692	0.764	0.810
Packers	0.620	0.720	0.736	0.756	0.780
Average	**0.679**	**0.779**	**0.798**	**0.804**	**0.854**

Figure 10.2: Zhang et al.'s evaluation of the positive, negative, and neutral classification results of each method using accuracy.

- **LLS**: an augmented method where opinion indicators that are identified in #4 of the outline above, are put into the original general opinion lexicon, and AFBS is run again. This method also does not use the final SVM sentiment classifier.

- **LMS**: the proposed method that utilizes all the techniques described.

The evaluations are done on 5 sets of tweets divided by topic: Obama, Harry Potter, Tangled, iPad, Packers. Each tweet set has about 40,000 to 400,000 tweets in it. 500 tweets are extracted from each set to be used for evaluation and the rest are used for training.

The metrics used to evaluate these methods are: accuracy, precision, recall, and F score. First they evaluate the accuracy of all three of their classification classes: positive; negative; and neutral. The results of this are shown in Table 10.2. Then they evaluate the positive and negative classification of each method using precision, recall, and F-score. The results of that are in Table 10.3. These evaluations show that LMS, the method by Zhang et al. with the most features is the most effective at identifying the sentiment of a tweet.

Contributions. The main contribution of the work done by Zhang et al. is the creation of an effective, unsupervised method for sentiment identification in tweets.

Query Entity	ME			FBS		
	Precision	Recall	F-score	Precision	Recall	F-score
Obama	0.170	0.202	**0.184**	0.564	0.556	**0.560**
Harry Potter	0.456	0.418	**0.436**	0.822	0.631	**0.714**
Tangled	0.454	0.510	**0.481**	0.927	0.627	**0.732**
iPad	0.263	0.294	**0.278**	0.360	0.352	**0.356**
Packers	0.247	0.327	**0.282**	0.550	0.445	**0.492**
Average	0.318	0.350	**0.332**	0.644	0.522	**0.570**

Query Entity	AFBS			LLS			LMS		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
Obama	0.522	0.582	**0.569**	0.569	0.708	**0.631**	0.595	0.708	**0.647**
Harry Potter	0.864	0.641	**0.736**	0.715	0.860	**0.781**	0.751	0.902	**0.820**
Tangled	0.884	0.679	**0.768**	0.636	0.851	**0.728**	0.827	0.928	**0.874**
iPad	0.436	0.356	**0.392**	0.576	0.802	**0.671**	0.636	0.831	**0.721**
Packers	0.672	0.484	**0.563**	0.551	0.714	**0.622**	0.629	0.753	**0.686**
Average	0.675	0.548	**0.605**	0.609	0.787	**0.686**	0.687	0.827	**0.749**

Figure 10.3: Zhang et al.'s evaluation of the positive and negative classification results of each method using precision, recall, and F-score[23]

Analysis of Solution and Application to SPOONS. The work done by Zhang et al. supports the basis for the tweet concurrence rating (TCR) word sentiment rating option described in Section 11.2 because it also relies on the assumption that all of the words within a tweet express the the same sentiment and that is also the sentiment of the tweet.

10.3 Contributions of this Part

The contributions being claimed in this part of the work are:

- 256 procedures for tweet sentiment estimation and evaluations of each of them;
- 4 analysis methods that are acceptable for use by Netflix engineers.

Chapter 11

Sentiment Preprocessor

Preprocessors are used to extract data from raw tweets that can be used by the counters. In this case, the sentiment preprocessor calculates a sentiment rating from the raw tweet and then the sentiment analysis method counters use those sentiment ratings to assemble time series that can be monitored for outage indications.

Below is a list of the preprocessing options that the sentiment preprocessor has available to it. They are listed in the order that they are run, if they are run.

11.1 Normalization

The normalization options try to remove words that don't contain any sentiment or an accurate sentiment and identify words that can be replaced with a placeholder that better describes their meaning. The options will be demonstrated using the tweet:

```
RT @gaballison: http://bit.ly/gadNWF :( Was trying to decide what
to watch on Netflix streaming and now I know: epic Doctor Who
```

```
marathon. #RIPBrig
```

URL Replacement. URL Replacement detects URLs in bodies of tweets and replaces URLS with a placeholder. This stops the rating and determination methods from trying to treat each URL as a unique word and instead combines them all under one word.

```
RT @gaballison: zlinkz :( Was trying to decide
what to watch on Netflix streaming and now I know: epic
Doctor Who marathon. #RIPBrig
```

Username Removal. All usernames can be removed and not used to identify the sentiment of a tweet because it isn't assumed that a username will express a consistent sentiment.

```
RT : zlinkz :( Was trying to decide
what to watch on Netflix streaming and now I know: epic
Doctor Who marathon. #RIPBrig
```

Movie/Show Title Replacement (TITLE). A tweet that says, "watching fighting duel of death", seems very negative at first. Many tweets mention titles of movies or shows (such as "fighting duel of death") that the poster is watching, just watched, or wishes Netflix carried. Words in these titles may carry sentiment that can confuse some of the analytical tools. Movie and show titles are identified based on a regularly updated table of titles that are offered through Netflix's streaming service. When identified, the titles are replaced with a static placeholder recognizable by the analysis methods.

```
RT : zlinkz :( Was trying to decide
what to watch on Netflix streaming and now I know: epic
ztitlez marathon. #RIPBrig
```

Emoticon Replacement (EMOT). Emoticon Replacement identifies emoticons that express strong sentiments and replaces them with a metaword so that their meaning will not be removed during the punctuation removal and they can be given a rating just like any other word.

```
RT : zlinkz emotzfrownz Was trying to decide
what to watch on Netflix streaming and now I know: epic
ztitlez marathon. #RIPBrig
```

Punctuation and Non-English Character Removal. All punctuation and characters not in the English alphabet are removed. This simplifies word extraction and comparison.

```
RT zlinkz emotzfrownz Was trying to decide
what to watch on Netflix streaming and now I know epic
ztitlez marathon RIPBrig
```

Lowercase Conversion. All letters are changed to lower case so that all evaluations are automatically case insensitive.

```
rt zlinkz emotzfrownz was trying to decide
what to watch on netflix streaming and now I know epic
ztitlez marathon ripbrig
```

Stopword Removal. Stopwords, or words that carry little or no semantic or sentiment information, are identified based on a static table of words mapped to levels. Stopwords are assigned levels which allow processes to use different sets of stop words. A full list of the stopwords used are in Appendix E. All words less than 3 characters long are also automatically considered stop words.

```
zlinkz emotzfrownz decide
watch streaming epic
ztitlez marathon ripbrig
```

Stemming. Stemming finds the root of a word. This allows words to be categorized by their roots which decreases the number of unique words being evaluated and emphasizes linguistic patterns. This preprocessor uses Porter's Stemmer for the English language [18].

```
zlinkz emotzfrownz decide
watch stream epic
ztitlez marathon ripbrig
```

11.2 Word Rating

The word rating options each try to assign a sentiment to each of the words that are remaining after all of the chosen normalization options have run. This is the first attempt at sentiment estimation that looks at each of the words individually and gives them the sentiment that they most commonly hold. The ratings are on a scale from 0 to 10 with 0 being the most negative sentiment and 10 being the most positive sentiment.

Afformative Norms of English Words (ANEW). ANEW[4] is Bradley and Lang's research corpus which was created to measure three aspects of human emotion (valence, arousal, and dominance) in regards to language. The ANEW values for each of the 1,034 words were determined through manual human labeling. The valence values from the ANEW corpus are used as the word sentiment rating for words in the corpus.

Valence Generation Search (VGS). This method assumes that positive words frequently occur with positive seed words and negative words frequently occur with negative seed words. The Yahoo Search API is used to search for a word

with an unknown valence and each of the three pairs of postive/negative seed word pairs: excellent/poor; positive/negative; and outstanding/terrible. Then the word rating is calculated using the following equation:

$$WordRating = 5 + 5 * \frac{\sum PositiveResults - \sum NegativeResults}{\sum AllResults}$$

This rating determination method used to calculate the rating of words that didn't already have a rating stored in the SPOONS database. It was run on about 50,000 words when it was first implemented. However, since then the public Yahoo Search API has been deprecated so the results of that initial run are being used as a static corpus.

Tweet Concurrence Rating (TCR). This method also assumes that positive words frequently occur with positive seed words and negative words frequently occur with negative seed words. However, instead of searching the Yahoo API for hits that contain both the seed word and the unknown word, this method searches the entire set of tweets in the SPOONS database for tweets that contain both the seed word and the unknown word. Then the word rating is calculated using the following equation:

$$BiasPct = \frac{\sum PositiveResults - \sum NegativeResults}{\sum AllResults})$$

$$StrengthPct = \frac{\sum PositiveResults - \sum NegativeResults}{\sum AllResults})$$

$$WordRating = 5 + 5 * \frac{(BiasWeight * BiasPct) + (StrengthWeight * StrengthPct * BiasPct)}{BiasWeight + StrengthWeight}$$

$$BiasWeight = 1, StrengthWeight = 1$$

The preprocessor will use either a combination of ANEW and VGS or TCR. ANEW and VGS are combined by first looking up the word in the ANEW corpus and if there's no rating then looking it up in the VGS results. If VGS also doesn't contain the word then the word is given a weight of 0.

11.3 Contextual Valence Shifting

The contextual valence shifting options modify the sentiment rating and/or weight given to a word based on the context of the tweet. These options use context hints about the importance and usage of the words to increase the accuracy of the sentiment estimation.

Negation (NEG). Negation uses the Stanford NLP Parser's[19] parts of speech tagger to identify a word that is being negated (negated word) and the word that is negating it (negating word). So for example, in the tweet "I am not happy", "happy" is the negated word and "not" is the negating word. Then it sets the weight of the negating word to 0 and adjusts the rating of the negated word using the following equation.

$$NewRating = CurrentRating + \frac{CurrentRating-5}{2}$$

Sentiment Holder Intensification (SHI). Adjectives and adverbs have been identified as sentiment holding words. The Stanford NLP Parser's[19] parts of speech tagger identifies these parts of speech and multiplies their rating by a constant intensification factor. An evaluation of the ideal intensification factor value is shown in Section 12.2.3.3.

Keyword Emphasis (KE). Keyword emphasis identifies keywords in the entire data set of tweets in the SPOONS database. These words are considered important words in tweets about Netflix. The theory is that words that are important in the overall meaning of tweets about Netflix will also be important in the valence determination of any tweet that they occur in. This option uses the keyword identification method developed by Matsuo et al[12] to calculate a chi

square value that represents the strength of the keyword in the set of tweets that occur during the tweet collection time period (defined in Section 3.1). Once the chi square value is computed for each word in the corpus, it can be used to adjust the weight that a word has in the sentiment determination calculation. Each of the words that occur more than 20 times are ranked in descending order based on the chi square value. Then the emphasis weight given to a word is calculated using the following equation:

$$WordWeight = 1 + \frac{MaxRank - WordRank}{MaxRank}$$

Where maxRank = 500. Since the goal is only to weight the most important keywords. Some manual calculations estimate that the 500 highest ranking keywords make up 56% of all word occurrences. This seems like a good cut off because that results in about the top fourth of words having their weight increased by 50% or more.

11.4 Tweet Rating Determination

Once the previous options have assigned individual word sentiment ratings and weights to each word in the tweet, the following options use that information to determine the sentiment of the tweet as a whole.

Means. These options just use the standard weighted mean calculations.

- **Arithmetic (ARIT)** $Sentiment = \frac{\sum Weight * Rating}{\sum Weight}$
- **Harmonic (HARM))** $Sentiment = \frac{\sum Weight}{\sum \frac{Weight}{Rating}}$
- **Quadratic (QUAD)** $Sentiment = \sqrt{\frac{\sum (Weight * Rating)^2}{\sum Weight}}$

Max Absolute Value (MAXA). This method looks for the most extreme sentiment expressed and uses that as the sentiment of the entire tweet.

$$Sentiment = max(|Rating - MidSentiment| * \frac{Weight-1}{2})$$

Chapter 12

Sentiment Estimation Evaluation

This section asseses a set of sentiment estimation options and determine which combination of options results in the sentiment rating that is the closest to manual human labeling.

