Automating Open Domain Event Summaries by Harnessing Collective Reactions on Twitter

Kain Nanne*
VU University Amsterdam
k.nanne@student.vu.nl

ABSTRACT

Microblogging sites have become popular platforms for online news reporting as well as socially participating in and interacting with the discussion of real-time events. This paper researches an automated solution to the inability of a human to wholly consume and comprehend the vast amount of data surrounding topics online. We introduce the Collective Reactions for Event Summarization (CRES) approach, which uses an original combination of proven algorithms to harness signals in online activity, social interactions, content metadata, and language overlap to build comprehensive summaries of events through collective reactions from the crowd. The methodology is open sourced as an end-to-end framework exemplified using twelve open domain events. Our experiments consider the two questions of: create a standard feature set for consistently classifying newsworthiness in open domain microblog documents, and provide a summary which improves upon the defined baseline when evaluated using CrowdTruth. Results show promising results towards consistent classification on open domain documents, and significant improvements to our baseline for automating event summarization on Twitter.

Keywords

twitter, automatic summarization, event detection, text mining, document classification

*MSc. Business Information Systems

1. INTRODUCTION

Microblogging is the activity of sharing a small amount of information over the web. These small documents of information can include combinations of text and media content, and are typically shared over the public domain. Microblogging has become increasingly popular for social as well as news reporting. A 2015 survey by the Pew Research Center found that over 60% of social media users of Facebook and Twitter actively source news from the sites, an increase in over 10% for both sites from 2013. [3] Particularly in news, given the short time it takes to write and ability to share with a mass audience instantaneously, it is attractive for organizations to share smaller pieces of information as real-time events unfold. As a result of this, an immense amount of information surrounding an event is fragmented across sites and accounts making it impossible for a human to wholly comprehend. As this information is disconnected and drowning in extraneous data, it is a difficult feat to gather and view related content let alone read it in a meaningful way.

The setting of this research paper is described by the following **problem statement**:

> *It has become impossible for a human to manually search, collect, filter and comprehend the vast amount of fragmented pieces of information surrounding an topic online as information is published too often, inherently duplicated and drowning in extraneous data.*

Twitter is the current largest microblogging site, with hundreds of millions of active users per month. In 2013, Twitter recorded a volume of roughly 500 million tweets a day, or 5,700 tweets per second. [10] With all the public content available on the site combined with the real-time nature of it being published creates an attractive platform to search for up to date information about current events. As well, Twitter recommends that its users apply hashtags to content, written with the # symbol, to tag content by topics and keywords. This, combined with the a robust and public API, provide an easily accessible database of indexed content for developers and researchers to query content.

The goal of this research paper is described by the following **motivational statement**:

> *From an event-focused data query on Twitter, extract k newsworthy documents that best represent the important developments of an event, and visualize these summary documents in an interactive way which aids in the learning about said*

event.

Our research project aims to develop an end-to-end system's framework for querying event data on Twitter, and providing a meaningful summary in a presentable form easily consumable by an end user. The system should harness trends in the event activity timeline, signals from the collective crowd, and employ machine learning to identify moments of importance as well as meaningful information for summarizing. Since applications and use cases for such an event summary can vary across end users or domains, our final system output will be a dynamic selection of k documents variable on demand.

This paper and its contained experiments address the following **research questions**:

> ***Q1:*** *For filtering newsworthy content by a supervised classifier, can we build a standardized feature set to yield consistently successful results when created on, and tested across, open domains?*
> ***Q2:*** *Given a set of documents related to an open domain event on Twitter, can we create an automated extraction of k newsworthy documents which, when evaluated on crowd-defined summary accuracy, improves upon the baseline interactions approach utilized by Twitter?*

The original contribution of this paper is two fold. First we conduct unique experiments establishing a standard feature set and model for classifying newsworthiness of open domain documents on Titter, and for comparing the effectiveness of our approach to the defined baseline. Second we provide an open sourced end-to-end framework successfully integrating the approach on twelve open domain events. An interactive dashboard is provided to showcase the event summaries using state-of-the-art visualization techniques inspired by Twitter engineering.

This paper is organized in the following way. After discussing previous and related literature, we document the data used in this research as well as how it was queried and processed. In section 4, an original approach is proposed to complete the task of automatic event summarization. During this section we describe the three components, and how our system utilizes them for summarization. The components are Event Detection, Classification, and Phrase Reinforcment located in sections 4.1, 4.2, 4.3 respectively. The summarization system and final dashboard are provided in section 5 with a description of the integrated, end-to-end framework. Experiments on classification validation and final summary evaluation are provide in section 6, followed by experimental results in 7. Finally a conclusion on the research and discussion of further opportunities is eventually given.

2. RELATED WORK

This research is closely related topic detection and tracking (TDT) originally studied by Allan et al. in 1998 for updating a news story as it unfolds over time. [1] Other research in automatic document summarization, tweet timeline generation (TTG), and progressive or temporal update summarization for microblog streams is closely related. Further variations include retrospective or online event detection, and first story detection (FSD). Terminology varies across research as there are many interpretations to the problem definition and application for automatically summarizing microblog data. Commonly though, has Twitter been used as a data source due to it's public popularity and accessibility.

Due to volume of related research, we narrow our scope and define our research to be most closely related to, and an attempt to improve on, the research focused on summarization of specific structured events over a defined timeline. Zhao et al. used event detection techniques to identify moments of importance in online NFL games with high accuracy. [19] Zubiaga et al. used similar event detection and document weighting to construct tweet moment summaries of structured events. [20] Sakaki et al. trained a classifier to detect natural disaster events in Japan, and used noise filtering to predict centers of earthquakes and probabilistic trajectories of typhoons. [15]

Most similar is that by Nichols et al. for summarizing sporting events on Twitter. [13] They combined a simple event detection technique with an language overlap algorithm to extract sentences summarizing important event moments. The most significant research improvements we contribute are the application to a more diverse dataset and inclusion of multiple event domains. We also employ a supervised classification algorithm instead of a greedy approach to spam filtering, and harness the crowd on a larger scale for evaluating our summary in a unique way.

Novel to related research, we develop an interactive dashboard for visualizing any open domain events. Our final visualization is an attempt to improve upon the state of the art visualization methods currently employed by Twitter, by providing a textual summary in addition to the interactive event timeline. Most recently, an interactive visualization was developed at Twitter to highlight all the goals scored during the 2016 Euro Cup. [1] A spike detection algorithm was used to detect underlying important trends in the timeline distributions and then manually added to annotate the spikes. This research works to study the process of automating such a project, apply it open domain events, and create a simple underlying open source framework.

3. DATA

Previous research by Nichols et al. proved a similar summarization approach, including event detection and phrase reinforcement, can be successful for summarizing specific sporting events. [13] We extend their research by applying our approach to a more diverse set of sporting events, and introduce a second domain. Therefore the events for this research are classified in two domains of Sporting Events and Technology Conferences. All events occurred from May to June of 2016 and vary in length data volume. The diversity on events within a particular domain was intentionally chosen to broaden the application, and research value of the summarization system. The two domains were chosen due to their typical audiences having a higher activity on Twitter.

We focus on events that are internationally anticipated and have explicitly defined event event hashtags. This makes the event data easy to detect and isolate from the rest of the noise on Twitter. Other events such as terrorism or natural disasters, which are not anticipated are difficult to search for. These unexpected events are typically followed by posts without predefined hashtags, or use various closely

[1] https://interactive.twitter.com/euro2016/

Domain	Event Title	Hashtag	Start Date	End Date	Volume	Max t/m	Total Eng.
Sport	UEFA Champions League Final	#uclfinal	2016-05-28	2016-05-28	192602	4387	1157004
Sport	French Open Finals	#rg16	2016-06-04	2016-06-05	23552	371	206522
Sport	Monaco Grand Prix	#monacogp	2016-05-28	2016-05-29	62212	855	737306
Sport	Stanley Cup Playoffs Final Game 7	#stanleycup	2016-06-12	2016-06-13	52528	2274	618064
Sport	24 Hours of Le Mans	#lemans24	2016-06-18	2016-06-19	31672	386	246008
Sport	NBA Playoffs Finals Game 7	#nbafinals	2016-06-01	2016-06-20	514439	11899	2913478
Tech	The Next Web Conference Europe	#tnweurope	2016-05-25	2016-05-27	7636	20	21634
Tech	Recode Code Conference	#codecon	2016-05-31	2016-06-02	5780	34	51438
Tech	Google I/O	#io16	2016-05-18	2016-05-18	25533	270	118518
Tech	Apple Worldwide Developer Conference	#wwdc2016	2016-06-13	2016-06-13	40027	449	116413
Tech	Lenovo Tech World	#lenovotechworld	2016-06-08	2016-06-10	13185	51	34050
Tech	Xbox E3	#xboxe3	2016-06-13	2016-06-13	86475	1246	339161

Table 1: Data Details

related hashtags making the event data difficult to isolate. Techniques of query expansion through clustering or topic modeling have been studied to provide additional ways of detecting and isolating new query terms from these types of events, although we chose to rely on a single query term for each event to allow for an equal evaluation of the framework across all events. In addition, this research focuses on retrospectively searching for previously concluded events. Alternative to event detection and summarization for online streams, we chose to work with historically searched data to allow for to eleminate potential interuptions in streams, accomodate for subsequently deleted data, and to allow for delayed engagements to accumulate. It is argued that the framework proposed and algorithms used can be directly applied to online event streams with no limitations.

3.1 Query

Data for this research was accessed through Twitter's public REST API. [2] Using the GET tweets endpoint, a user with authorization is able to access historical data up to seven days old. This option provides access to significant engagements associated with tweets, which due to the delay of social reactions are not entirely, instantly associated with streaming data. Tweet objects accessed from the API were written directly to a file in original JSON format, to ensure no data was lost and the process of searching could happen most efficiently.

A program was written in Python to automate the recursive searching for data. Each call returned 100, English JSON tweet objects. The minimum tweet id from the latest call was used as the maximum tweet id for the subsequent call, allowing for automation. Twitter API rate limits and connection frequencies were respected and handled locally to avoid account punishment. A sample of this searching program is provided on the project GitHub page, referenced in the Appendix.