12.1 Procedure

The ratings from each of the sentiment option configurations are compared to the manually labeled ratings from the survey by calculating the pearson correlation coefficient between the two data sets. The pearson correlation coefficient is calculated using the following equation:

$PearsonCoefficient = \frac{\mu_{ME} - \mu_M \mu_E}{\sigma_M \sigma_E}$

Where $\mu_X = \frac{\sum X}{N}$, $\sigma_X = \sqrt{\frac{\sum \mu - x}{N}}$, E = estimated sentiment, M = manually labeled sentiment, and N = total number of tweets estimated.

Every combination of the sentiment options described in Chapter 11 would result in over 10,000 sentiment processor configurations to evaluate. To reduce

the number of configurations being evaluated, some of the normalization preprocessing options are always included. These options are:

- URL replacement;
- username removal;
- punctuation and non-English character removal;
- stopword removal;
- and stemming.

By holding these preprocessing values constant, the keyword emphasis and tweet concurrence rating methods can also use these methods in their set up process which decreases the set up time and improves identification of unique, important words.

The remaining options will be evaluated by comparing the results of the combinations that do include them to the ones that don't. Figure 12.1 shows how the different options and option categories are combined to create the sentiment preprocessors that are evaluated.

12.1.1 Experimental Data Set

This experiment will compare sentiment ratings from each of the sentiment preprocessor's configurations with ratings manually assigned to tweets.

Since it is infeasible to manually rate all of the tweets in the SPOONS database, a subset of 1011 tweets have been chosen for this experiment. This set of tweets was evaluated based on its similarity to the total data set in the SPOONS database using 4 criteria:

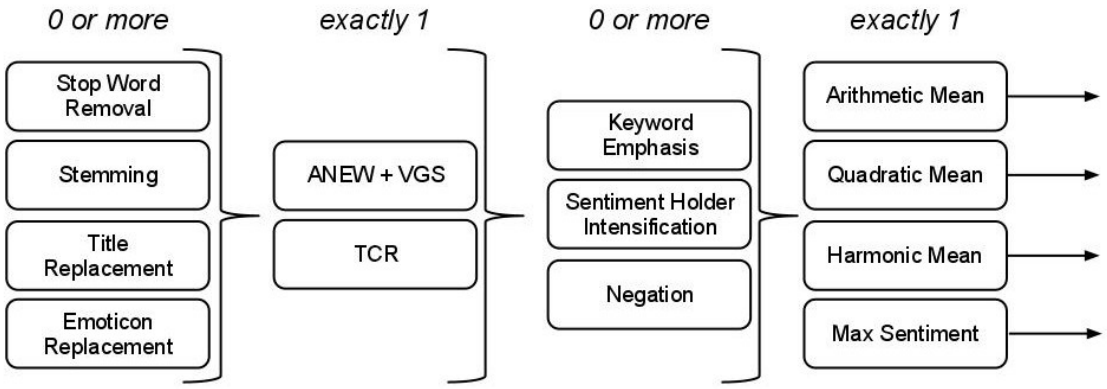

Figure 12.1: A diagram of the different combinations of sentiment options that will be evaluated.

- **Percentage of Media Tweets**

  Total: 25.6%

  Survey: 25.3%

- **Percentage of Outage Tweets**

  Total: 28.5%

  Survey: 28.6%

- **Distribution of Tweet Lengths**

  Total: Average = 86.3 characters, Standard Deviation = 38.5 characters

  Survey: Average = 93.4 characters, Standard Deviation = 36.6 characters

- **Distribution of Tweet IDs**

  Total: Average = 32500000, Standard Deviation = 24100000

  Survey: Average = 32200000, Standard Deviation = 24100000

Based on the results of the experimental tweet set evaluation shown above, it is reasonable to suggest that this subset of tweets is a good representation of the entire SPOONS data set of tweets.

12.1.1.1 Survey Result Editing

The manual rating of the experimental tweet set was done through a survey of California Polytechnic State University San Luis Obispo students. Cal Poly students from computer science courses each rated 50 of the 1011 experimental tweets on a scale from 0 to 10, negative to positive. Overall, there was an average of 7.4 ratings per tweet. Then the tweet ratings were processed using the following procedure:

1. Find the average rating per tweet.
2. Remove any ratings that are more than 4 away from the average.
3. Recalculate the average rating per tweet.
4. Calculate each rater's average distance from the current average rating.
5. Remove any rating from raters that have an average distance that is greater than 1.
6. Recalculate the average rating per tweet.
7. Remove all tweets that have less than 4 ratings.

This processing removes ratings that are outliers and the rates from people who generally rated tweets far from the average. After this processing, there are 994 tweets and each tweet's sentiment is calculated from 4 or more manual ratings that have a standard deviation that is less than 2.5.

12.2 Results

For each configuration of the sentiment preprocessor, the pearson correlation coefficient was calculated to compare the results of the processor with that configuration to the manual survey ratings. Table 12.1 shows the 20 configurations with the highest pearson correlation results out of 256 total results. The full set of results are in Appendix D.

The combination of emoticon replacement, TCR, negation, keyword emphasis, and arithmetic mean yielded the highest pearson correlation coefficient of 0.388. Unfortunately this coefficient does not indicate a high level of correlation between the manual ratings and the ratings from the sentiment preprocessor. The next two chapters use this sentiment preprocessing configuration and look into whether or not these ratings are accurate enough to detect outages.

The rest of this section describes the results and gives more information about the configurable sentiment options.

12.2.1 Normalization

Emoticon Replacement. The emoticon replacement was included in all of the top 13 results and in 16 of the top 20 results. The only results that it wasn't included in are repeats of combinations above them, just without the emoticon replacement. This means that in the top 20 results, the emoticon replacement improved every configuration that it was added to.

Title Replacement. The title replacement is not included in any of the top 20 results. This means that the addition of title replacement did not improve

Normalization	Word Rating	Contextual Valence Shifting	Determination	Pearson Correlation Coefficient
EMOT	TCR	NEG KE	ARIT	0.388
EMOT	TCR	NEG	HARM	0.384
EMOT	TCR	NEG	ARIT	0.380
EMOT	TCR	NEG KE	MAXA	0.378
EMOT	TCR	KE	ARIT	0.377
EMOT	TCR	NEG KE SHI	MAXA	0.375
EMOT	TCR	NEG KE SHI	ARIT	0.373
EMOT	TCR		HARM	0.371
EMOT	TCR		ARIT	0.367
EMOT	TCR	KE	MAXA	0.367
EMOT	TCR	KE SHI	MAXA	0.364
EMOT	TCR	KE SHI	ARIT	0.363
EMOT	TCR	NEG SHI	ARIT	0.360
	TCR	NEG KE	ARIT	0.360
EMOT	TCR	NEG KE	QUAD	0.359
EMOT	TCR	NEG	QUAD	0.352
EMOT	TCR	KE	QUAD	0.350
	TCR	NEG	HARM	0.350
	TCR	NEG	ARIT	0.349
	TCR	KE	ARIT	0.348

Table 12.1: The results of the top 20 most effective sentiment preprocessor configurations.

Normalization	Word Rating	Contextual Valence Shifting	Determination	Pearson Correlation Coefficient
EMOT	TCR	NEG KE	ARIT	0.388
EMOT	TCR	NEG	HARM	0.384
EMOT	TCR	NEG	ARIT	0.380
EMOT	ANEWVGS	NEG KE SHI	HARM	0.117
EMOT	ANEWVGS	KE SHI	HARM	0.116
	ANEWVGS	NEG KE SHI	HARM	0.116

Table 12.2: The results for the top 3 configurations for each of the word rating options.

sentiment estimation. This is possibly because the title replacement was too agressive and replaced words that held meaning or because in the case of the survey tweets, the sentiment expressed in titles happens to be indicative of the sentiment of the title. To try to reduce the effects of the title replacement, it was limited to only replacing titles that contained more than one word. This did improve the scores of the configurations that the option was included in, but not enough to put it in the top 20.

12.2.2 Word Rating

TCR was significantly stronger at word rating than the ANEW and VGS combination. No configuration using ANEW and VGS did better than a configuration using TCR. Table 12.2 shows the top 3 configurations for both TCR and ANEW/VGS. As it shows, the highest ANEW/VGS score is about 0.271 lower the the highest TCR score.

12.2.3 Contextual Valence Shifting

12.2.3.1 Negation

Negation was included in the top 4 results and in 12 of the top 20 results. The only results that it wasn't included in are repeats of combinations above them, just without negation. This means that in the top 20 results, negation improved every configuration that it was added to.

12.2.3.2 Keyword Emphasis

Keyword emphasis was included in the top result and in 12 of the top 20 results. This means that in the top configuration and in some other configurations, keyword emphasis correctly identified important words and emphasized their sentiment ratings which increased the accuracy of the sentiment estimation.

12.2.3.3 Sentiment Holder Intensification

As table 12.1 shows, sentiment holder intensification actually reduces the effectiveness of any configuration that it's added to. The only results that it was included in are repeats of combinations that did better without it. This means that in the top 20 results, sentiment holder identification decreased the effectiveness of every configuration that it was added to.

To determine if there was an intensification factor that would make the sentiment holder intensification option more effective, the evaluation was rerun with a range of intensification factors. The best results for each factor are in table 12.3. This table shows that as the intensification factor approaches 1, which weights sentiment holders the same as other words, the pearson correlation co-

Intensification Factor	Normalization	Word Rating	Contextual Valence Shifting	Determination	Pearson Correlation Coefficient
N/A	EMOT	TCR	NEG KE	ARIT	0.388
1	EMOT	TCR	NEG KE SHI	ARIT	0.387
1.1	EMOT	TCR	NEG KE SHI	ARIT	0.386
1.2	EMOT	TCR	NEG KE SHI	ARIT	0.385
1.25	EMOT	TCR	NEG KE SHI	ARIT	0.384
1.3	EMOT	TCR	NEG KE SHI	ARIT	0.383
1.4	EMOT	TCR	NEG KE SHI	ARIT	0.382
1.5	EMOT	TCR	NEG KE SHI	ARIT	0.381
1.6	EMOT	TCR	NEG KE SHI	ARIT	0.379
1.7	EMOT	TCR	NEG KE SHI	MAXA	0.378
1.75	EMOT	TCR	NEG KE SHI	ARIT	0.377
1.8	EMOT	TCR	NEG KE SHI	MAXA	0.377
1.9	EMOT	TCR	NEG KE SHI	MAXA	0.375
2.0	EMOT	TCR	NEG KE SHI	MAXA	0.375
3.0	EMOT	TCR	NEG KE SHI	MAXA	0.361
4.0	EMOT	TCR	NEG KE SHI	MAXA	0.346
5.0	EMOT	TCR	NEG KE SHI	ARIT	0.334
6.0	EMOT	TCR	NEG KE SHI	ARIT	0.323
7.0	EMOT	TCR	NEG KE SHI	MAXA	0.320
8.0	EMOT	TCR	NEG KE SHI	MAXA	0.314
9.0	EMOT	TCR	NEG KE SHI	MAXA	0.309
10.0	EMOT	TCR	NEG KE SHI	MAXA	0.308

Table 12.3: The effectiveness of sentiment holder intensification for a range of intensification factors.

efficient approaches the result of the configuration that doesn't use SHI. This means that any level of weighting on sentiment holders decreases the accuracy of tweet sentiment estimation.

12.2.4 Tweet Rating Determination

All four of the determination options are represented in the top 20 results. The best determination option seems to depend highly on the rest of the configuration.

The arithmetic mean determination is used in the configuration that achieved the highest pearson correlation so it is the one used in the next evaluation.