Twitter was queried using a predefined hashtag, recognized on Twitter by the symbol, for collecting data on each event. Each event hashtag used for this research was selected by an acknowledgement from the official Twitter account of that event's governing body, either in a their account bio or in their first tweet about the event. For example the official hashtag for Google I/O was identified by @Google to be #io16, and the hashtag for Monaco Grand Prix was identified by account @F1 to be #monacogp. More recently Twitter itself has been acknowledging official event hashtags by adding flairs or emojis next to certain hashtags when used during an event. A recent post on the Twitter blog preempts the Copa America sporting event by detailing the official hashtags and ways to partake in official conversation.[14] This research makes the assumption that all data with the official event hashtag is relevant to the event. Qualitative results reinforce this conclusion, identifying tweets that are irrelevant only to be spam, and not irrelevant news.

[2]https://dev.twitter.com/rest/public

3.2 Process

After searching we transform the JSON data into a flat database for easy ingestion in Python, which was the language used for the entirety of this research. The Tweet JSON objects were converted to Pandas DataFrames to allow for efficient manipulation and analysis. Nested object dictionaries of 'entities' and 'user' were manually extracted and added as DataFrame columns, then dropped to save space. All events were stored using an equal date range relative to the event start and finish, as stated on the official event website in days. Date range was defined as minus one day to plus one day from the event dates in UTC time. By standardizing in this way we can comfortably handle global events, and compare events on equally representative data distributions.

4. APPROACH

This research proposes using **Collective Reactions for Event Summarization** (CRES) as an approach to automatically summarize data surrounding events on Twitter.

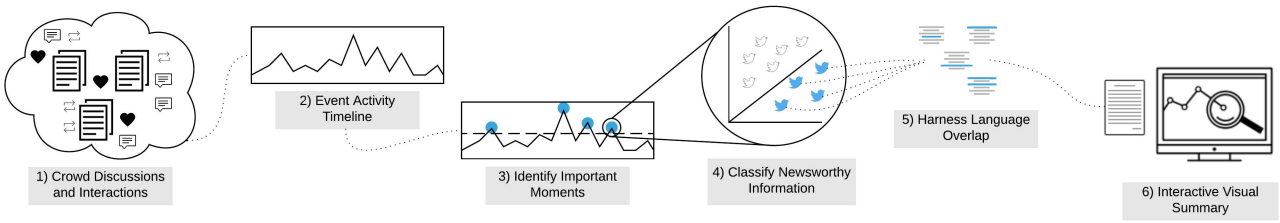

Figure 1: CRES Flowchart

The approach harnesses signals in online activity, social interactions, content metadata, and language overlap to build comprehensive summaries of events through collective reactions from the crowd. The methodology given is an original combination of previously proven components for similar summarization tasks on Twitter. We test our approach on retrospective events within two domains, but argue there are no limitations to extending the entire framework for any open domain event streams. This section briefly describes the process involved in each step, before detailing them individually later in the paper.

A visual description of how the approach is realized is diagrammed in Figure 1. As an event takes place, organizations and members of the public start discussing relevant information, and socially interact with each other's content while doing so. As this data is isolated from the noise onine by the use of a specific hashtag, it can be aggregated into a timeline. Using trend analysis, we are able to identify specific moments where discussion or interactions significantly increased representing important moments during the event. Once these important moments are identified, a statistical model can be trained to predict specifically which posts are of interest for summarizing an event. In the case of this research, tweets of interest to our system, are predicted by a human-trained machine to be "newsworthy". Using only these documents at each moment of importance, an additional algorithm is employed to identify and harness language overlap between posts to weight tweets on their importance for summarizing that specific moment. This weighting by language overlap, referred to as phrase reinforcement, is completed for each of k important moments, and the summary is built by selecting the top document from each moment and displaying them in chronological order.

The foundation of the approach lies in the open source algorithms referred to as Event Detection and Phrase Reinforcement. These summarization components, as well as an original classification component for document filtering, are detailed fully in their own sections 4.1, 4.3, and 4.2 below.

4.1 Event Detection

Our approach considers a dynamic ranking to summary creation, which is a method commonly used in automatic summarization research. This is necessary to address the basic problem of a user not being able to consume all data available. To allow for different users of certain use cases to only read data which they may have time for, our final summary length is a dynamic parameter of k documents, able to be changed by the user at any moment depending on application requirements. To do this we introduce an event detection algorithm to identify and rank moments throughout the event timeline.

The underlying assumption of event detection is that as something happens, individuals will tweet about it in real-time causing bursts in Twitter activity. In 2011 Twitter blogged about how their site's activity was a reflection of global pulse, citing when volume spiked over 5,000 tweets per second multiple times during an earthquake.[6] Within our event timeline, we can identify specific times of importance for summarization in this way. We refer to these moments as subevents. Using theory related to anomaly detection in signal processing for time series data, we can identify patterns that may significantly deviate from an underlying distribution in the dataset. By computing an expected frequency of occurrences per a historic time window we can measure differences in behavior above or below the historic distribution for a current interval of time, which is an indication of relative event activity. Furthermore, by calculating expected historical frequency using a moving average, this approach can easily be applied to online data. Related Twitter-Specific Frameworks that implement similar theory include Twitter's Breakout Detection[8], and Anomaly Detection.[9] Most applicable to our event summarization research is the anomaly detection. It uses a statistical model to score datapoint on the probability of it being included in the underlying distribution. [4] We simplify the theory here, instead of fitting a model and predicting anomalies or breakouts, to only rank the deviations of data points above or below our expected frequency and use a dynamic threshold for selecting the most significant points. This allows for the extension of this research's framework to a diverse set of event summary applications, and can be applied only with a few datapoints initially available while streaming.

The measure of distribution is defined as volume of tweets per minute, and an anomaly in this frequency would indicate a subevent. An interval of 1 minute is used to bin tweets as this was qualitatively studied as a good balance between the amount of relevant data to irrelevant noise. As this interval increases, we see an increase in the tweets per bin available, which can provide more data for summarization beneficial to the PR algorithm, although more noise as well which is irrelevant to the desired subevent and therefore can skew the final PR weights.

Subevent significance scores are measured by the deviation from each minute, to an expected frequency. Expected frequency is calculated with a window of 2 minutes using:

$$expected = \frac{1}{n}\sum_{i=0}^{n-1} y - i$$

Moving average is also referred to as the rolling mean, and

is visualized in Figure 2. The deviation is then measured as $tweetsperminute - expected$ and can be seen visualized in Figure 3. The threshold is calculated by taking the deviation at rank k. For this research we use a k of 25, giving us enough data to evaluate the performance of this method across events and domains. Subevents are included in the final summary if $y^i expected > threshold$ The top subevents are appended to the original timeline in Figure 4. The final visualization includes a feature to control the threshold parameter and dynamically display more or less tweets for summarization.

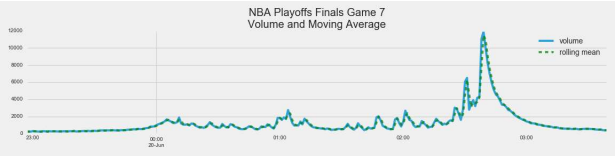

Figure 2: Volume and Moving Average

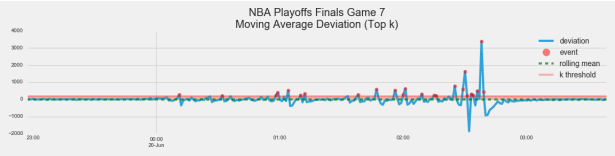

Figure 3: Deviation from Moving Average

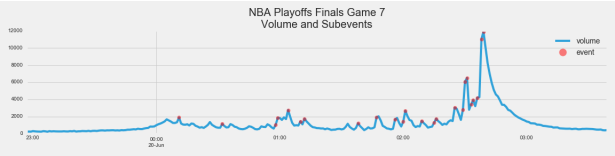

Figure 4: Volume and Top k Subevents

Nichols et al. showed that their implementation of a similar algorithm for detecting important moments in sporting events performed well when evaluated on recap articles for soccer games.[13] Our application to other events, and especially into the domain of technology conferences limits the ability to perform a similar evaluation as there exists no such standard articles for summarizing these events. Therefore, this research defines the summary moments in terms of moments of collective reaction, and makes the single assumption that the subevents detected are indeed the most desirable moments in the event timeline due to volume of crowd activity per minute. The actual performance of the event detection algorithm to identify moments of importance will be reflected when evaluating the final summary compared to the baseline.

4.2 Classification

Twitter is a social platform with an especially diverse array of crowd sourced natural language data. For content generation machines like chat bots or summarizers, solely relying on all natural language as a source without filtration or annotation can be very dangerous, as demonstrated by a recent Twitter-bot developed by Microsoft which quickly made questionable use of crowd data.[11] For this reason we incorporate a supervised classification model, using human annotated ground truth data, as an aid to help improve the phrase reinforcement algorithm. Specifically, we use it to classify subsequently filter content from each important moment, before summarization.

Classification has been used previously as a method in multi-document summarization to identify only those documents which are considered valuable to the task at hand. However, the notion of a valuable tweet is very subjective and varies across research and application. Since there was no existing dataset ground truth annotations reflecting importance for open domain event summarization on Twitter, we created our own. Despite manually annotating the classifier's training data ourselves, the final evaluation results use crowdsourced annotations and therefore provide the valuable reinforcement to the success of the classifier.

Choudhury et al. classified Twitter users into content generatiojn categories and found that while there are differences in the representation of groups across events, the population of verified journalists or organizations creating consistently reliable and newsworthy content is relatively small compared to total individuals posting content. [5] Using this information, instead of first filtering our document corpus according to user metadata, we make the assumption that any tweet could contain valuable content and annotate a raw sample from all of our events in attempt to identify any content features that would signify our desired annotation. Our desired annotation is of a document which contains news relevant to the event, where news is defined as a report of something that happened recently. The CrowdFlower platform was used to create a standard and open sourced annotation task for making these annotations consistently. The annotations, as document labels for trianing the classification algorimth, were manually annotated by the researchers of this paper to save money and time, although the task was designed for and subsequently used to gather the crowdsourced annotations used in evaluating the final summaries. The annotation task and its output is detailed further in Section 6.2.1 when discussing summary evaluation.