Chapter 13

Sentiment Analysis Methods

Analysis methods use a series of preprocessors, counters, predictors, and monitors to detect outages in Netflix services. The four sentiment analysis methods are the average sentiment analysis method, summed sentiment analysis method, average negative sentiment analysis method, and summed negative sentiment analysis method.

13.1 Preprocessor

The sentiment analysis methods use the most effective sentiment preprocessor configuration determined by the sentiment estimation evaluation to calculate the estimated sentiment for each tweet in the data set.

13.2 Counters

Each of the sentiment analysis methods aggregates the sentiment estimations for each tweet differently. However, all of the counters invert the sentiment scale so that 0 is positive and 10 is negative. This way the sentiment time series increases when there is an increase in negative sentiment which indicates a problem.

13.2.1 Average Sentiment

The average sentiment counter calculates the average sentiment value for a time frame using the following equation:

For all tweets in the frame,

$$FrameValue = \frac{\sum((MaxSentiment{=}10) - TweetSentiment)}{TotalTweetCount} * 10$$

By averaging the sentiment, the volume of the tweets posted during the time frame is extracted out so that the only information being observed is the sentiment. The resulting value is multiplied by 10 to increase the differentiation between normal and anomalous traffic and enhance the detail that is shown on the graphs on the user interface. This means that the values on this graph range from 0 to 10 times the max sentiment, i.e. from 0 to 100.

Figure 13.1 shows the average sentiment analysis method time series for the week of December 18-25, 2011. The time series are mostly linear with a bit of variability. The lines stay around 50. To convert this frame value to sentiment it needs to be divided by 10 and subtracted from the max sentiment: $sentiment = 10 - \frac{50}{10}$. So this graph indicates that on average the sentiments being expressed during that time frame is around 5, which is the mid point on the sentiment

Figure 13.1: The predicted (purple) and actual (green) time series for the average sentiment analysis method.

scale. When there is an outage or negative event then the time series values will increase.

13.2.2 Summed Sentiment

The summed sentiment counter calculates the sum of the sentiment values for a time period using the following equation:

For all tweets in the time period,

$$FrameValue = \sum ((MaxSentiment = 10) - TweetSentiment)$$

By observing the sum of the sentiment of the tweets, the volume of the tweets posted during the time frame affects the time series and can provide another level of information.

Figure 13.2 shows the summed sentiment analysis method time series for the week of December 18-25, 2011. The time series are both similar to the total

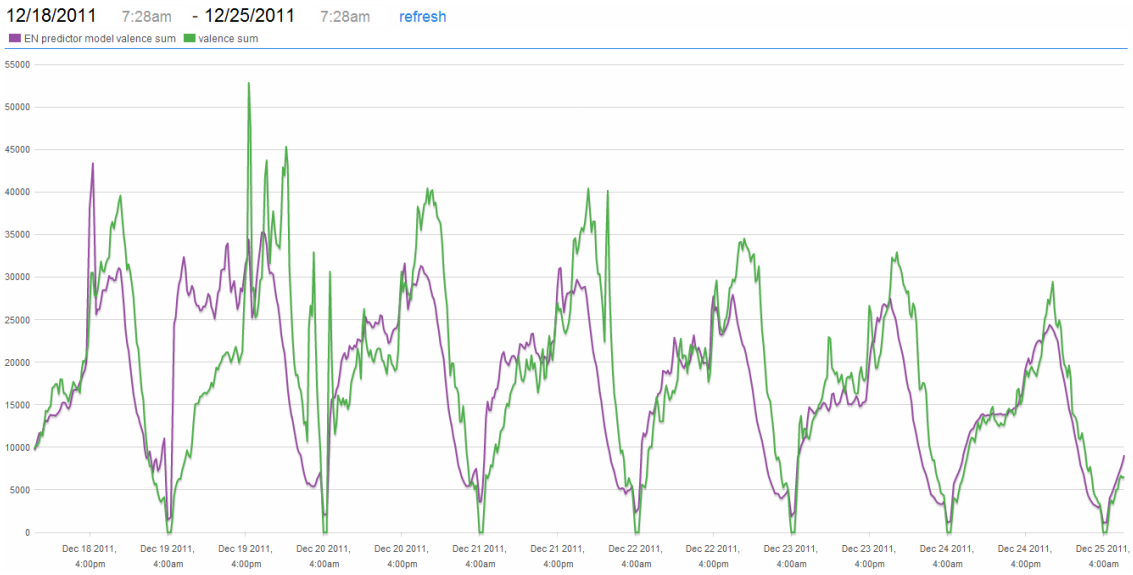

Figure 13.2: The predicted (purple) and actual (green) time series for the summed sentiment analysis method.

volume time series in that they follow a clear daily cyclical pattern. However, the sentiment aspect of these series make their patterns more jagged.

13.2.3 Average Negative Sentiment

The average negative sentiment counter calculates the average sentiment value of the negative tweets for a time frame using the following equation:

For all tweets in the frame where sentiment $<$ (Mid Sentiment $= 5$)

$$FrameValue = \frac{\sum((MaxSentiment{=}5) - TweetSentiment)}{TotalTweetCount} * 10$$

By averaging the negative sentiment, the volume of the tweets posted during the time frame is extracted out so that the only information being observed is the negative sentiment. By limiting the set of tweets to only the tweets that express negative sentiment, limiting the max sentiment to 5, some of the noise from positive tweets that can obscure the outages that are being indicated by the

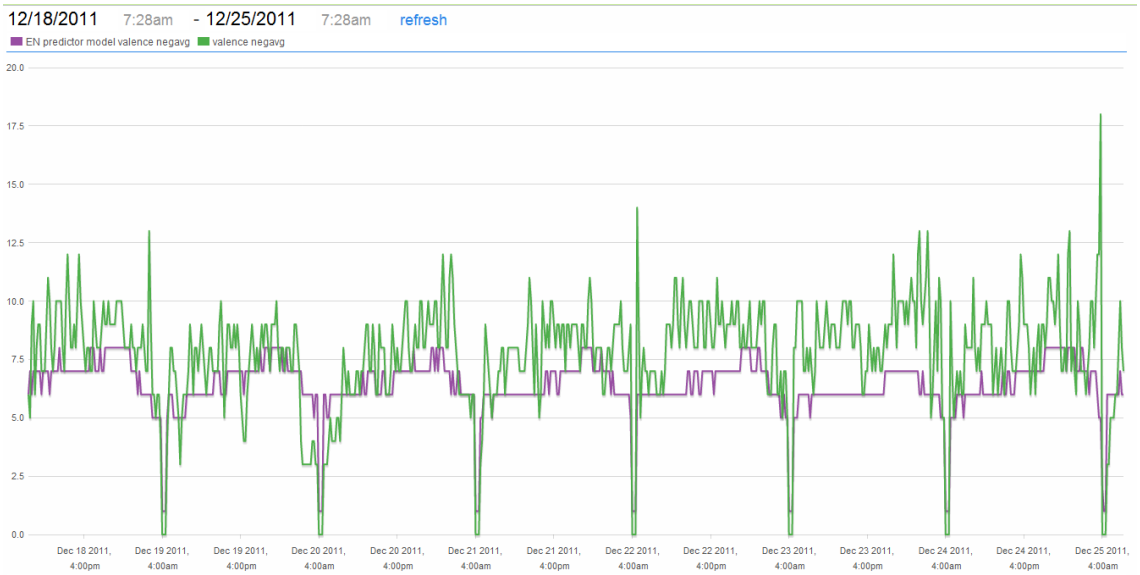

Figure 13.3: The predicted (purple) and actual (green) time series for the average negative sentiment analysis method.

negative tweets is removed. The resulting value is multiplied by 10 to increase the differentiation between normal and anomalous traffic and enhance the detail that will shown on the graphs on the user interface. This means that the values on this graph will range from 0 to 10 times the max sentiment, so 0 to 50.

Figure 13.3 shows the average negative sentiment analysis method time series for the week of December 18-25, 2011. The time series are mostly linear with a bit of variability. The lines stay around 7.5. To convert this to sentiment it needs to be divided by 10 and subtracted from the max sentiment: $sentiment = 5 - \frac{7.5}{10}$. So this graph indicates that on average the sentiments being expressed during that time frame are 4.25. This is because during normal times most of the volume will be around 5 (as shown by the average sentiment analysis method). When there is an outage or negative event then the time series values will increase.

13.2.4 Summed Negative Sentiment

The summed negative sentiment counter calculates the sum of the sentiment values for the negative tweets for a time frame using the following equation:

For all tweets in the frame where sentiment $<$ (Mid Sentiment $= 5$)

$$FrameValue = \sum ((MaxSentiment = 5) - TweetSentiment)$$

By looking at the sum of the sentiment of negative tweets, the volume of the tweets posted during the time frame affects the time series and can provide another level of information. By limiting the set of tweets to only the tweets that express negative sentiment, limiting the max sentiment to 5, some of the noise from positive tweets that can obscure the outages that are being indicated by the negative tweets is removed

Figure 13.4 shows the summed negative sentiment analysis method time series for the week of December 18-25, 2011. The time series are both similar to the total volume time series in that they follow a clear daily cyclical pattern, but the sentiment aspect of it makes the pattern more jagged. These time series look very similar to the summed sentiment analysis time series, except for they are about a tenth of the height and even more jagged. Observing tweets with negative sentiment decreases the size of the dataset that is being observed which increases the variability in the signal.

13.3 Predictors and Monitors

All of the sentiment analysis methods use both the model and trend predictors and monitors.

Figure 13.4: The predicted (purple) and actual (green) time series for the summed negative sentiment analysis method.

Chapter 14

Detection Evaluation

This detection evaluation evaluates how well the events detected by the sentiment analysis methods correspond to actual outage events reported by Netflix.

14.1 Evaluation Procedure

Since the detection evaluation is the main evaluation used to compare and rank all SPOONS analysis methods, the description of the evaluation procedure was shown as part of the Introduction in Chapter 3.

14.2 Results

Table 14.1 shows the results of the detection evaluation run on each of the sentiment analysis methods. The best results for each of the methods are highlighted.

All four of the methods were able to achieve the required precision when they

Method	Monitor	Threshold	Minutely Precision	Intersection Recall
Summed	**Trend**	**6.60**	**0.504**	**0.413**
Average Negative	**Trend**	**1.20**	**0.581**	**0.296**
Average	**Trend**	**1.05**	**0.527**	**0.255**
Average Negative	Model	0.65	0.505	0.143
Summed Negative	**Trend**	**6.95**	**0.583**	**0.102**
Average	Model	3.30	0.510	0.031
Summed	Model	-	-	-
Summed Negative	Model	-	-	-

Table 14.1: The results of the sentiment analysis method detection evaluation.

Method	Outages Alerted	Outages without Alerts
Summed	81	115
Average Negative	58	138

Table 14.2: The observed values for comparing the summed sentiment and average negative sentiment analysis methods.

used the trend monitor. The average sentiment and average negative sentiment analysis method were also able to get a precision greater than 0.5 with the model monitor, but had higher recalls with the trend monitor.

These results indicate that the summed sentiment analysis method is the most effective at detecting outages. This means the additional volume information increased the effectiveness of the method.

To determine if the best recall from summed sentiment analysis method is significantly better than the recall of the other methods, the chi square test is run on the summed sentiment analysis method and the second most effective method, the negative average sentiment analysis method. The observed recall values are shown in Table 14.2 and the chi square results are shown in Table 14.3.

The difference between the top 2 methods is significant which means that the

Method	Method	Chi Square Value	p Value
Summed	Average Negative	5.88	$0.01 < p < 0.05$

Table 14.3: The chi square results for the sentiment analysis methods.

the summed sentiment analysis method is significantly better than any of the other sentiment analysis methods.

Chapter 15

Conclusions and Future Work

15.1 Conclusions

15.1.1 Sentiment Estimation

To determine what sets of sentiment estimation options make up the best configuration for the sentiment preprocessor, every configuration was run on an experimental set of tweets. These tweets were also manually labeled by Cal Poly students as part of a survey. The results of each of the configurations were compared to the survey results using the pearson correlation coefficient.

The combination of emoticon replacement, TCR, negation, keyword emphasis, and arithmetic mean yielded the highest pearson correlation coefficient of 0.388. Unfortunately this coefficient does not indicate a high level of correlation between the manual ratings and the ratings from the sentiment preprocessor. However, since it was the most effective configuration, it is currently being used in the sentiment preprocessor.

15.1.2 Sentiment Outage Detection

There are four sentiment analysis methods that observe either all or negative sentiment and calculate either the sum or average of that sentiment for a time period. The sentiment analysis methods were evaluated using the detection evaluation procedure.

All of the methods were successful with the trend monitor and the two averaging methods were also successful with the model monitor. The summed sentiment analysis method was signficantly more effective than any of the other methods. This method was probably strongly enhanced by the effect of the volume of tweets on the time series. Since only English tweets were used in the sentiment analysis methods, the results of this method can be compared to the results of the English volume analysis method. The summed sentiment analysis method detected 81 outages while the English volume analysis method detected 67 outages. This results in a p score between 0.10 and 0.20. While this is not a statistically signficant difference, it does mean that there is an 80-90% probability that the affects of sentiment on the time series increased the effectiveness of the outage detection.

15.2 Limitations and Future Work

Sentiment Estimation. The highest pearson correlation coefficient that was achieved using the current sentiment estimation options is 0.388, which does not indicate a high level of correlation between the manual ratings and the ratings from the sentiment preprocessor. So there is definitely room for improvement for that metric.

For example, the survey that was used to manually determine the sentiment of

the set of tweets could be continued and with more ratings the average sentiments would be closer to what the sentiment preprocessor assigns the tweets which would increase the pearson correlation.

Outage Detection. Even though all of the sentiment analysis methods were successful, there's always room for improvement.

One way to possibly increase the outage detection metrics, the keyword emphasis method might be improved by only being run on a document consisting of tweets that were posted during outage times and updated every time Netflix reported official outage events in the system. This would change the emphasis from keywords of tweets about Netflix to keywords of outage posts.

This work effectively combined the sentiment estimation data with the English volume time series data to create the most effective outage detection method. So the outage detection results of that method could probably be improved upon by combining sentiment with even more successful volume analysis methods. The sentiment estimation information can be added to any volume time series that observes the sum of tweets by instead observing the sum of the estimated sentiment of those tweets.

Real Time. The methods described in this part are not currently set up to be dynamically reconfigured and have not been evaluated in a real time scenario. The detection evaluation, keyword emphasis sentiment estimation option, and tweet concurrence rating sentiment estimation option all train using the entire set of tweets in the SPOONS database which are also used to evaluate the effectiveness of the sentiment analysis methods.

The keyword emphasis and tweet concurrence rating options could be set up

to recalculate their values on a regular schedule which would allow them to change with trends over time.

As discussed in Section 9.2, there are also a number of ways to improve the detection evaluation to evaluate the real time effectiveness of the methods.

Part 4

Conclusion

15.3 Contributions

This work described several methods that can be used for detecting outages in Netflix services. It shows that there is a definite correlation between outage time periods and both the volume of tweets and sentiment of tweets during that time period. The general contributions that were presented in this thesis are:

- the implementation and evaluation of outage detection methods that monitor tweet volume over time (Chapters 5 through 8);
- several sentiment estimation procedures and an evaluation of each (Chapters 11 and 12)
- the implementation and evaluation of outage detection methods that monitor tweet sentiment over time (Chapters 13 and 14);
- and evaluations and discussions of the real time applicability of the system (Chapters 3, 8, 9, and 15).

15.4 Future Work

Even though this work presents several successful methods, there are still many ways to expand this research and increase the effectiveness of the outage detection. The largest frontiers for extending this work are:

- increasing the pearson correlation between manual and sentiment preprocessor ratings;
- determining the most effective way to tune in real time;

- combining the different types of counters to create new methods;
- creating a more detailed list of outage events that have more accurate times and include information about customer visibility and severity;
- evaluating the overlap between the events determined by each of the methods and merging all of the event lists from the different methods to increase recall.

Part 5

Additional Resources

Appendix A

Time Series Analysis

A time series is a sequence of observations that are arranged according to the time of their outcome. In this work, there are two series of data that are analyzed over time, the volumes of posts and the estimations of sentiment. From these time series, the monitors need to determine what observed values of traffic volume or estimated sentiment indicate an outage.

In order to be able to determine outliers, first they determine what the values of traffic volume or estimated sentiment are under normal conditions. In general, time series are analyzed in order to extract meaningful statistics and other characteristics of the data. This work uses time series analysis to extract models of normal volume and sentiment.

The rest of this chapter gives more background about time series analysis and how it is applied to the analysis methods implemented in SPOONS.

A.1 Additive Model

The additive model of time series analysis states that a time series can be decomposed into three components:

- **Trend**: the long term direction.
- **Seasonal**: the systematic, calendar related movements.
- **Irregular**: the unsystematic, short term fluctuations.

$$Series = TrendModel + SeasonalModel + IrregularValue$$

The irregular value of a data point is equal to the actual value subtracted from the calculated expected model value, which is the sum of the seasonal and trend components. This indicates the likely hood of a point representing a malfunction in a Netflix service.

$$IrregularValue = Series - ExpectedModel$$

$$ExpectedModel = TrendModel + SeasonalModel$$

The seasonal function is the most dominant characteristic of the volume dependent time series: the volume and summed sentiment time series. Volume data points are all offset based on the time of day. These time of day offsets were determined through exponential smoothing and averaging (explained below). The averaged sentiment series aren't affected by any cyclic offsets because the volume of tweets is averaged out to distill the sentiment ratings being observed. So their seasonal components are closer 0 making the average sentiment expected models practically equivalent to their trend models.

The trends in both the sentiment and volume time series are increasing functions. This is not directly calculated or adjusted for in either of the prediction

methods. Instead they both use methods of weighting newer values more highly so that the seasonal function is being constantly recalculated.

A.2 Exponential Smoothing

Exponential Smoothing smooths a series of values and calculates the current trend of a curve. These can then be used to predict the next values expected in the series. This section describes the Exponential Smoothing calculations and how the values of the constants were chosen for this application.

A.2.1 Single Exponential Smoothing

Single Exponential Smoothing constructs the smoothed value S by weighting the current actual value x_t and previous actual values x_{t-n} based on how long ago they occurred and a predetermined smoothing factor, α . For $t > 1$ and $0 < \alpha < 1$:

$$S_t = \alpha A_{t-1} + (1 - \alpha) S_{t-1}$$

The first smoothed value is equal the the first actual value because there are no previous values to take into account.

$$S_1 = A_1$$

The most recent previous value A_{t-1} is given a weight of α. Then the remaining weight is split between values before $t - 1$ with the same formula. When the formula is expanded, it becomes apparent that each previous term is weighted exponentially less than the one more recent than it.

$$S_t = \alpha A_{t-1} + (1 - \alpha) S_{t-1}$$

$$S_t = \alpha A_{t-1} + (1-\alpha)(\alpha A_{t-2} + (1-\alpha)S_{t-2})$$

$$S_t = \alpha A_{t-1} + \alpha(1-\alpha)A_{t-2} + (1-\alpha)^2 S_{t-2}$$

$$S_t = \alpha[A_{t-1} + (1-\alpha)A_{t-2} + (1-\alpha)^2 A_{t-3+} + (1-\alpha)^3 A_{t-4} + \ldots] + (1-\alpha)^{t-2} A_1$$

A.2.2 Double Exponential Smoothing

Double Exponential Smoothing is the same as Single Exponential Smoothing except it takes into account the trend of the previous values, b_t. For $t > 1$, $0 < \alpha < 1$, and $0 < \ < 1$:

$$S_t = \alpha A_{t-1} + (1-\alpha)(S_{t-1} + b_{t-1})$$

$$b_t = \gamma(S_t - S_{t-1}) + (1-\gamma)b_{t-1}$$

The first smoothed value is equal the the first actual value because there are no previous values to take into account.

$$S_1 = A_1$$

The simplest first trend value is just to take the difference between the first two actual values.

$$b_1 = A_2 - A_1$$

A.2.3 Exponential Smoothing Parameter Determination

This example demonstrates the effects of exponential smoothing on the total volume time series from Jan 22, 2011 12:00 AM to Jan 22, 2011 12:00 AM. The effectiveness of the smoothing is determined by observing how closely it matches the original shape while maintaining a consistent pattern.

Figure A.1 shows the effects of Single Exponential Smoothing with $\alpha =$

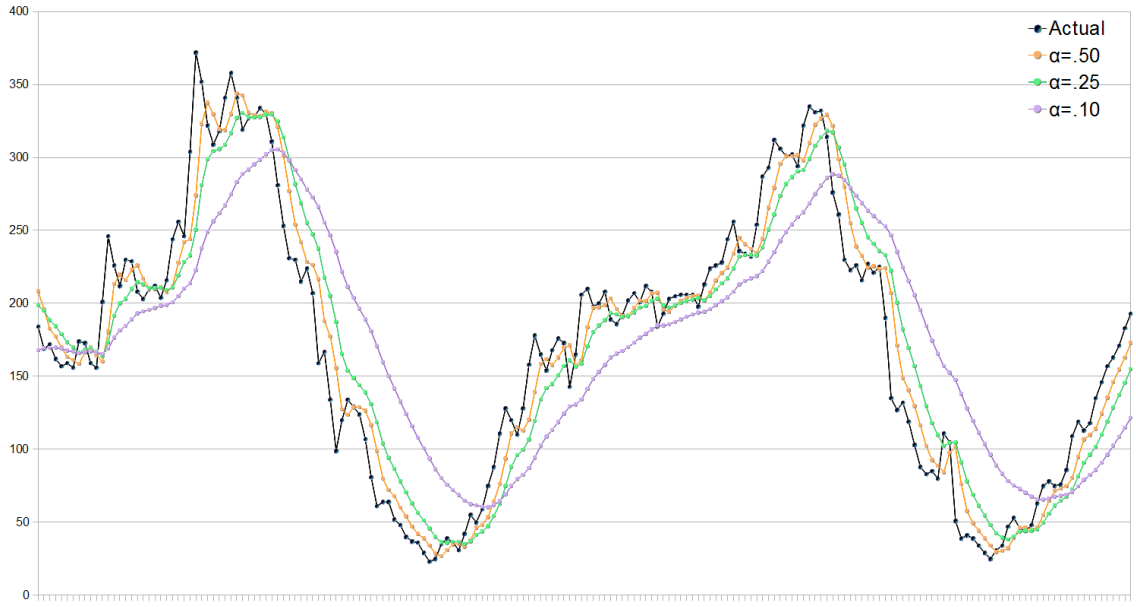

Figure A.1: The results of applying Single Exponential Smoothing over 2 days of total volume data using a range of α values.

$\{0.50, 0.25, 0.10\}$. $\alpha = 0.50$, the orange line, isn't smooth enough to give a general shape. $\alpha = 0.10$, the purple line, has a consistent shape, but is so far off from the actual values that it would be difficult to determine which values were normal and which were outliers. The best fit line for this graph is $\alpha = 0.25$, the green line.

Figure A.2 shows the effects of Double Exponential Smoothing with $\gamma = \{0.75, 0.25, 0.10\}$. Again, the higher $\gamma = 0.75$, orange line, was better able to match the peaks and $\gamma = 0.10$, the purple line, was smoother. In this case the ideal value demonstrated by $\gamma = 0.25$, the blue line.

Figure A.3 shows the comparison between the best Single (green) and Double (blue) Exponential Smoothing results. The result of the Double Exponential Smoothing is just as smooth as the Single Exponential Smoothing result, but is far better at fitting the curve.