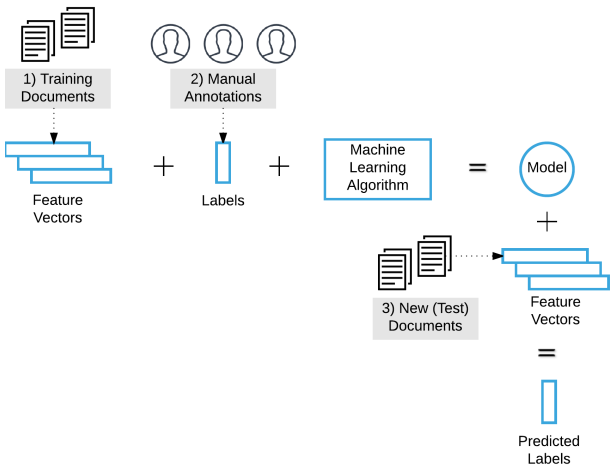

Figure 5: Supervised Classification Flowchart

As diagrammed in Figure 5, a machine learning algorithm is trained by combining a sample of documents across our events with corresponding manually annotations. The statistical model is then applied to predict the class of new documents, for testing as well as for final system application. The task of the statistical model is to classify a document to be within one of two binary classes, 0 being non newsworthy and 1 being newsworthy. During application in the final system, this classification step is applied to each document in each moment of importance after event detection. After labels are predicted for each document, the document corpus for each moment is filtered and only those documents considered newsworthy are sent as an input to the phrase reinforcment algorithm for summarizing.

4.3 Phrase Reinforcement

The Phrase Reinforcement (PR) algorithm is used to extract the most representative single document for each important moment within the event timeline by harnessing language overlap in the document corpus. The algorithm was developed by Beaux P. Sharifi in 2010 [17] specifically for Twitter summarization, and was extensively studied in comparison to state of the art methods in [18] and [16]. The algorithm makes the assumption that there is inherent overlap between the language people use when posting online, specifically when talking about the same topic at the same time.

To apply the algorithm a defined phrase is used as the root of language overlap, and then terms are weighted using a combination of term frequency and distance from a root phrase. Sharifi originally used trending topics as root phrases. [16] More recently, Nichols et al. proposed using PR to extract a single document from multiple individual moments in an event timeline. [13] We adapt this implementation by first finding a common root phrase for each moment, and then weight only those documents containing the root for maximum efficiency. Root phrases are computed using a tiered approach, first considering most occurring single term, then consider top bigram and n-gram if it passes a tuned threshold respectively. The threshold used is frequency greater than 10% of the tier above and at least occurring 10 times.

A sample of four tweets is used for visually expressing the construction of an initial phrase tree. The tweets are taken from the event NBA Playoffs Finals Game 7 at the moment in time 2016-06-20 02:30.

> ***1:*** *Clutch BLOCK from LeBron James, but awful offense from both teams. #NBAFinals*
> ***2:*** *No one can hit a shot down the stretch. Watch a block by Lebron James! #NBAFinals*
> ***3:*** *SUPER HUMAN BLOCK BY LEBRON JAMES!!!!!!! #NBAFinals*
> ***4:*** *Huge clutch block by LeBron James! #NBAFinals #GSWvCLE*

The text is processed for generalizing language across documents to allow for more overlap, which necessary for the success of the PR algorithm. After the tweets are cleaned and have stopwords removed they look like the following.

> ***1:*** *clutch block lebron james awful offense teams nbafinals*
> ***2:*** *one hit shot stretch watch block lebron james nbafinals*
> ***3:*** *super human block lebron james nbafinals*
> ***4:*** *huge clutch block lebron james nbafinals gswvcle*

Building a phrase tree from the above four tweets would result in the below graph where the root phrase is identified as "block lebron james" due to being the most overlapped phrase. The root phrase is denoted in blue, while terms with frequencies greater than one are represented in grey.

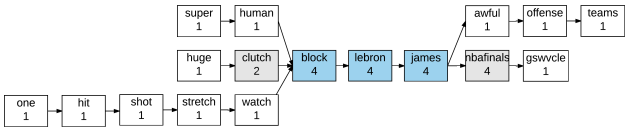

Figure 6: Sample Phrase Reinforcement Tree

After the language tree is built from all documents in the corpus, each node is indexed relative to the root. This allows for the counting of each word at each index, which is represented in the Figure 6. The node count is the number of times that token appears at that node index normalized to the root node. Finally each node for every document is given a weight, the summation of which is the single document weight. Node weights are calculated using:

$$weight_{node} = count_{node} - (distance_{root} * ln(count_{node}))$$

This paper makes an additional step, and proposes adding a social parameter to the weighting scheme. This alteration accounts for social engagements when retweets are excluded during the searching phase for efficiency. The final document weight is calculated using:

$$weight = sum(weight_{node}) + (sum(weight_{node}) * ln(retweets))$$

Documents are then ranked by summary weight descending where the highest weighted document is considered the most representative of that corpus. This process is repeated for each moment to be summarized. In the case of this research this was computed 25 times for each event, and summary type.

4.4 Summarization

Now that each component has been addressed in detail, we are able to describe how they are combined in this research to demonstrate and evaluate improvements on a baseline summarization approach.

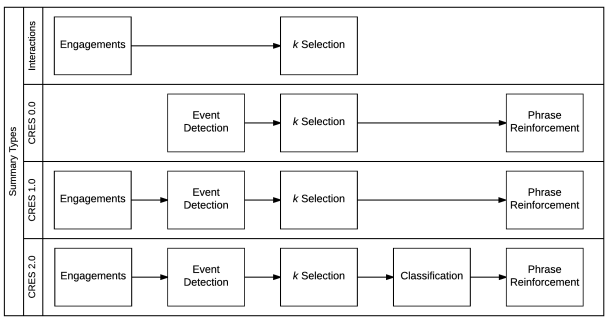

Figure 7: Summarization Method Flowcharts

To create the final summaries we combine and compare multiple combinations of the aforementioned components proven in previous research. Three variations are compared to a baseline approach making four in total.

The baseline is referred to as the Interactions method, and considered to be the current state of the art employed by Twitter. This interactions method is just are rank of all tweets by their total engagements. Engagements is calculated as a sum of retweets and likes. This method is a basic simulation of the actual engagements approach employed by Twitter since they do not disclose their proprietary algorithm. However, Twitter briefly explains top ranked tweets to be *"Tweets that many other Twitter users have engaged with and thought were useful."* [3] The only known difference is that Twitter extends the definition of engagements to include clicks and other interactions with the tweet content, which is information not made available through the public API. [4]

Considering our CRES approach, starting with event detection, we identify spikes in the tweet frequency timeline suggesting moments of importance based on increases in activity frequency. The first method compared to the baseline, referred to as CRES0.0, sends all documents in each of the top k moments as input to the phrase reinforcement algorithm for weighting. Then from each important moment, the top weighted document is selected to be in the summary.

The second method compared to the baseline called CRES1.0, or third in Figure 7, includes an engagements parameter during the event detection calculation which skews the momemnts of importance to reflect times of a higher number of interactions. Instead of representing the distribution by count of tweets per minute, we use $tweetsperminute + tweetsperminute * ln(engagementsperminute)$ per minute for creating the event distribution as well as computing moving average. This gives us an adjusted distribution accounting for subtle changes in engagements, meaning those moments with higher interactions are favored over moments of equal tweet frequency.

Lastly, the final variation to our CRES approach which is referred to in the flowchart and evaluation as CRES2.0, includes the trained, open domain classification model as a corpus filtering step before using phrase reinforcement for summarization. During this method, after top k moments on the minute level have been selected through event detection, each docuent is classified using the model. Only those documents predicted to be newsworthy are sent as an input to the phrase reinforcement algorithm. The intention is that this will contribute to a higher quality corpus for language overlap.

4.5 Evaluation

The most commonly used evaluation methods for summarization systems in previous research use the measures of precision and recall, when comparing expected summary to actual summary where actual summary is a manually created summary from reading all or a significant number of documents. The problem with this method, which also happens to be the reason we create an automated system, is that it has become infeasible for a human to read all documents in the dataset and construct a reasonable summary. That is why in many cases the same old datasets are used, which limits the evaluation method to only those historical in certain domains that may not be applicable to our application. Despite being the most common, Nenkova et al. note that even this method is susceptible to variation in human interpretations, granularity of similar text documents and semantic equivalence among multiple entries. As sometimes experienced during new system development, since there exists no database of annotated tweets for open-domian events on Twitter, we are unable to measure the precision and recall for our entire dataset. While annotation by hand each of the hundreds of thousands of tweets we have collected is not a feasible task, we measure the accuracy of our summary system to in predicting newsworthy content, as measured by the crowd. By using an standard framework for crowdsourcing ground truth, we are able to gather annotations from the public crowd which reflect the views in which the data was created. This is done to eliminate any bias that my arise from a single expert annotation or interpretations of newsworthiness. Our qualitative analysis comparing manual annotations to crowd annotations of tweets being newsworthy showed, that due to high disagreement within the crowd, tweet annotations by the crowd, after CrowdTruth aggregation, were less likely to be considered news. This strict crowd evaluation suggests that the results of experiments reflect lower bounds of summary accuracy, and that our system output could be in fact more valuable when evaluated on specific applications. This also means that during summary evaluation we look specifically for improvements over the baseline, rather than aiming for a summary accuracy towards 100

5. FRAMEWORK

This paper provides and end-to-end framework for a crowd-integrate summarization system on Twitter. The framework is intended to lay the foundation for automating the process of summarizing and visualizing discussions around an event on Twitter. The system code, in modules representing the algorithms previously discussed, is written in python and open sourced via the project GitHub page.[5]

After all summaries are extracted for each method, and all events, we present them in an interactive way. A timeline for each event is created using the volume of tweets per minute, and each important moment is identified by appending the summary tweet. The final visualization is in the form of an interactive dashboard with features such as zoom, dynamic sliders, tool-tips on hover, tweet interactivity and buttons.