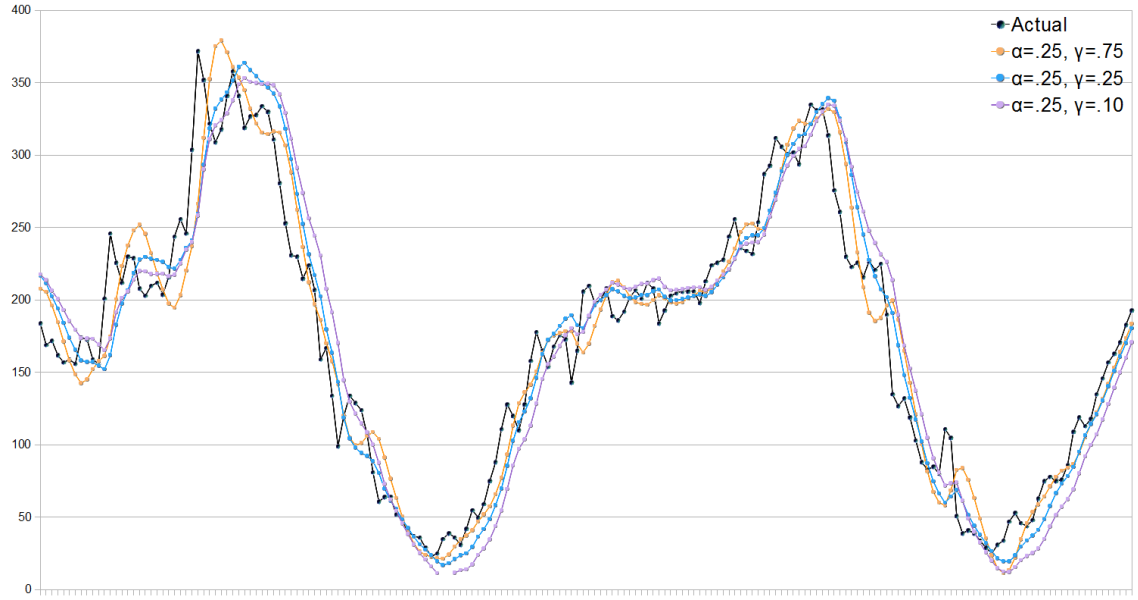

Figure A.2: The results of applying Double Exponential Smoothing over 2 days of total volume data using $\alpha = 0.25$ and a range of γ values.

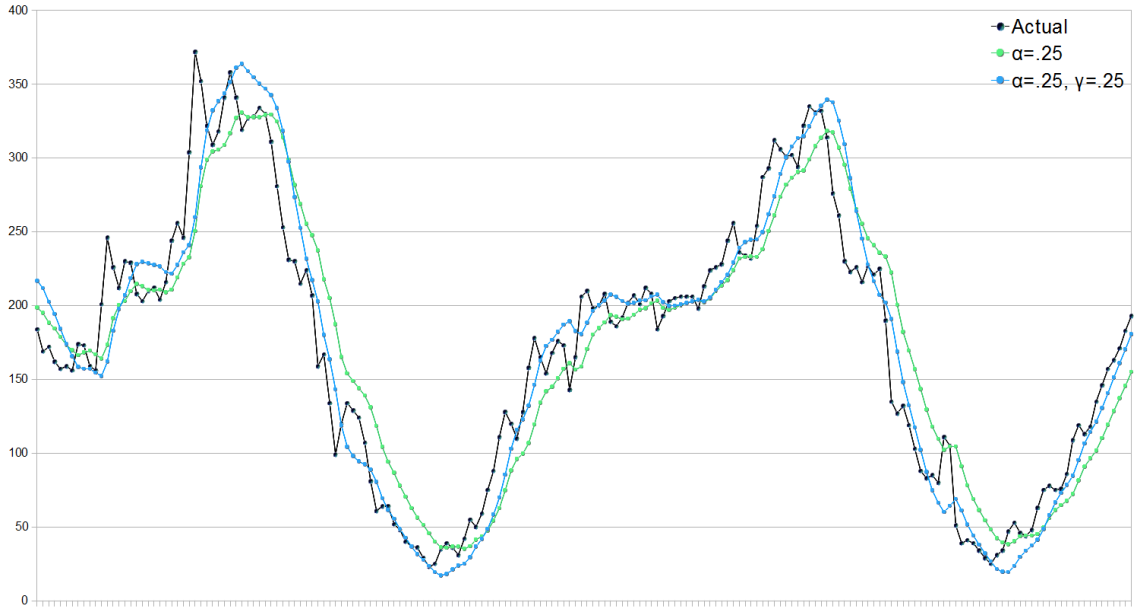

Figure A.3: A comparison of the best Single and Double Exponential Smoothing results.

Based on these results, Double Exponential Smoothing with $\alpha = 0.25$ and $\gamma = 0.25$ are used in the trend predictor's smoothing calculations.

Appendix B

Author Filtering

B.1 Introduction

User profiling is commonly used to improve the quality of Internet services with large customer bases. By creating a process that automatically analyzes customer data, a company can customize their service for each user. Two examples of well known companies that have integrated user profiling into their services are Google and Amazon. Google prioritizes search results and Amazon suggests products[20].

These companies use the information about a user's past trends to predict what the user will do in the future. This concept can be applied to the SPOONS system by using an author's previous tweets to predict their future posting trends.

B.1.1 Survey and Analysis of Related Research

Demir et al.[6] implemented a system that compares the effectiveness of features used by machine learning algorithms in the classification of tweets into cat-

egories. One difficult part of using Twitter is that there are so many posts, that users may become overwhelmed by the amount of raw data. The classification of tweets allows for filtering based on a specified set of categories. The categories that were used in this implementation were news, opinions, deals, events, and private messages. The study defined 8 features of a tweet that were used by the machine learning package WEKA[7] to determine the classification of a tweet.

Metrics and Validation. To validate the accuracy of their method, the group manually classified 5407 tweets and compared the accuracy of a Naive Bayes classifier using their 8 features to the accuracy of a Naive Bayes classifier using the traditional bag of words approach. The accuracy of each method was determined using 5-fold cross validation. Overall the 8 feature approach had an accuracy of about 95% while the standard Bag-Of-Words method had an accuracy of about 70%. However, the accuracy of the Bag-Of-Words method was increased by about 15% when it included the tweet's author as a feature of its classification.

Contributions. The contributions of the work done by Demir et al.[6] are:

- the creation a classification heuristic that improves the accuracy of tweet classifications over the standard Bag-Of-Words method;
- and the determination of authorship as crucial factor in tweet classification.

Application to SPOONS. One of Demir et al.'s conclusions is that the authorship of a tweet plays a crucial role in its classification. This conclusion strengthens the basis for the use of author profiling in the SPOONS system to filter out media reporting tweets based on the classification of the tweet's author.

B.2 Contributions of this Chapter

The contributions being claimed in this chapter are:

- a verification of Demir et al.'s[6] conclusion that the authorship of a tweet plays a crucial role in its classification;
- and an evaluation of the effectiveness of using author profiling for classification of tweets in the SPOONS data set.

B.2.1 Definition of Terms

The Twitter API returns a value for each tweet called "author". This value is the username on the account that posted the tweet. All tweets with an unknown author were excluded from the data set used in the following results. In this chapter, the term author will mean a poster who has contributed 5 or more tweets to the data set being analyzed.

A "media author" is an author that has posted at least one tweet during at least one of the reported media events. An "outage author" is an author that has posted at least one tweet during at least one of the reported outage events. If an author posts during a time when a media event overlaps with an outage event, then the author is considered both a media author and an outage author.

An author's media volume is the sum of the number of tweets that the author has posted during a media event. An author's outage volume is the sum of the number of tweets that the author has posted during an outage event. An author's event volume is the sum of their media and event volumes. If a tweet was posted during a time when a media event overlaps with an outage event, then the author's media volume and outage volume both increase by 1, therefor

their event volume is increased by 2. However, if a tweets is posted during a time when multiple media events or multiple outage events are occurring then the appropriate volume is only increased by 1, not by the number of events within the type.

B.2.2 Definition of Data Set

This evaluation was run in November of 2011 using a sets of tweets, outage events, and media events between January and November of 2011. These evaluations were run once to decide if an author profiling analysis method should be implemented during the time when the filtered volume analysis methods were being developed.

B.3 Evaluation of Distinct Author Types

As discussed in the survey section, Demir et al[6]. concluded that the authorship of a tweet plays a crucial role in its classification. However this project is using different sets of tweets and tweet type definitions. The goal of this section is to look at how strongly a tweet's author indicates the type of the tweet.

Graph B.1 shows how distinct the media and outage author sets are. The authors represented by the points on the negative side of the y-axis all post during more outage events than media events. The authors represented by the points on the positive side of the y-axis post more during media events than outage events. Authors with x = 0 have posted equally during the two event types. One thing this graph shows is that there are more media authors than outage authors. This is only because the total amount of time that media events

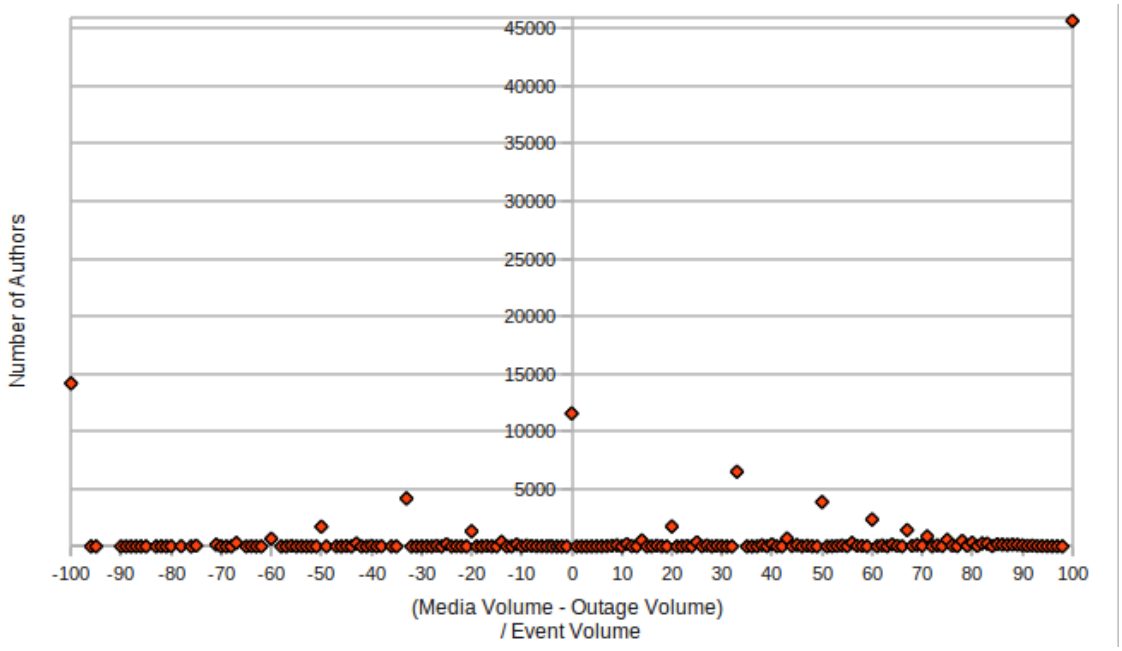

Figure B.1: An evaluation of the distinction between media and outage authors.

cover is higher than the time that outages cover; both media and outage events have an average of 2.2 authors per frame.

The points at x = -100 and x = 100 represent authors who posted only during one type of event. As the graph shows that there are a large number of authors that post exclusively for one type of event. 38% of authors posted exclusively for media events and 12% of authors posted exclusively for outage events.

To understand the significance of the rest of the values use the following equations:

$$TypePercent = \frac{TypeVolume}{EventVolume}$$

$$MediaPercent = \frac{x+100}{2}$$

$$OutagePercent = \frac{100x}{2}$$

For example, authors who are represented by points with an x value of 33

have media posts as 66.5% of their Event Volume and outage posts as the other 33.5%. Therefore authors who are represented by points with an x value of 33 or greater have posted 2 media posts for every outage post.

The results from this graph indicate that the two author sets are largely distinct.

B.4 Evaluation of Author Contributions

The next step is to evaluate whether or not authors of either type post enough during events to make analysis of their tweets effective. Reminder, authors are posters who have posted more than 5 tweets. Therefore an analysis of the tweets that authors have posted will be an analysis of a subset of the tweets in the data set.

An analysis method that filters tweets based on their authors would use tweets that an author had previously posted to determine the type of the author. Then all future tweets from media authors would be excluded and future tweets from outage authors would be emphasized. The method would need to use a predetermined threshold that would define what author total would be required before the type of the author could be determined. Increasing this threshold would increase the accuracy of the author type classification, but it also delays when the author's classification can begin to affect the traffic analysis.

Graph B.2 shows that even if all of the authors were completely distinct and an analysis method could use a threshold of 5 tweets to accurately classify all authors then only about 11% of media and 4% of outage tweets would be affected. More realistically the threshold would have to be at least 10 tweets; the

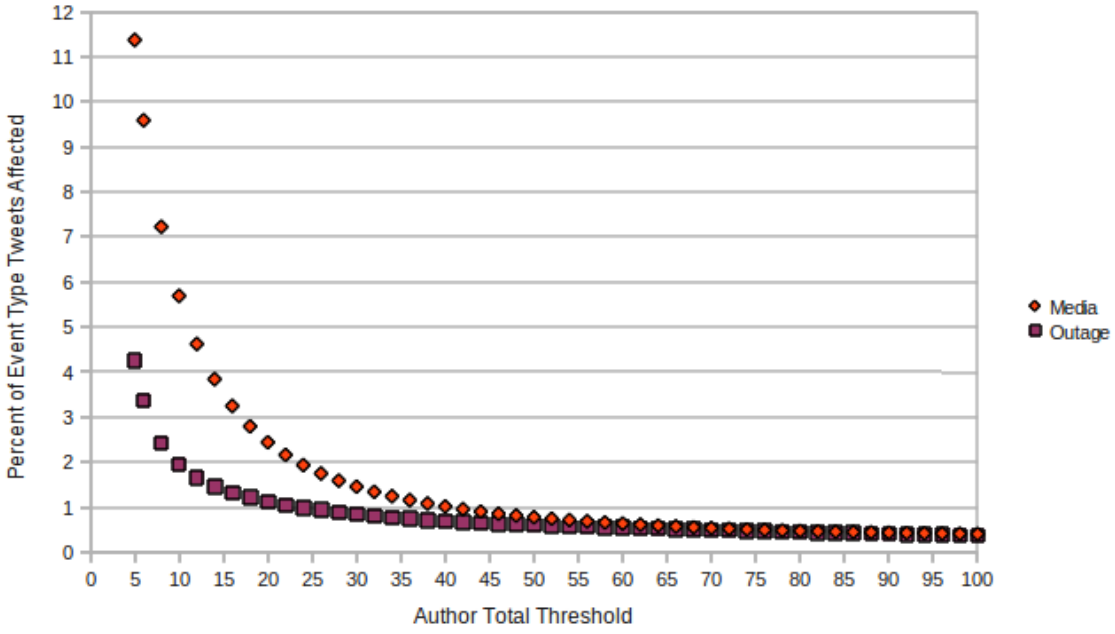

Figure B.2: A map of author total threshold value to the percent of an event type's tweets in the traffic that will be affected.

author sets aren't completely distinct so some of the type volumes included in the affected percent wouldn't actually add their event type; and classifiers aren't perfect so some authors would be missed or possibly wrongly classified. These factors would drastically drop both of the already low percentages.

The main conclusion drawn from graph B.2 is that the authors in this data set are not prolific enough to make author classification based on tweet content an effective volume analysis method.

B.5 Evaluation of Author Posting Frequencies

This section evaluates whether poster classification based just on a poster's total volume would accurately differentiate between media and outage posters. This method would use all posters, not just authors with total volumes of 5 or

more tweets. This method could be useful if for example, media posters tend to post regularly and frequently, but outage posters only post sporadically to complain about outages.

To look for an indication that one type of poster generally has a higher volume than another type an experiment was done to estimate the average total volume of posters who post during each event type. Authors' total volumes were determined using the same data as the previous section. Posters with less than 5 tweets were assigned a volume of 1 which gives results that reflect the maximum effect these posters would have on the method. Each poster's total count was weighted by the number of tweets they posted during the event type. Media posters have an average volume of about 49 tweets per author and outage posters have an average of about 56 tweets per author.

The conclusion of the results from this experiment is that there isn't a significant difference between the average volume of media and outage posters, so they can't be differentiated from their total volume exclusively.

B.6 Conclusions and Future Work

B.6.1 Conclusion

This appendix evaluated whether or not the author of a tweet can be used to classify the tweet as media or outage. The results of the experiments described are that mostly authors who post during media events are distinct from authors who post during outage events, however the authors in this data set are not prolific enough to make creating profiling for them and classifying their future tweets worthwhile.

The results from the evaluation of distinct author types (section B.3) in the SPOONS data set support Demir et al.'s conclusions is that the authorship of a tweet plays a crucial role in its classification.

B.6.2 Limitations and Future Work

A weakness of the author profiling evaluations is that tweets that were posted during a time when a media and outage event overlap are counted as both a media and outage tweet and the author of the tweet is counted as both a media and outage author. 6% of media tweets and 18% of outage tweets were posted during an overlapping time. These overlaps are a weakness in the evaluations because tweets that occurred during those times cannot be accurately classified as either outage or media. Some of the evaluations might be improved by looking for tweets that are in one category and not in the other. The time requirements for those queries are significantly higher than the ones that were used. However, that increase in time could probably be compensated for by a large one time cost of adding an index to the author column on the tweets table.

This evaluation was only run once and that was in November of 2011. It's possible that the usefulness of author profiling has increased since the time of this evaluation. Twitter is constantly increasing its tweet flow and Netflix is gaining popularity, so it's possible that the number of tweets about Netflix has increased to the point where authors are now posting enough to be profiled.

Appendix C

Full Volume and Sentiment Detection Evaluation Results

This chapter contains the complete set of volume and sentiment detection evaluation results.

Method	Monitor	Threshold	Metric Calc	True Positive	False Positive	False Negative	Precision	Recall
Volume Analysis Methods								
Linkless	Trend	3.60	Min. Inter.	0.186 80	0.168 0	0.108 116	0.525 1.000	0.632 0.408
Keyword	Trend	5.00	Min. Inter.	0.056 68	0.049 24	0.238 128	0.532 0.739	0.190 0.347
English	Trend	5.15	Min. Inter.	0.157 67	0.154 0	0.137 129	0.504 1.000	0.534 0.342
All	Trend	6.75	Min. Inter.	0.101 51	0.092 1	0.193 145	0.523 0.981	0.343 0.260
Keyword	Model	1.75	Min. Inter.	0.033 22	0.033 0	0.261 174	0.501 1.000	0.112 0.112
All	Model	- -	Min. Inter.	- -	- -	- -	- -	- -
Eng	Model	- -	Min. Inter.	- -	- -	- -	- -	- -
Linkless	Model	- -	Min. Inter.	- -	- -	- -	- -	- -
Sentiment Analysis Methods								
Summed	Trend	6.60	Min. Inter.	0.209 81	0.206 0	0.085 115	0.504 1.000	0.712 0.413
Average Negative	Trend	1.20	Min. Inter.	0.146 58	0.105 0	0.148 138	0.581 1.000	0.497 0.296
Average	Trend	1.05	Min. Inter.	0.124 50	0.111 0	0.170 146	0.527 1.000	0.420 0.255
Average Negative	Model	0.65	Min. Inter.	0.078 28	0.077 2	0.216 168	0.505 0.933	0.265 0.143
Summed Negative	Trend	6.95	Min. Inter.	0.025 20	0.018 4	0.269 176	0.583 0.833	0.084 0.102
Average	Model	3.30	Min. Inter.	0.003 6	0.002 2	0.291 190	0.510 0.750	0.009 0.031
Summed	Model	- -	Min. Inter.	- -	- -	- -	- -	- -
Summed Negative	Model	- -	Min. Inter.	- -	- -	- -	- -	- -

Appendix D

Full Sentiment Processor Results

This appendix contains the the results for each of the 256 sentiment preprocessor estimation option configurations. This is an extension of the results shown in Table 12.1.

Normalization	Word Rating	Contextual Valence Shifting	Determination	Pearson Correlation Coefficient
EMOT	TCR	NEG KE	ARIT	0.388
EMOT	TCR	NEG	HARM	0.384
EMOT	TCR	NEG	ARIT	0.380
EMOT	TCR	NEG KE	MAXA	0.378
EMOT	TCR	KE	ARIT	0.377
EMOT	TCR	NEG KE SHI	MAXA	0.375
EMOT	TCR	NEG KE SHI	ARIT	0.373
EMOT	TCR		HARM	0.371
EMOT	TCR		ARIT	0.367
EMOT	TCR	KE	MAXA	0.367
EMOT	TCR	KE SHI	MAXA	0.364
EMOT	TCR	KE SHI	ARIT	0.363
EMOT	TCR	NEG SHI	ARIT	0.360
	TCR	NEG KE	ARIT	0.360
EMOT	TCR	NEG KE	QUAD	0.359
EMOT	TCR	NEG	QUAD	0.352
EMOT	TCR	KE	QUAD	0.350
	TCR	NEG	HARM	0.350
	TCR	NEG	ARIT	0.349
	TCR	KE	ARIT	0.348
EMOT	TCR	SHI	ARIT	0.348
EMOT	TCR	NEG KE SHI	QUAD	0.347
	TCR	NEG KE SHI	ARIT	0.345
EMOT	TCR	NEG SHI	MAXA	0.344
EMOT	TCR	NEG	MAXA	0.343
EMOT TITLE	TCR	NEG KE	ARIT	0.342
TITLE	TCR	NEG KE	ARIT	0.342
EMOT	TCR		QUAD	0.341
	TCR	NEG KE	QUAD	0.338
	TCR	NEG KE	MAXA	0.338
EMOT	TCR	KE SHI	QUAD	0.338
	TCR		ARIT	0.336
	TCR		HARM	0.335
EMOT	TCR	NEG SHI	QUAD	0.334
	TCR	KE SHI	ARIT	0.334
TITLE	TCR	NEG	ARIT	0.332
EMOT TITLE	TCR	NEG	ARIT	0.332
EMOT	TCR	SHI	MAXA	0.332
EMOT	TCR		MAXA	0.331
EMOT TITLE	TCR	KE	ARIT	0.331
TITLE	TCR	KE	ARIT	0.331