The task of this research was to develop an open sourced, and universally applicable, interactive visualization to present the final event summaries. Since no existing frameworks were found in research, we looked towards best practices employed by development at Twitter. It was decided the visualization needs to very simple, with focus on the event timeline and highlighting newsworthy moments within it. The visualization is modeled after developmental work at Twitter known as Reverb. A recent post by Twitter shows an interactive timeline of tweets per minute surrounding Lebron James for the entire NBA playoffs, with specific moments highlighted with single tweets.[6] However, different than Twitter's which requires interaction learn about events, our visualization is focused on highlighting the summary.

[3]https://support.twitter.com/articles/131209

[4]https://support.twitter.com/articles/20171990

[5]

[6]http://reverb.guru/view/078793698302806370

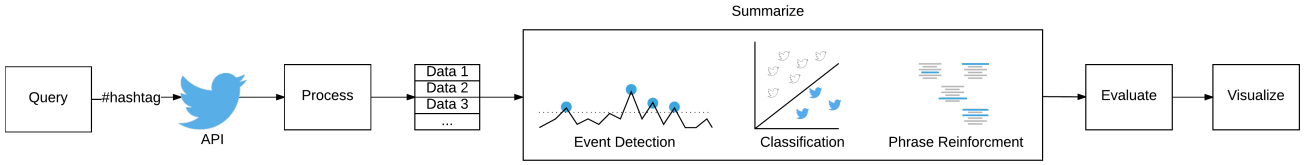

Figure 8: End-to-End Framework

The visualization is designed with the intentions that learning of the event should be obvious and require little effort. A screenshot of the final interactive dashboard, with event timeline visualization, is shown in Figure 9.

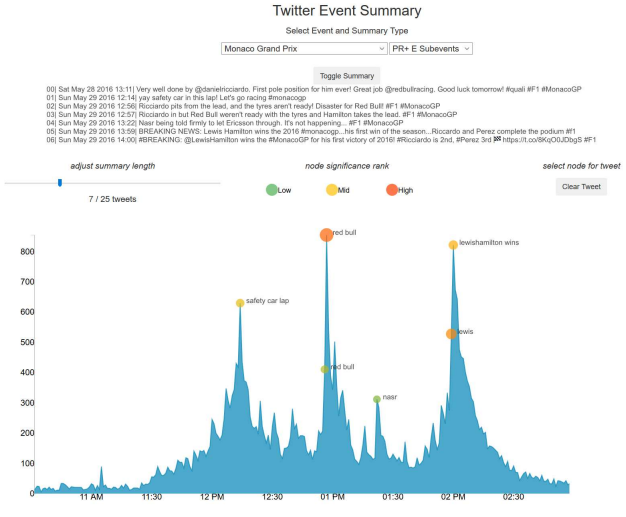

Figure 9: Interactive Event Summary

There are three main components to the visualization: dynamic length. event moments, and interaction. First the summary length can be manipulated using a slider to dynamically change the event coverage. By increasing the number of tweets, the event detection threshold is lowered and therefore, as we show in our evaluation, the number news moments will increase as the precision gradually decreases. Second, each subevent moment is represented by a node on the event timeline. Each node is colored and sized by its significance for capturing the eye of the reader. For summaries using the PR algorithm, the root phrase is appended to each node. Unlike the interactions method, which does not use language overlap and therefor has no identified root of importance, this added feature allows for most efficiently consuming the event's important moments. Lastly, each node is interactive where clicking on the node will display that tweet. Added functionality to the visualization include the ability to toggle the summary, as well as clear the currently displayed tweet. The final dashboard is shown in the Appendix 9, and can be also be seen live on the project GitHub page.

6. EXPERIMENTS

6.1 Classification Research Question

> ***Q1:*** *For filtering newsworthy content by a supervised classifier, can we build a standardized feature set to yield consistently successful results when created on, and tested across, open domains?*

6.1.1 Classification Experimental Setup

As mentioned in section 4.2 we attempt to train an optimal classifier to predict newsworthiness in tweets for open domain events. Therefore we take a random sample from each of our twelve events, and use the standardized annotation task to annotate them consistently on containing news for their respective event. 400 documents from the 12 events, or 4800 total documents were manually annotated as an input to our classification model. As the annotation task has a three point scale of newsworthiness, from not containing news, possibly containing news, and containing news, documents that were annotated to possibly contain news were excluded from training to gain a higher level of separation between the two classes.

Four types of machine learning algorithms were used in testing for our optimal model: Multinomial Naive Bayes, K Nearest Neighbor, Random Forest, and Support Vector Classifier. To evaluate the performance of a classification model alone for predicting news we setup a series of fifteen experimental runs, which will each test one combination of three feature sets, within or across event domains. Each of the four models is trained and tested for each run, and the model with the resulting highest average precision will be selected for final system implementation. The tests are intended to understand how different features, described more below, affect the performance of classifying newsworthy content. One large question we are looking to answer is if we can build a standard feature set to yield consistently successful results when created on, and tested across, open domains.

To combat generalization, or what is referred to as the accuracy paradox, we apply the technique of undersampling to create a more balanced dataset for training. During annotation we found the number of tweets per event identified to contain news ranging anywhere from nine to twenty seven percent. By sampling an equal distribution for the dominant class as done with the other, we can represent each class within the training data the same. This will in turn create a more meaningful accuracy score, and help prevent overfitting or generalization. In addition to undersampling to prevent overfitting, we remove duplicate tweets based on the text field before training.

6.1.2 Features

Feature sets vary in classification research. One common feature generation strategy has been to feature vectors from

Feature	Description	chi-squared	p-value
sfpp	boolean, single first person pronoun	56.621874	5.28×10^{-14}
user_verified	boolean, user is verified	40.839829	1.65×10^{-10}
unique_named_entities	count unique named entities	22.183582	2.48×10^{-6}
hashtag_weight	sum hashtag counts, event scaled	18.07856	2.12×10^{-5}
count_question_marks	count question marks	12.378509	4.34×10^{-4}
term_weight	sum term counts, event scaled	11.950546	5.46×10^{-4}
count_named_entities	count named entities	10.51347	1.19×10^{-3}
nes_cnt_PERSON	count person named entities	10.256057	1.36×10^{-3}
count_characters	count characters	10.146032	1.45×10^{-3}
mention_weight	sum mention counts, event scaled	9.8796	1.67×10^{-3}
tweet_type	boolean, tweet type is media	9.453787	2.11×10^{-3}
tfidf_mean	mean of tweet-event TFIDF vector	8.622938	3.32×10^{-3}
elongation	boolean, contains elongation	8.466972	3.62×10^{-3}
nes_cnt_ORGANIZATION	count organization named entities	7.004034	8.13×10^{-3}
user_bio_len	count characters in user bio	6.386065	1.15×10^{-2}
ellipsis	boolean, contains ellipsis	4.866248	2.74×10^{-2}
hashtags_per_word	frequency hashtags per word	4.486266	3.42×10^{-2}
user_default_profile	boolean, user has default profile	3.704591	5.43×10^{-2}
count_tokens	count tokens in tokenized text	3.675128	5.52×10^{-2}
count_entities_media	count media entities	3.528236	6.03×10^{-2}
text_sentiment_negative	text sentiment negativity from 0 to 1	2.906538	8.82×10^{-2}
count_tokens_stopped	count tokens in tokenize text after stop removal	2.862156	9.07×10^{-2}
text_sentiment_subjectivity	text sentiment subjectivity from 0 to 1	2.796173	9.45×10^{-2}
count_stops	count stop tokens in tokenized text	2.636521	1.04×10^{-1}
retweet_count	count of social retweets	2.339687	1.26×10^{-1}

Table 2: Top k Classification Feature Details

term occurrences, referred to as a bag of words (BOW). First a sparse matrix of size documents by unique terms is created and filled with term counts for each document. To account for differences in document length, each vector is then normalized to total document terms resulting in term frequencies. The commonly employed next step is to scale the term frequencies by the number of term occurrences in the entire corpus. This results in Term Frequency times Inverse Document Frequency (TFIDF) scores which are weighted higher for those terms not occurring as often in the dataset. [12] This approach has been popular and well documented in research, although has limitations. Previous research claiming success from models training using this method does not consider prediction applications of their models outside the datasets used for testing. Since bag of words is built from terms existing in the corpus, the model can only perform well on documents that use the same language. To account for all language in open domain events for example, we would need to gather a large and diverse data sample from all domains. We test these assumptions in our classification experiments and consider an alternative, non semantically constrained feature set.

Non semantically constrained features can be computed on any tweet document. The intention for creating such an open source model, is that anyone could reproduce the feature vectors, on any tweet as long as they have the appropriate tweet JSON object. The advantage to these features over the bag of words is that while the use of language will change over time or across domains, features constructed using tweet structure, semantics or the social data and metadata surrounding how it was created and who created it are universally applied. This assumption is tested as we evaluate both types of features across domains.

As commonly done for classification tasks, we use a univariate statistical test for rank feature selection. We select the top k-best features through a χ^2 test of independence. Top twenty five custom features are shown in the Table 2, as these were used for making the final predictions. We can interpret the chi-squared value in the table below as a decreasing value of dependence between feature and class. The p-value measures the significance of this dependence, where a chi-squared value of greater than 10.83 or p-value less than 0.0001 to be significant. [12]

6.1.3 *Classification Hypotheses*

For our classification experiments we test the following hypotheses:

> ***h1:*** *Considering intra-domain training and testing, BOW features will yield higher precision than custom features.*
> ***h2:*** *Considering inter-domain training and testing, custom features will yield higher precision than BOW features.*
> ***h3:*** *Considering only BOW features, we expect intra-domain tests to yield higher precision than inter-domain tests.*
> ***h4:*** *Considering only custom features, we expect no difference in precision between intra-domain tests and inter-domain tests.*

6.2 Evaluation Research Question

Q2: *Given a set of documents related to an open domain event on Twitter, can we create an automated extraction of k newsworthy documents which, when evaluated on crowd-defined summary accuracy, improves upon the baseline interactions approach utilized by Twitter?*

6.2.1 Evaluation Experimental Setup

In order to evaluate the success of our final summaries, we face a similar issue in most document summarization tasks; there exists no previous ground truth data to evaluate against. Specifically for this type of research in event monitoring and summarization, the ground truth data would be a manually created summary which could be used to measure overlap against and compare metrics such as precision and recall. Due to lack of existing gold standard datasets specifically for open domain Twitter event summarization, and the inability to manually construct a reasonable summary due to sheer volume of data, we are forced like many in previous research to develop our own. Since our problem statement is framed to recieve k documents of event updates, we focus our evaluation on measuring the precision of our k documents to be event updates. Although, alternative to traditional gold standard data collection methods which employ experts and rely on a single answer of truth when annotating, we use an open sourced crowdsourcing approach which relies on subjectivity in human interpretations for gathering a more representative version of the truth. As we are collecting data from the social domain, and summarizing public events using techniques which rely on the language from the crowd, employing the crowd to evaluate our results seems most appropriate. For evaluating success in open domain applications, we argue the crowd's evaluation should be regarded as the most important.

To evaluate our summaries and compare their performance on event summarization, we design an experiment to measure their newsworthiness and compare accuracy. Specifically we are interested in gathering data on whether or not a tweet contains news about an event. This annotation is then correlated with certain quality characteristics, also annotated for each tweet. We gather ground truth annotations for each of the k=25 documents in each summarization approach's summary, for each event. We are then able to measure total accuracy of each summary by taking the amount of news identified in each summary, divided by the number of documents. This is done for the full length summary at k=25 documents as well as for the running sum of documents as k increases from 1, and compared for each event. Aside from length of summary, we are also able to compare summary accuracy across events and approaches to total event volume and the amount of engagements each event received.

6.2.2 Annotation Task

A template for annotating each document is designed using, and crowdsourced through the platform CrowdFlower. [7] This annotation task is shown in Figure 10.

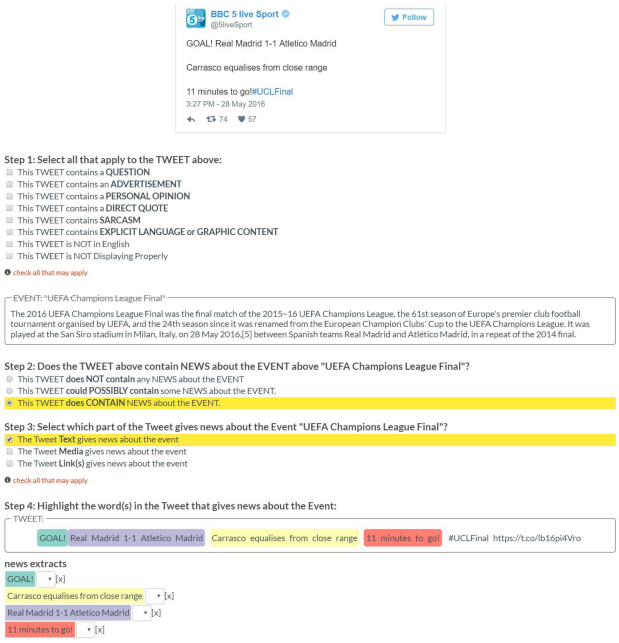

Figure 10: Document Annotation Task

The user is first asked to read a tweet and mark whether a number of quality characteristics apply to it. We then ask the user to read an event summary to become informed of the context of the tweet, which is the first two sentences taken from the event's Wikipedia article, and ask the user to identify if the tweet contains a news update about the event. News is defined for the user as "A report about something that has happened recently."

6.2.3 CrowdTruth Metrics

Unlike traditional methods of ground truth collection which rely on single annotation agreement, the CrowdTruth Framework[8] harnesses disagreement to yield higher quality results. Due to subjectivity in human interpretations of semantic data, disagreement between annotators is naturally captured as more people label natural language data. This is especially true in data that is inherently social, on microblogging sites such as Twitter. It is often not obvious in Tweets, which parts of speech are objects or subjects, and in which tone or context language is written. The inclusion of sarcasm, opinion, slang and hidden sentiment all produce extra ambiguity.

Using the standardized annotation task when gathering annotations, we request multiple annotations from various crowd workers for each document, which accounts for and reflects human subjectivity in the interpretations of each document. Final annotation data amounted to ten annotations per document on each of the twelve events, with four summary methods of twenty five documents each, or $10 annotations * 25 documents * 12 events * 4 summaries = 12000 annotations$. Using CrowdTruth's standardized metrics we then measure the relationship between workers, documents and annotations for identifying any which may signify low quality annotations or crowd workers. A three step process is employed to ensure highest quality when finally aggregating the annotations: identify ambiguous doc-

[7] https://www.crowdflower.com/

[8] https://github.com/CrowdTruth/CrowdTruth

uments, compute spam workers, and aggregate final document annotations. Below is a description of the standard CrowdTruth calculations involved in this process, followed by details on how they are used in harnessing multiple crowd annotations into a single, yet high quality aggregated annotation for each document. More details on these calculations can be found in the python code on the project GitHub page.

Document-Annotation Scores: Calculated as the cosine distance between the each aggregate document vector and the unit vector for every annotation. [2] Cosine Similarity measures the cosine distance from 0 to 1 between two equal length vectors, and is calculated as

$$\cos(d_v, u_v) = \frac{d_v \dot{u}_v}{||d_v||\dot{||}u_v||}$$

Document Clarity: Calculated as the max annotation score for each document. [2] The scores are a measure of clarity, where a higher score represents a more clear interpretation of the annotation for that document. Lower scores depict higher disagreement within the crowd, and suggest difficulty for training a machine.

Worker-Document Disagreement: Calculated as the average of all cosine distances between each worker's document vector and the aggregate document vector, minus that worker. [2]

Worker-Worker Disagreement: Calculated as $1avg(k)$ where k is the pairwise metric, between every worker vector and all other worker vectors. [2]

First we measured the annotation scores for each document and compute a document clarity. We then exclude the documents with clarity scores lower than one standard deviation below the mean from calculating worker metrics.

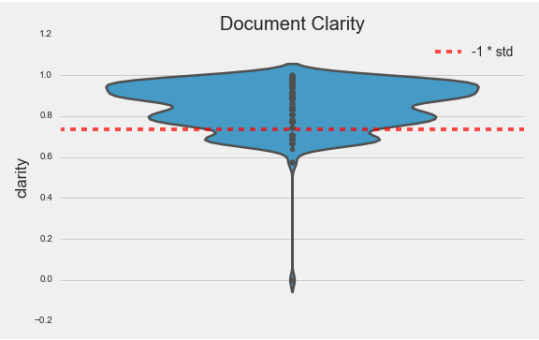

Figure 11: Document Clarity Scores

After computing worker metrics using only those high clarity documents, document-annotation vectors from spam workers are removed where spam workers are defined as outliers in both worker-worker disagreement and worker-document disagreement. An outlier for this dataset was chosen at outside bounds of plus or minus o.75 times one standard deviation from the mean.

Figure 12: Worker-Document Metric

Figure 13: Worker-Worker Metric

We then aggregate the annotation vectors by taking that document annotation with a clarity score greater than .80 as the final annotation. Following the CrowdTruth results demonstrated in [7], we select a threshold for document clarity that yields the highest agreement between the crowd annotation and our expert annotation for those same documents. Dumitrache et al. showed that the higher this score, the easier it is for a machine as well as human to classify.

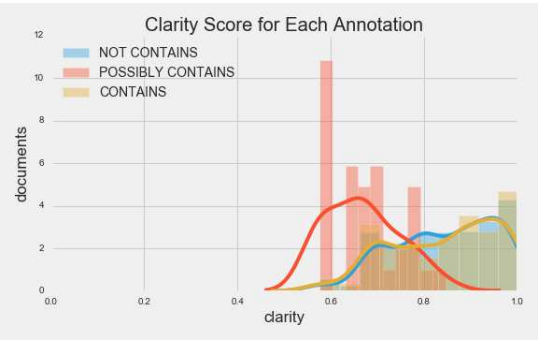

Figure 14: Clarity Scores for Each Annotation

We can see clarity for those documents annotated as containing news or not containing news skew towards 1, meaning their distribution has a higher clarity. While those documents annotated to possibly contain news skew towards a lower clarity score. This reinforces the general assumption of uncertainty, and suggests people's judgements are generally consistent in determining what contains or does not contain news.

After aggregating the annotation scores for each document, and verifying their quality, we are able graph our results and test our hypothesis supporting the underlying research question.

6.2.4 *Evaluation Hypotheses*

For our evaluation experiments we test the following hypotheses:

> ***h5:*** *Summary accuracy of our new approach will significantly improve upon the baseline interactions approach.*
> ***h6:*** *As k increase, total accuracy will also increase.*
> ***h7:*** *As rank increases, total cumulative accuracy will decrease.*
> ***h8:*** *Summary accuracy is positively correlated with event volume.*

7. RESULTS

7.1 Classification Results

For the application of event summarization on Twitter, we considered the scenario of needing to learn as much newsworthy information as possible in a short amount of time. The final summary should present only only that content only which has a high probability of being newsworthy. Therefore we are interested in maximizing precision of isolating only that content considered to be newsworthy, and can likely dismiss recall due to the volume of data we have access to. Precision is the fraction of correctly predicted positives to all predicted as positive, while recall is measured as a fraction of correctly predicted positives to all positives. The F1 score can be interpreted as a weighted mean of these two measures. [12] Our results below show how the precision scores compare across our experimental runs.

Similar to previous research, we found the best performing model to be a Random Forest Classifier, for this task. The Random Forest classifier trained on all 12 events was finally used for predicting documents for the final summarization task. The results of the classification experiments for this top performing model are shown in Table 3.

To better represent the results in the table, we plot the precision scores for each feature set across domains, and feature types.