Normalization	Word Rating	Contextual Valence Shifting	Determination	Pearson Correlation Coefficient
	TCR	NEG KE SHI	MAXA	0.331
EMOT TITLE	TCR	NEG KE SHI	ARIT	0.330
TITLE	TCR	NEG KE SHI	ARIT	0.330
TITLE	TCR	NEG	HARM	0.329
EMOT TITLE	TCR	NEG	HARM	0.329
	TCR	NEG SHI	ARIT	0.329
	TCR	KE	QUAD	0.329
	TCR	NEG	QUAD	0.328
EMOT TITLE	TCR	NEG KE	MAXA	0.328
TITLE	TCR	NEG KE	MAXA	0.328
	TCR	KE	MAXA	0.326
	TCR	NEG KE SHI	QUAD	0.326
TITLE	TCR	NEG KE	QUAD	0.326
EMOT TITLE	TCR	NEG KE	QUAD	0.326
EMOT	TCR	SHI	QUAD	0.324
EMOT TITLE	TCR	NEG KE SHI	MAXA	0.321
TITLE	TCR	NEG KE SHI	MAXA	0.321
	TCR	KE SHI	MAXA	0.319
TITLE	TCR	KE SHI	ARIT	0.319
EMOT TITLE	TCR	KE SHI	ARIT	0.319
TITLE	TCR		ARIT	0.318
EMOT TITLE	TCR		ARIT	0.318
	TCR		QUAD	0.317
	TCR	KE SHI	QUAD	0.317
TITLE	TCR	KE	MAXA	0.316
EMOT TITLE	TCR	KE	MAXA	0.316
TITLE	TCR	KE	QUAD	0.316
EMOT TITLE	TCR	KE	QUAD	0.316
EMOT TITLE	TCR	NEG	QUAD	0.316
TITLE	TCR	NEG	QUAD	0.316
EMOT TITLE	TCR	NEG KE SHI	QUAD	0.316
TITLE	TCR	NEG KE SHI	QUAD	0.316
	TCR	SHI	ARIT	0.316
TITLE	TCR	NEG SHI	ARIT	0.315
EMOT TITLE	TCR	NEG SHI	ARIT	0.315
EMOT TITLE	TCR		HARM	0.315
TITLE	TCR		HARM	0.315
	TCR	NEG SHI	QUAD	0.311
TITLE	TCR	KE SHI	MAXA	0.309
EMOT TITLE	TCR	KE SHI	MAXA	0.309
TITLE	TCR	KE SHI	QUAD	0.306
EMOT TITLE	TCR	KE SHI	QUAD	0.306
EMOT TITLE	TCR		QUAD	0.304

Normalization	Word Rating	Contextual Valence Shifting	Determination	Pearson Correlation Coefficient
TITLE	TCR		QUAD	0.304
	TCR	NEG	MAXA	0.303
TITLE	TCR	SHI	ARIT	0.302
EMOT TITLE	TCR	SHI	ARIT	0.302
TITLE	TCR	NEG SHI	QUAD	0.301
EMOT TITLE	TCR	NEG SHI	QUAD	0.301
	TCR	SHI	QUAD	0.300
	TCR	NEG SHI	MAXA	0.299
EMOT	TCR	NEG KE	HARM	0.298
EMOT TITLE	TCR	NEG	MAXA	0.296
TITLE	TCR	NEG	MAXA	0.296
EMOT	TCR	KE	HARM	0.295
TITLE	TCR	NEG SHI	MAXA	0.292
EMOT TITLE	TCR	NEG SHI	MAXA	0.292
	TCR		MAXA	0.290
EMOT TITLE	TCR	SHI	QUAD	0.290
TITLE	TCR	SHI	QUAD	0.290
	TCR	SHI	MAXA	0.286
TITLE	TCR		MAXA	0.283
EMOT TITLE	TCR		MAXA	0.283
	TCR	NEG KE	HARM	0.279
TITLE	TCR	SHI	MAXA	0.278
EMOT TITLE	TCR	SHI	MAXA	0.278
	TCR	KE	HARM	0.277
EMOT	TCR	NEG KE SHI	HARM	0.254
EMOT TITLE	TCR	NEG KE	HARM	0.253
TITLE	TCR	NEG KE	HARM	0.253
EMOT	TCR	NEG SHI	HARM	0.252
EMOT	TCR	KE SHI	HARM	0.251
TITLE	TCR	KE	HARM	0.251
EMOT TITLE	TCR	KE	HARM	0.251
EMOT	TCR	SHI	HARM	0.244
	TCR	NEG KE SHI	HARM	0.239
	TCR	KE SHI	HARM	0.237
EMOT TITLE	TCR	NEG KE SHI	HARM	0.227
TITLE	TCR	NEG KE SHI	HARM	0.227
EMOT TITLE	TCR	KE SHI	HARM	0.225
TITLE	TCR	KE SHI	HARM	0.225
TITLE	TCR	NEG SHI	HARM	0.223
EMOT TITLE	TCR	NEG SHI	HARM	0.223
	TCR	NEG SHI	HARM	0.218
EMOT TITLE	TCR	SHI	HARM	0.216
TITLE	TCR	SHI	HARM	0.216
	TCR	SHI	HARM	0.211

Normalization	Word Rating	Contextual Valence Shifting	Determination	Pearson Correlation Coefficient
EMOT	ANEWVGS	NEG KE SHI	HARM	0.117
EMOT	ANEWVGS	KE SHI	HARM	0.116
	ANEWVGS	NEG KE SHI	HARM	0.116
	ANEWVGS	KE SHI	HARM	0.116
EMOT	ANEWVGS	NEG KE	HARM	0.105
	ANEWVGS	NEG KE	HARM	0.105
TITLE	ANEWVGS	KE SHI	HARM	0.103
EMOT TITLE	ANEWVGS	KE SHI	HARM	0.103
TITLE	ANEWVGS	NEG KE SHI	HARM	0.103
EMOT TITLE	ANEWVGS	NEG KE SHI	HARM	0.103
EMOT	ANEWVGS	KE	HARM	0.103
	ANEWVGS	KE	HARM	0.103
EMOT TITLE	ANEWVGS	NEG SHI	HARM	0.096
TITLE	ANEWVGS	NEG SHI	HARM	0.096
EMOT TITLE	ANEWVGS	SHI	HARM	0.094
TITLE	ANEWVGS	SHI	HARM	0.094
	ANEWVGS	NEG KE	MAXA	0.089
EMOT	ANEWVGS	NEG KE	MAXA	0.089
EMOT	ANEWVGS	NEG SHI	HARM	0.088
	ANEWVGS	NEG SHI	HARM	0.088
	ANEWVGS	NEG SHI	MAXA	0.086
EMOT	ANEWVGS	NEG SHI	MAXA	0.086
EMOT	ANEWVGS	SHI	HARM	0.086
	ANEWVGS	SHI	HARM	0.086
EMOT	ANEWVGS	NEG KE SHI	MAXA	0.083
	ANEWVGS	NEG KE SHI	MAXA	0.083
	ANEWVGS	KE	MAXA	0.077
EMOT	ANEWVGS	KE	MAXA	0.077
EMOT	ANEWVGS	NEG	MAXA	0.077
	ANEWVGS	NEG	MAXA	0.077
EMOT	ANEWVGS	KE SHI	MAXA	0.077
	ANEWVGS	KE SHI	MAXA	0.077
	ANEWVGS	SHI	MAXA	0.076
EMOT	ANEWVGS	SHI	MAXA	0.076
EMOT	ANEWVGS	NEG KE SHI	ARIT	0.074
	ANEWVGS	NEG KE SHI	ARIT	0.074
TITLE	ANEWVGS	NEG KE	HARM	0.074
EMOT TITLE	ANEWVGS	NEG KE	HARM	0.074
EMOT TITLE	ANEWVGS	KE	HARM	0.072
TITLE	ANEWVGS	KE	HARM	0.072
TITLE	ANEWVGS	NEG SHI	MAXA	0.070
EMOT TITLE	ANEWVGS	NEG SHI	MAXA	0.070
EMOT	ANEWVGS	NEG KE	ARIT	0.069

Normalization	Word Rating	Contextual Valence Shifting	Determination	Pearson Correlation Coefficient
	ANEWVGS	NEG KE	ARIT	0.068
EMOT	ANEWVGS		MAXA	0.066
	ANEWVGS		MAXA	0.066
EMOT	ANEWVGS	KE SHI	ARIT	0.064
	ANEWVGS	KE SHI	ARIT	0.064
EMOT	ANEWVGS	NEG	HARM	0.064
	ANEWVGS	NEG	HARM	0.064
EMOT	ANEWVGS	NEG SHI	ARIT	0.063
	ANEWVGS	NEG SHI	ARIT	0.063
EMOT TITLE	ANEWVGS	NEG KE	MAXA	0.062
TITLE	ANEWVGS	NEG KE	MAXA	0.062
TITLE	ANEWVGS	NEG	MAXA	0.061
EMOT TITLE	ANEWVGS	NEG	MAXA	0.061
TITLE	ANEWVGS	SHI	MAXA	0.059
EMOT TITLE	ANEWVGS	SHI	MAXA	0.059
EMOT	ANEWVGS	KE	ARIT	0.058
	ANEWVGS	KE	ARIT	0.058
EMOT	ANEWVGS	NEG KE SHI	QUAD	0.056
TITLE	ANEWVGS	NEG KE SHI	ARIT	0.056
EMOT TITLE	ANEWVGS	NEG KE SHI	ARIT	0.056
EMOT TITLE	ANEWVGS	NEG KE SHI	MAXA	0.056
TITLE	ANEWVGS	NEG KE SHI	MAXA	0.056
	ANEWVGS	NEG KE SHI	QUAD	0.056
EMOT	ANEWVGS	NEG	ARIT	0.055
	ANEWVGS	NEG	ARIT	0.054
EMOT TITLE	ANEWVGS	NEG SHI	ARIT	0.053
TITLE	ANEWVGS	NEG SHI	ARIT	0.053
EMOT TITLE	ANEWVGS	NEG KE SHI	QUAD	0.052
TITLE	ANEWVGS	NEG KE SHI	QUAD	0.052
EMOT	ANEWVGS	SHI	ARIT	0.052
EMOT	ANEWVGS		HARM	0.052
	ANEWVGS	SHI	ARIT	0.052
	ANEWVGS		HARM	0.052
EMOT	ANEWVGS	NEG KE	QUAD	0.050
	ANEWVGS	NEG KE	QUAD	0.050
TITLE	ANEWVGS	NEG SHI	QUAD	0.050
EMOT TITLE	ANEWVGS	NEG SHI	QUAD	0.050
TITLE	ANEWVGS	KE SHI	MAXA	0.049
EMOT TITLE	ANEWVGS	KE SHI	MAXA	0.049
TITLE	ANEWVGS	KE	MAXA	0.049
EMOT TITLE	ANEWVGS	KE	MAXA	0.049
TITLE	ANEWVGS		MAXA	0.049
EMOT TITLE	ANEWVGS		MAXA	0.049

Normalization	Word Rating	Contextual Valence Shifting	Determination	Pearson Correlation Coefficient
EMOT	ANEWVGS	NEG SHI	QUAD	0.048
	ANEWVGS	NEG SHI	QUAD	0.048
EMOT	ANEWVGS	KE SHI	QUAD	0.047
TITLE	ANEWVGS	KE SHI	ARIT	0.047
EMOT TITLE	ANEWVGS	KE SHI	ARIT	0.047
	ANEWVGS	KE SHI	QUAD	0.047
TITLE	ANEWVGS	NEG KE	ARIT	0.046
EMOT TITLE	ANEWVGS	NEG KE	ARIT	0.046
EMOT TITLE	ANEWVGS	NEG	HARM	0.043
TITLE	ANEWVGS	NEG	HARM	0.043
EMOT TITLE	ANEWVGS	KE SHI	QUAD	0.043
TITLE	ANEWVGS	KE SHI	QUAD	0.043
EMOT TITLE	ANEWVGS	NEG KE	QUAD	0.042
TITLE	ANEWVGS	NEG KE	QUAD	0.042
EMOT	ANEWVGS		ARIT	0.042
	ANEWVGS		ARIT	0.042
EMOT TITLE	ANEWVGS	SHI	ARIT	0.042
TITLE	ANEWVGS	SHI	ARIT	0.042
EMOT	ANEWVGS	NEG	QUAD	0.040
	ANEWVGS	NEG	QUAD	0.040
EMOT	ANEWVGS	KE	QUAD	0.040
EMOT TITLE	ANEWVGS	SHI	QUAD	0.040
TITLE	ANEWVGS	SHI	QUAD	0.040
	ANEWVGS	KE	QUAD	0.040
TITLE	ANEWVGS	NEG	ARIT	0.039
EMOT TITLE	ANEWVGS	NEG	ARIT	0.039
EMOT	ANEWVGS	SHI	QUAD	0.038
	ANEWVGS	SHI	QUAD	0.038
TITLE	ANEWVGS	NEG	QUAD	0.038
EMOT TITLE	ANEWVGS	NEG	QUAD	0.038
EMOT TITLE	ANEWVGS	KE	ARIT	0.036
TITLE	ANEWVGS	KE	ARIT	0.036
TITLE	ANEWVGS	KE	QUAD	0.032
EMOT TITLE	ANEWVGS	KE	QUAD	0.032
TITLE	ANEWVGS		HARM	0.031
EMOT TITLE	ANEWVGS		HARM	0.031
EMOT	ANEWVGS		QUAD	0.029
	ANEWVGS		QUAD	0.029
EMOT TITLE	ANEWVGS		ARIT	0.027
TITLE	ANEWVGS		ARIT	0.027
TITLE	ANEWVGS		QUAD	0.026
EMOT TITLE	ANEWVGS		QUAD	0.026