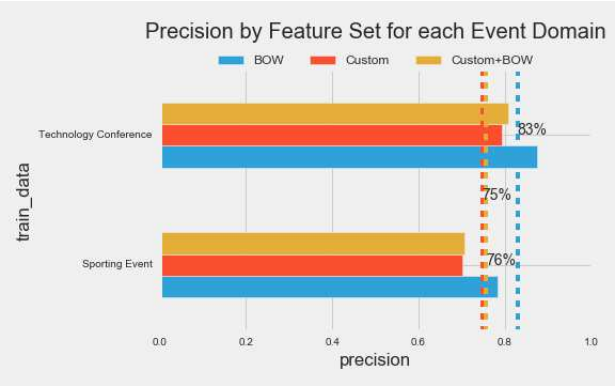

Figure 15: Classification Precision by Feature Set for each Event Domain

As shown in Figure 15, we see the prediction results from using BOW features are slightly higher, on average across domains, for domain-specific tests compared to those modeling results which uses the custom features. This is what we expected, as stated in h1, because the model will identify and weigh heavily on the language specifically found in the news for that one domain.

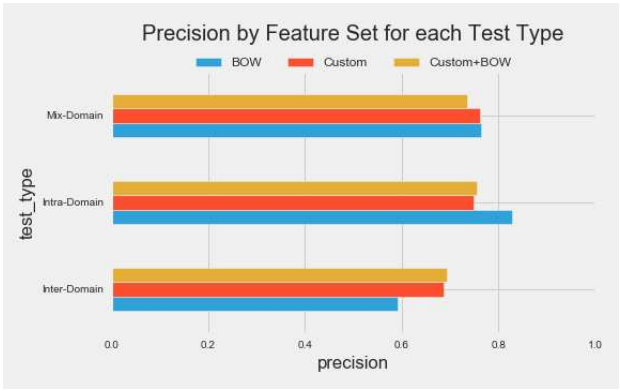

Figure 16: Classification Precision by Feature Set for each Test Type

When analyzing our three tests shown in Figure 16, we observe a slight decrease in the BOW models for inter-domain testing than for intra-domain. This is due to events not using the same language across domains. For example tweets about a technology conference will not contain the words "goal" or "score" for newsworthy content, or even at all, while sporting events will not use language like "stage", "announcing" or "release".

As we hypothesized and shown in Figure 16, while the BOW model results decrease across domain, the custom feature models remain relatively consistent. This is attributed to the fact that unlike the BOW features built on language which differ across event domains, the custom feature set is not semantically constrained to an event or domain and therefore can be built and applied to any event, independent of domain. We would expect that more data, on different domains than the two given, to reinforce this conclusion.

To address our four hypothesis previously stated, we make four unpaired t-tests to find if there are significance differences in the mean performance of our feature sets, within or across domains. Each of the four test results, using degrees of freedom of 2 and a 95% confidence interval, is outlined below according to the respective hypothesis.

h1: Comparing the mean results of each feature set for intra-domain testing, we found no significant difference between the groups.

h1: Feature Sets Within Domain					
Group	**Run**	**Train**	**Test**	**Features**	**Precision**
Group1	Intra	Sport	Sport	Custom	.701299
Group1	Intra	Tech	Tech	Custom	.793814
Group2	Intra	Sport	Sport	BOW	.782051
Group2	Intra	Tech	Tech	BOW	.875
G1: mean=0.747 std=0.065 , G2: mean=0.828 std=0.066 N=2 t=1.2348 std. error=0.66 **p-value=0.3423**					

h2: Comparing the mean results of each feature set for

Run Type	Train	Test	Features	Accuracy	Precision	Recall	F1
Intra-Domain	Sport	Sport	Custom	0.674847	0.701299	0.642857	0.670807
Intra-Domain	Sport	Sport	BOW	0.754601	0.782051	0.7261	0.753086
Intra-Domain	Sport	Sport	Custom+BOW	0.656442	0.705882	0.571429	0.631579
Intra-Domain	Tech	Tech	Custom	0.707071	0.793814	0.669565	0.726415
Intra-Domain	Tech	Tech	BOW	0.691919	0.875	0.547826	0.673797
Intra-Domain	Tech	Tech	Custom+BOW	0.707071	0.806452	0.652174	0.721154
Inter-Domain	Sport	Tech	Custom	0.665657	0.678261	0.630303	0.653403
Inter-Domain	Sport	Tech	BOW	0.557576	0.609195	0.321212	0.420635
Inter-Domain	Sport	Tech	Custom+BOW	0.642424	0.688	0.521212	0.593103
Inter-Domain	Tech	Sport	Custom	0.630221	0.694853	0.464373	0.556701
Inter-Domain	Tech	Sport	BOW	0.519656	0.574074	0.152334	0.240777
Inter-Domain	Tech	Sport	Custom+BOW	0.621622	0.700405	0.425061	0.529052
Mixed-Domain	Mix	Mix	Custom	0.709141	0.761905	0.615385	0.680851
Mixed-Domain	Mix	Mix	BOW	0.720222	0.764706	0.642857	0.698507
Mixed-Domain	Mix	Mix	Custom+BOW	0.692521	0.735099	0.60989	0.666667

Table 3: Classification Experiment Results

inter-domain testing, we do find a significant difference between the groups.

h2: Feature Sets Across Domain					
Group	**Run**	**Train**	**Test**	**Features**	**Precision**
Group1	Inter	Sport	Tech	Custom	.678261
Group1	Inter	Tech	Sport	Custom	.694853
Group2	Inter	Sport	Tech	BOW	.609195
Group2	Inter	Tech	Sport	BOW	.574074
G1: mean=0.689 std=0.012 , G2: mean=0.592 std=0.025 N=2 t=4.8875 std. error=0.019 **p-value=0.0394**					

h3: Comparing the mean results of custom features between intra-domain and inter-domain testing, we find no significant difference between the groups.

h3: Custom Features Within and Across Domain					
Group	**Run**	**Train**	**Test**	**Features**	**Precision**
Group1	Intra	Sport	Sport	Custom	.701299
Group1	Intra	Tech	Tech	Custom	.793814
Group2	Inter	Sport	Tech	Custom	.678261
Group2	Inter	Tech	Sport	Custom	.694853
G1: mean=0.748 std=0.065 , G2: mean=0.687 std=0.012 N=2 t=1.298 std. error=0.047 **p-value=0.3238**					

h4: Comparing the mean results of BOW features between intra-domain and inter-domain testing, we do find a significant difference between the groups.

h4: BOW Features Within and Across Domain					
Group	**Run**	**Train**	**Test**	**Features**	**Precision**
Group1	Intra	Sport	Sport	BOW	.782051
Group1	Intra	Tech	Tech	BOW	.875
Group2	Inter	Sport	Tech	BOW	.609195
Group2	Inter	Tech	Sport	BOW	.574074
G1: mean=0.829 std=0.066 , G2: mean=0.592 std=0.025 N=2 t=4.768 std. error=0.05 **p-value=0.0413**					

From our experiments, the most significant results were in support of our hypotheses h2, and h4. Specifically we found that our custom features performed significantly better than BOW features when evaluated across domain, and that BOW features performed significantly worse across domains than it did within. This results reinforce our assumption, and make it clear that to build an optimal model for open domain classification we need to employ custom features.

We did not find any significant differences in the average model performance between BOW and custom feature sets for intra-domain testing. Our results showed that the tested run within the domain of technology confereces outperformed the sporting domain using either feature set. We attribute our insignificant results to the differences in domain and the fact that we have a small sample size of only two domains. We suggest further tests into new domains would still reinforce our hypothesis.

From our experiments we also observed no difference in prediction results during the mixed domain test. This suggests that, if enough data is sampled from a diverse set of events and domains, the two methods are competitive in success. We therefore conclude that it is better to build an optimal open domain model using our custom feature set, for yielding just as successful results as the BOW features.

Despite the predictive accuracy of our classifier not being especially close to perfect, we propose the classifier will still be helpful when used as an added component to our larger framework, particularly used for filtering the document selection pool at each important moment in an event timeline. Instead of perfectly classifying the news from noise, the classification filter simply enhances the PR algorithm by giving it a document corpus having a higher probability of being newsworthy. Our summary evaluation results discussed below support this claim.

7.2 Evaluation Results

After aggregating the annotation results from our crowdsourcing task we are able to evaluate each summary on its accuracy, or the quality due to appearance of newsworthy tweets. A sample of the aggregated annotation results are included in Table 4. The sample is taken from the top five highest ranked subevents from the top performing events, from the most accurate summary methods.

A quick study of the annotations in Table 4 reflect the

Event	Rank	Text	News
Apple Worldwide Developer Conference	1.0	The iMessage app will make it easier to replace yours words with emoji #WWDC2016	0
Apple Worldwide Developer Conference	2.0	Say hello to Siri on the Mac. #macOS #Sierra #WWDC2016	0
Apple Worldwide Developer Conference	3.0	Cue introduces "Single Sign-On" – sign in once on Apple TV, and get access to all of your streaming apps. Coming to iOS as well. #WWDC2016	1
Apple Worldwide Developer Conference	4.0	Lyrics for Apple Music, cool! #wwdc2016	1
Apple Worldwide Developer Conference	5.0	MANY new updates coming to "Messages" + being opened to developers. #WWDC2016	1
Monaco Grand Prix	1.0	DISASTER for Daniel Ricciardo there. Oh dear. Red Bull gives you wings, but can't always give you tyres. #MonacoGP	0
Monaco Grand Prix	2.0	BREAKING: Lewis Hamilton wins the 2016 #MonacoGP, the 44th victory of his career. Live: #F1	1
Monaco Grand Prix	3.0	Chequered flag! Lewis Hamilton wins the Monaco Grand Prix with Daniel Ricciardo in 2nd and Sergio Perez in 3rd! Max sadly retired #MonacoGP	1
Monaco Grand Prix	4.0	L7: Safety car in the this lap, we're going racing! #MonacoGP	1
Monaco Grand Prix	5.0	Ricciardo responds ... A very slow stop because they did not have the tires ready! Disaster in the Red Bull pits #MonacoGP #F1	1

Table 4: Crowd Annotation Sample for Top Tweets in Highest Performing Summaries

underlying issue with relying solely on event detection and phrase reinforcement, as well as the strictness or uncertainty in the crowd. The top two tweets from the Apple event may reflect the most active moments although present information which may be newsworthy within context, although may not be obvious to an uninformed reader. The top tweet for the Monaco Grand Prix event is not defined as newsworthy again due to ambiguities in the language. The tweet describes the most reacted to moment in the race where a driver's pit crew were not ready for the driver to pit. Although the content posted, and therefore that which comes through our system, was overwhelmingly sarcastic and did not contain explicit descriptions of the situation. This in turn caused uncertainty in the crowd's annotation and therefore resulted in an annotation of non news.

Table 5 contains all crowd annotated accuracies, for each summary. These performance measures for each event are also shown in Figure 17.

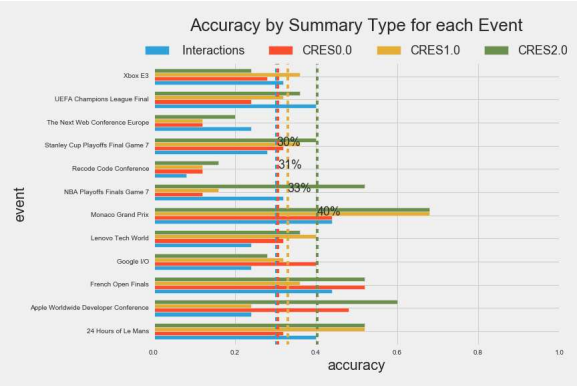

Figure 17: Summary Accuracy by Event

We measure summary accuracy as the number of correct predictions made divided by the total number of predictions made. Meaning accuracy is the total number of newsworthy tweets divided by the total number of tweets. Accuracy is measured for the full length summary at k=*25* documents as well as for the running sum of documents as k increases. Aside from length of summary, we also test to see if accuracy is dependent on event volume or number of engagements received.

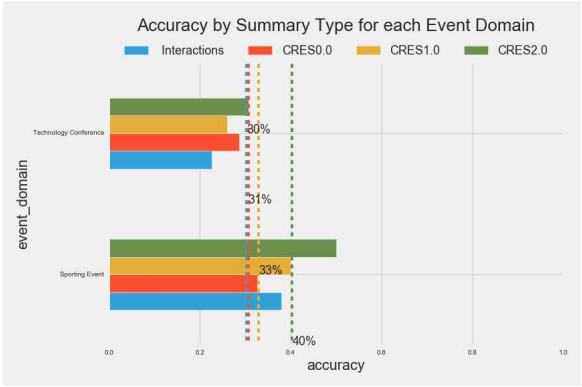

Figure 18: Summary Accuracy by Event Domain

To address our previously stated hypotheses during evaluation we construct a series of statistical tests. First we address if, according to **h5**, the results of our CRES approach have improved upon the baseline approach. We construct an unpaired t-test to measure if there is a significant difference in the accuracy of our full CRES approach from the baseline. When averaging the accuracy of all summaries by each method, and testing for a difference we found no significant difference using a confidence interval of 95%. However, it appeared from our results that there may have been differences in performance changes in each domain we tested on.

Event	Interactions	CRES 0.0	CRES 1.0	CRES 2.0
24 Hours of Le Mans	0.40	0.32	0.52	0.52
French Open Finals	0.44	0.52	0.36	0.52
Monaco Grand Prix	0.44	0.44	0.68	0.68
NBA Playoffs Finals Game 7	0.32	0.12	0.16	0.52
Stanley Cup Playoffs Final Game 7	0.28	0.32	0.36	0.40
UEFA Champions League Final	0.40	0.24	0.32	0.36
Apple Worldwide Developer Conference	0.24	0.48	0.24	0.60
Google I/O	0.24	0.40	0.32	0.28
Lenovo Tech World	0.24	0.32	0.40	0.36
Recode Code Conference	0.08	0.12	0.12	0.16
The Next Web Conference Europe	0.24	0.12	0.12	0.20
Xbox E3	0.32	0.28	0.36	0.24

Table 5: Summary Evaluation Results

Therefore we test to see if there is a significant difference in performance when using the CRES approach on average within a domain. The average performance of each summary approach is visualize in Figure 18.

A similar unpaired t-test is run for each summary approach group, to test if there is a significance difference in the performance of our approach over the baseline in each of the two domains. The difference is tested on the mean performance of summary accuracy for a domain, where *N=6*. We measure at a confidence interval of 95%. When interpreting the results from the sporting events domain, we find that our *p-value=.0476*, does show slightly significant results that we have improved over the baseline. While from the technology domain, the resulting *p-value=.2959* does not reinforce a significant improvement. Despite the trends in Figure 18 showing slight, and incremental increases over the baseline when our CRES approach is used, the results from our experiments are only slightly significant, if not at all on average.

Despite showing only slight improvements on summary accuracy, on average, using this new CRES approach compared to original Interactions approach, we observed two things: the average accuracy is significantly different for each domain, and the improvements for each summary approach differs across domains. First, language surrounding sporting events is much easier to interpret as newsworthy (e.g. scores, saves, crash, penalty), compared to that of technology conferences where the language is more ambiguous and may require event-specific knowledge to fully understand. (e.g. version, keynote, beta, VR). We attribute the lower accuracy to the crowd simply annotating less newsworthy content in the technology domain due to language ambiguity or unfamiliarity. Second, the results from the technology domain support our initial assumptions that each iteration of our summarization methodology should slightly increase the summary accuracy. However, we see different performance improvements for the sporting domain. This may be attributed to the contrast in audiences between the domains and their social engagements. Unlike Technology Conferences which typically attract an audience with specific domain knowledge, audiences of Sporting Events in our dataset were larger. Analyzing audience engagement metrics show the two domains have a comparable number of tweets per user with technology conferences having the higher of the two, while sporting events yield significantly higher engagements per tweet and per user suggesting engagements may play a larger role in determining the moments of activity within the event, instead of using event detection alone.

After evaluating our results regarding improvements over the baseline, we test whether our other hypotheses are true considering trends in summary length, and event metadata. To test these trends we construct a series of linear regressions of summary accuracy, on each variable in question.

To address **h6**, in Figure 19, we observe that as the summary length increases, the amount of news included in the summary also increases. Our results show that this relationship for each summary approach, including the interactions method, is significant.

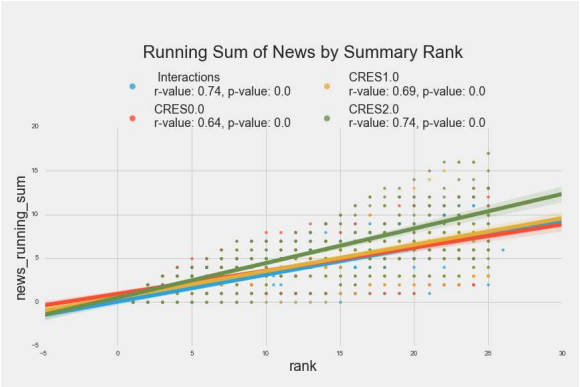

Figure 19: Regression of News by Rank

Figure 20 depicts that as the threshold for summary decreases, or as k increases, the running accuracy slightly decreases. Our results do show, what he hypothesized in **h7**, that the most accurate summaries seem to be those with fewest documents on average, although the results are not that significant, if at all for Interactions or CRES2.0. However, given our sample summary size of only *k=25*, it is possible that this trend is not immediately apparent although could be as the summary increases further in length. More research would need to be conducted in order to claim this statement as true.

Regarding hypothesis **h8**, we do observe a relationship between summary accuracy and data volume, as shown in Figure 21. Although, the results cannot be considered conclusive due to the distribution of our data. Only a single event, NBA Finals, had a volume of above 500,000, which is rep-

resented as an outlier in our dataset. Interestingly though, the results are significant for CRES2.0 and Interactions, but negatively correlated for the CRES0.0 and CRES1.0. Although removing the outlier event returns results that are insignificant for all approaches. A larger sample of events with a more broad distribution in volume would be needed to reinforce our claim. The same conclusions are drawn considering **h9**, as shown in Figure 22. For our sample, the distribution of engagements was similar to the distribution of data, and therefore we find no differences in these two results.

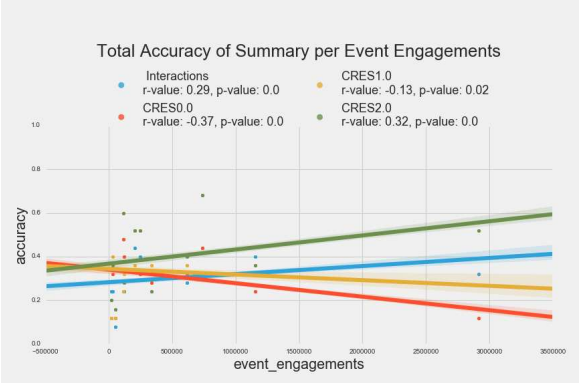

Figure 22: Regression of Accuracy by Event Engagements

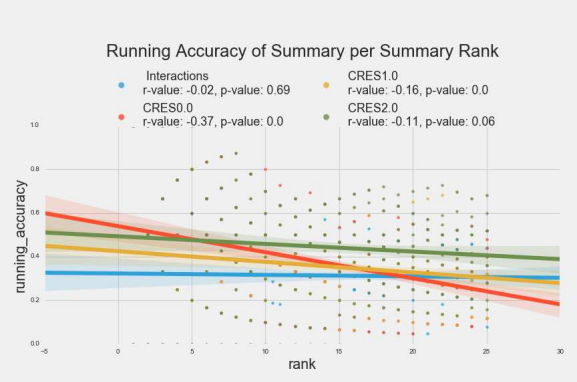

Figure 20: Regression of Accuracy by Rank

7.2.1 Analysis of Task Difficulty

Results from our supervised classifier were not as high as they could have been. We attribute this fact to the difficulty in the task itself. The task of interpreting natural language, especially that which is created on a highly social and contextual platform like Twitter is a difficult task for humans. We observe this difficulty when analyzing our crowdsourcing results.