Appendix E

Stop Words

able	after	any	aside	beginning
about	afterwards	anybody	ask	beginnings
above	again	anyhow	asking	begins
abst	all	anymore	auth	behind
accordance	almost	anyone	available	being
according	alone	anything	away	believe
accordingly	along	anyway	back	below
across	already	anyways	became	beside
act	also	anywhere	because	besides
actually	although	apparently	become	between
added	always	approximately	becomes	beyond
adj	among	are	becoming	biol
adopted	amongst	aren	been	both
affected	and	arent	before	brief
affecting	announce	arise	beforehand	briefly
affects	another	around	begin	but

came	due	first	gotten	home
can	during	five	had	how
cannot	each	fix	happens	howbeit
cause	edu	followed	hardly	however
causes	effect	following	has	hundred
certain	eight	follows	hasn	immediate
certainly	eighty	for	have	immediately
com	either	former	haven	importance
come	else	formerly	having	important
comes	elsewhere	forth	hed	inc
contain	end	found	hence	indeed
containing	ending	four	her	index
contains	enough	from	here	information
could	especially	further	hereafter	instead
couldnt	etc	furthermore	hereby	into
date	even	gave	herein	invention
did	ever	get	heres	inward
didn	every	gets	hereupon	isn
different	everybody	getting	hers	itd
does	everyone	give	herself	its
doesn	everything	given	hes	itself
doing	everywhere	gives	hid	just
don	except	giving	him	keep
done	far	goes	himself	keeps
down	few	gone	his	kept
downwards	fifth	got	hither	keys

know
known
knows
largely
last
lately
later
latter
latterly
least
less
lest
let
lets
like
liked
likely
line
little
look
looking
looks
ltd
made
mainly
make
makes
many
may
maybe
mean
means
meantime
meanwhile
merely
might
million
miss
more
moreover
most
mostly
mrs
much
mug
must
myself
name
namely
nay
near
nearly
necessarily
necessary
need
needs
neither
netflix
never
nevertheless
new
next
nine
ninety
nobody
non
none
nonetheless
noone
nor
normally
nos
not
noted
nothing
now
nowhere
obtain
obtained
obviously
off
often
okay
old
omitted
once
one
ones
only
onto
ord
other
others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
owing
own
page
pages
part
particular
particularly
past
per
perhaps
placed
please
plus
poorly
possible
possibly
potentially
predominantly
present
previously
primarily
probably
promptly
provides
put
que
quickly
quite

ran	sec	significantly	strongly	thereby
rather	section	similar	sub	thered
readily	see	similarly	substantially	therefore
really	seeing	since	successfully	therein
recent	seem	six	such	thereof
recently	seemed	slightly	sufficiently	therere
ref	seeming	some	suggest	theres
refs	seems	somebody	sup	thereto
regarding	seen	somehow	sure	thereupon
regardless	self	someone	take	these
regards	selves	somethan	taken	they
related	sent	something	taking	theyd
relatively	seven	sometime	tell	theyre
research	several	sometimes	tends	think
respectively	shall	somewhat	than	this
resulted	she	somewhere	that	those
resulting	shed	soon	thats	thou
results	shes	sorry	the	though
right	should	specifically	their	thoughh
run	shouldn	specified	theirs	thousand
said	show	specify	them	throug
same	showed	specifying	themselves	through
saw	shown	state	then	throughout
say	showns	states	thence	thru
saying	shows	still	there	thus
says	significant	stop	thereafter	til

tip	unto	way	wherever	within
together	upon	wed	whether	without
too	ups	welcome	which	words
took	use	went	while	world
toward	used	were	whim	would
towards	uses	weren	whither	wouldn
tried	using	what	who	www
tries	usually	whatever	whod	yes
truly	value	whats	whoever	yet
try	various	when	whole	you
trying	very	whence	whom	youd
twice	via	whenever	whomever	your
two	viz	where	whos	youre
under	vol	whereafter	whose	yours
unfortunately	vols	whereas	why	yourself
unless	want	whereby	widely	yourselves
unlike	wants	wherein	willing	zero
unlikely	was	wheres	wish	
until	wasn	whereupon	with	

Glossary

Alert An email sent to Netflix engineers to alert them of an outage detected by the SPOONS system. See Section 2.3.1 for more information.

Analysis Method A control element that runs a preprocessor, a counter, predictors, and monitors and produces a set of detected outage events. See Section 2.2 for more information.

Counters The component of an analysis method that aggregates data about a time period. See Section 2.2.2 for more information.

Detection Evaluation An evaluation procedure that is run on all Analysis Methods and determines how well the events detected by the method correspond to the actual outage events reported by Netflix using metrics: precision; recall; and F0.5 score. See Chapter 3 for more information.

Filtered Volume Analysis Methods Analysis methods that filter tweets, create a time series of the filtered volume over time, and monitor it for indications of outages. See Chapter 5 for more information.

Media Event A time period when Netflix related news is released and highly discussed, e.g. information about quarterly earnings reports, new prod-

ucts/services announcements or profiles of key Netflix personnel in the media.

Monitors The components of an analysis method that compare the actual time series created by the counter to the expected time series generated by the predictors and look for indications of an outage event. See Section 2.2.4 for more information.

Netflix Inc. [NASDAQ: NFLX] is the world's leading Internet subscription service for enjoying movies and TV series with more than 23 million streaming members in the United States, Canada, Latin America, the United Kingdom and Ireland[14].

Outage Event A time period when there is a problem with a service provided by Netflix.

Predictors The components of an analysis method that algorithmically map normal expected traffic patterns. See Section 2.2.3 for more information.

Preprocessors The component of the analysis methods that extract information from raw tweets that is then used by a counter. See Section 2.2.1 for more information.

Real Time Some of Netflix's services stream to customers in real time which means the users expect to get immediate responses from those services. So when they go down, the customers want the problem to be fixed immediately. These analysis methods need to have real time responses that are as close to immediate detection as possible. This means that the system needs to use whatever information it has available to it up to right before the outage to detect the event and alert Netflix engineers.

Sentiment The sentiment of a tweet is a combination of: the emotion it expresses; an attitude about something; and the mood of the author while writing it.

Sentiment Analysis Methods Analysis methods that use estimated sentiments to create a time series of sentiment over time, and then monitor it for indications of outages. See Chapter 13 for more information.

Sentiment Estimation The use of natural language parsing to determine the sentiment of a tweet. Also known as "opinion mining". See Chapter 11 for more information.

Sentiment Estimation Options The use of natural language parsing to determine the sentiment of a tweet. Also known as opinion mining". See Part 11 for more information.

Sentiment Preprocessor Configurations A set of sentiment estimation options that the preprocessor uses to estimate the sentiment of a tweet. See Part 11 for more information.

SPOONS Swift Perception Of Online Negative Situations. This is the name of the system that this work is implemented in. See Part 1 for more information.

Time Series Analysis The analysis of a series of data points over time. In this work those data points are the volume or estimated sentiment of a subset of the traffic about Netflix on Twitter during a time period. See Appendix A for more information about time series analysis and how it's applied to this work.

Tweet A post to a Twitter service. See Section 1 for more information.

Twitter Twitter is an online social networking service that only allows its users to post 140 characters of text. See Section 1 for more information.

User Profiling The practice of gathering information about users of a service and creating profiles that describe them. See section B.1 for more information.

Web UI A web user interface that provide Netflix engineers with information from the SPOONS system. See Section 2.3.2 for more information.

Bibliography

[1] Chi square test. http://www2.lv.psu.edu/jxm57/irp/chisquar.html.

[2] ACM/IEEE-CS Joint Task Force. Software engineering code of ethics and professional practice, 1999. http://www.acm.org/about/se-code.

[3] E. Augustine, C. Cushing, A. Dekhtyar, M. Tognetti, and K. Paterson. Outage detection via real-time social stream analysis: Leveraging the power of online complaints. In *WWW 2012: Proceedings of the 21st World Wide Web Conference*. ACM, 2012.

[4] M. M. Bradley and P. J. Langs. Affective norms for English words (ANEW): Stimuli, instruction manual, and affective ratings. Technical report, Center for Research in Psychophysiology, University of Florida, Gainesville, Florida, 1999.

[5] A. Culotta. Detecting influenza outbreaks by analyzing twitter messages. In *KDD Workshop on Social Media Analytics*, 2010.

[6] E. Demir, M. Demirbas, H. Ferhatosmanoglu, D. Fuhry, and B. Sriram. Short text classification in twitter to improve information filtering. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Re-*

search and development in information retrieval, 2010. http://www.cse.ohio-state.edu/ hakan/publications/TweetClassification.pdf.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[8] F. Jabr. Using twitter to follow trends beats the stock market. *NewScientist*, (2829), Sept. 2011. http://www.newscientist.com/article/mg21128295.900-using-twitter-to-follow-trends-beats-the-stock-market.html.

[9] K. Levchenko, B. Meeder, M. Motoyama, S. Savage, and G. M. Voelker. Measuring online service availability using twitter. In *Proc. of the 3rd Workshop on Online Social Networks (WOSN 2010)*, 2010.

[10] B. Liu. *Web Data Mining*. Springer, 2007.

[11] Y. Matsu, M. Okazak, and T. Sakak. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW 2010: Proceedings of the 19th World Wide Web Conference*, 2010. http://ymatsuo.com/papers/www2010.pdf.

[12] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. In *Proceedings of the 16th International FLAIRS Conference*, pages 293–296, 2003.

[13] K. McEntee. personal communication, 2011.

[14] Netflix Inc. Netflix releases fourth-quarter 2011 financial results. *Netflix Press Release*, 2011. http://netflix.mediaroom.com/index.php?s=43&item=438.

[15] NIST/SEMATECH. 6.4.3.1. single exponential smoothing. e-Handbook of Statistical Methods. http://www.itl.nist.gov/div898/handbook/.

[16] NIST/SEMATECH. 6.4.3.2. double exponential smoothing. e-Handbook of Statistical Methods. http://www.itl.nist.gov/div898/handbook/.

[17] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 122–129. AAAI Press, 2010.

[18] M. F. Porter. An algorithm for suffix stripping. In K. Sparck Jones and P. Willett, editors, *Readings in information retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. http://tartarus.org/martin/PorterStemmer.

[19] The Stanford Natural Language Processing Group. *The Stanford Parser*. http://nlp.stanford.edu/software/lex-parser.shtml.

[20] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 675–684, New York, NY, USA, 2004. ACM.

[21] Twitter. #numbers, Mar. 2011. http://blog.twitter.com/2011/03/numbers.html.

[22] Twitter. Terms of service, June 2011. https://twitter.com/tos.

[23] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu. Combining lexicon-based and learning-based methods for twitter sentiment analysis. Technical report, HP, Jun 2011. http://www.hpl.hp.com/techreports/2011/HPL-2011-89.html.