During the annotation task we gathered data on certain quality characteristics that the crowd may apply to Twitter content including opinions, sarcasm, explicit language and advertisements among others. We correlated these characteristics with each other, as well as news to get a better understanding of how the crowd interprets Twitter content.

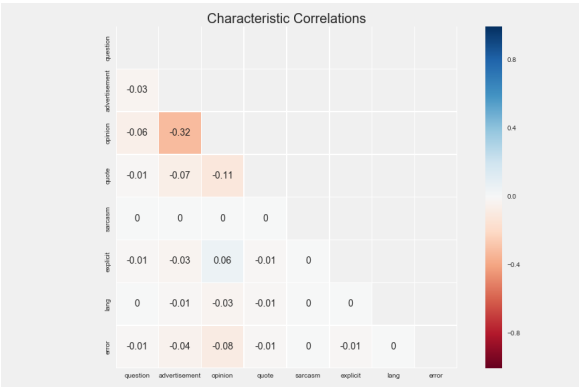

Figure 23: Correlations of Quality Characteristics

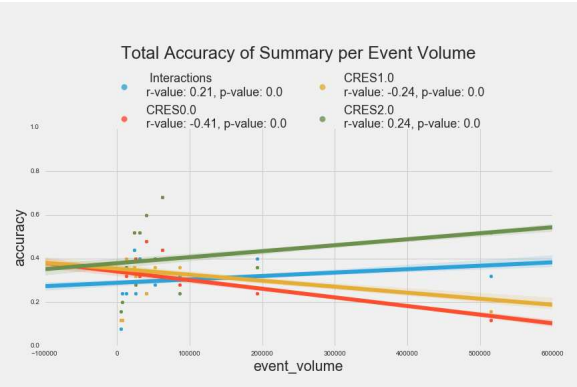

Figure 21: Regression of Accuracy by Event Volume

We observe a strong negative relationship between opinion and advertisements, and a positive relationship between opinions and explicit language. The results suggest there are relationships between characteristics in tweets, although more data would need to be gathered to claim these as significant.

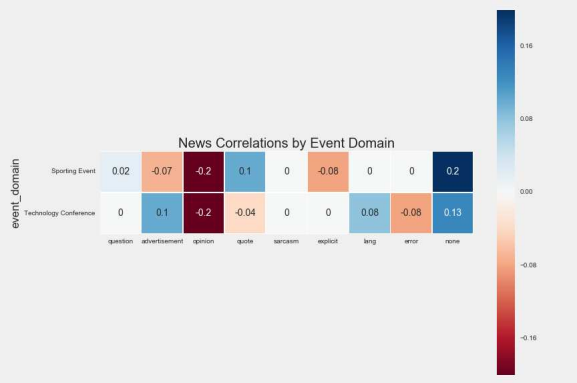

Figure 24: Correlations of News and Quality Characteristics

While certain quality characteristics make annotations unclear, tweets with certain characteristics are sometimes still interpreted as news. We find that correlations between characteristics and annotations differ across events, meaning the interpretation of language depends on the context in which it is created. We observe negative correlations between opinions and news for each event domain, while positive relationships occur between news and other characteristics and these relationships also differ across domains.

8. CONCLUSION

This paper researched an automated solution to the inability of human to wholly consume and comprehend the vast amount of data surrounding topics online. We introduced the Collective Reactions for Event Summarization (CRES) approach, which uses an original combination of proven algorithms to harness signals in online activity, social interactions, content metadata, and language overlap to build comprehensive summaries of events through collective reactions from the crowd.

We provided, and successfully demonstrated the use of, an open sourced framework for automating the extraction of newsworthy documents surrounding the discussion of open domain events on Twitter. The framework is open sourced on GitHub, and exemplified using twelve open domain events. Our experiments tested the research questions considering the ability to: create as standard feature set for consistently classifying newsworthiness in open domain microblog documents, and provide a summary which improves upon the defined baseline when evaluated using CrowdTruth. Our results proved we could conduct a consistent classification on open domain documents, and showed slight improvements to our baseline when evaluating on crowd-defined accuracy for a maxium summary length of 25 documents.

As well we found the replacement of an event detection for an interactions ranking does not necessarily show consistently beneficial results from our experiments. However, the inclusion of an engagements parameter within the event detection algorithm does increase the performance over a baseline interactions approach consistently. Furthermore, by adding a selection pool filtering step using a supervised classification model increases the accuracy of the final summaries on average. We found that the accuracy of the final summaries vary across events and domains, and are dependent on event volume. In addition, we examined how the annotation of tweets is a difficult task and how the variance in human annotation can be seen reflected across event domains as well as other characteristics associated with tweet content.

9. DISCUSSION

The CRES approach given is simple and domain independent, although highly dependent on the data. This means performance varies across events where conversations have inherently different structures. This strategy for success is expressed well by well known phrase in data science, garbage in garbage out. Two things are important here for a quality summarization. Volume and language. First, the volume of digital content must be high enough to reflect the activity of the live event. The event detection algorithm is dependent on an underlying distribution in the data, to project deviations in the form of content bursts from this steady conversation stream. Second, the language used in digital content must represent current happenings in the live event. The Phrase Reinforcement Algorithm depends on language overlap to be successful and therefore will not yield quality results if the language is strictly opinionated, irrelevant or contains no overlap between user posts.

We find the approach works well for events where there exists unanticipated excitement, and the digital conversation emulates these happenings. Bursts in online content, and the language used, is therefore highly reflective of live the activity. As well, the more data generated by the event for including within each time interval, the more language overlap will exist, strengthening the representation of the live event and therefore the strength of the summary algorithm.

The summary is accurate, although highly contextual. When reading the tweets selected for summarization, they are precise at representing the underlying event, although do not express more meaning than what was posted by online users. For example, during sporting events, the summaries display key events like goals, saves, penalties etc. very well, due to finding overlap in the player mentioned in that activity. Although due to the nature of an extractive summary being created by tweets themselves, a summary for a point in time may be "Wow what a save by Oblak!! #uclfinal" instead of what could be presented in a manually created summary looking something like, "Jan Oblak saves with his feet after Gareth Bale takes a freekick for Real Madrid." This paper argues that due to the ability extend this methodology to any domain, the summary generated is highly useful for applications in live event summarization most likely to be adopted by a domain expert where context can be implied.

The interval at one minute works quite well on events with large datasets, where there is already high density of tweets at this level. Sporting events are a good example of this. For events with less data, increasing the threshold will contribute positively to summary generation, as more tweets will have overlapping language, although at a cost of granularity. By including a larger time window of the event, the overlapping terms are seen as more generalized and therefore may not define specific moments in time.

Advantages of our method is that it requires no previous event or domain knowledge base, unlike other supervised approaches. The rudimentary event detection algorithm is easy to implement and is limited in computational resources or scalability requirements. The final summarization product is also dynamically created and can be customized by

multiple tuning parameters. Summary length is the most easily modified and can be modified from a live dashboard to increase or decrease the number of tweets providing information about the event. The n-gram root phrase used in by the phrase reinforcement algorithm can be tuned in length to change the strictness of the algorithm in first looking for language overlap before summarizing. Requiring larger n-grams is expected to produce a more precise description of that sub event although has a higher demand in data, while a single term root will produce a more generalized description around a single term, although will consider a greater number of documents. Lastly the time parameters in the event detection computation can be modified to change the intervals in which tweets are grouped by as well as the window which is considered for moving average deviations. Tuning these parameters all have a direct effect on the summary and should be used with the knowledge of event volume, activity distribution, and the required preciseness of the final summary to document subevents.

10. REFERENCES

[1] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study final report. *Proceedings of the Broadcast News Transcription and Understanding Workshop (Sponsored by DARPA)*, 1998.

[2] L. Aroyo and C. Welty. The three sides of crowdtruth. *Human Computation*, 1(1), 2014.

[3] P. R. Center. The evolving role of news on twitter and facebook.

[4] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[5] M. D. Choudhury, N. Diakopoulos, and M. Naaman. Unfolding the event landscape on twitter: classification and exploration of user categories. *In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 241–244, 2012.

[6] A. Chowdhury. Global pulse, 2011.

[7] A. Dumitrache, L. Aroyo, and C. Welty. Achieving expert-level annotation quality with crowdtruth: The case of medical relation extraction. *ISWC*, BDM2I, 2015.

[8] A. Kejariwal. Breakout detection in the wild, 2014.

[9] A. Kejariwal. Introducing practical and robust anomaly detection in a time series, 2015.

[10] R. Krikorian. New tweets per second record, and how!

[11] P. Lee. Learning from tay's introduction, 2016.

[12] C. Manning, P. Raghavan, and H. Schuetze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[13] J. Nichols, J. Mahmud, and C. Drews. Summarizing sporting events using twitter. *In Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pages 189–198, 2012.

[14] L. Sachetto. Turn to twitter for all things copa america, 2016.

[15] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. *In Proceedings of the 19th international conference on World wide web*, pages 851–860, 2010.

[16] B. Sharifi. Automatic microblog classification and summaraization. *Master's thesis, University of Colorado at Colorado Springs*, 2010.

[17] B. P. Sharifi, M.-A. Hutton, and J. Kalita. Automatic summarization of twitter topics. *National Workshop on Design and Analysis of Algorithm*, 2010.

[18] B. P. Sharifi, D. I. Inouye, and J. K. Kalita. Summarization of twitter microblogs. *The Computer Journal*, bxt109, 2013.

[19] S. Zhao, L. Zhong, J. Wickramasuriya, V. Vasudevan, R. LiKamWa, and A. Rahmati. Sportsense: real-time detection of nfl game events from twitter. 2012.

[20] A. Zubiaga, D. Spina, E. Amigo, and J. Gonzalo. Towards real-time summarization of scheduled events from twitter streams. *In Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 319–320, 2012.

APPENDIX

A. CODE AND LIVE DASHBOARD

The source code, and live dashboard can be found on the project page on GitHub:

`https://github.com/knanne/vu_msc_tweetsumm`