

“Winter is here”: Summarizing Twitter Streams related to Pre-Scheduled Events

Anietie Andy
andyanietie@gmail.com
University of Pennsylvania

Derry Wijaya
wijaya@bu.edu
Boston University

Daphne Ippolito
daphnei@seas.upenn.edu
University of Pennsylvania

Chris Callison-Burch
ccb@cis.upenn.edu
University of Pennsylvania

Abstract

Pre-scheduled events, such as TV shows and sports games, usually garner considerable attention from the public. Twitter captures large volumes of discussions and messages related to these events, in real-time. Twitter streams related to pre-scheduled events are characterized by the following: (1) spikes in the volume of published tweets reflect the highlights of the event and (2) some of the published tweets make reference to the characters involved in the event, in the context in which they are currently portrayed in a sub-event. In this paper, we take advantage of these characteristics to identify the highlights of pre-scheduled events from tweet streams and we demonstrate a method to summarize these highlights. We evaluate our algorithm on tweets collected around 2 episodes of a popular TV show, Game of Thrones, Season 7.

1 Introduction

Every week, pre-scheduled events, such as TV shows and sports games capture the attention of vast numbers of people. The first episode of the seventh season of Game of Thrones (GOTS7), a popular fantasy show on HBO, drew in about 10 million viewers during its broadcast¹.

During the broadcast of a popular pre-scheduled event, Twitter users generate a huge amount of time-stamped tweets expressing their excitement/frustrations, opinions, and commenting about the characters involved in the event, in the context in which they are currently portrayed in a sub-event. For example, the following are some tweets that were published during a three minute time period of an episode of GOTS7:

- Bend the knee... Jon snow #gots7

¹[https://en.wikipedia.org/wiki/Game_of_Thrones_\(season_7\)#Ratings](https://en.wikipedia.org/wiki/Game_of_Thrones_(season_7)#Ratings)

- finally Jon and Dany meet and i'm freaking out #gots7
- Daenerys: have you not come to bend the knee? Jon Snow: i have not Daenerys.

These tweets reflect a part of or the whole scene that happened during this time period on the show i.e. Jon Snow meeting with Daenerys.

Monitoring tweet streams - related to an event, for information related to sub-events can be time consuming partly because of the overwhelming amount of data, some of which are redundant or irrelevant to the sub-event. In this paper, we propose a method to summarize tweet streams related to pre-scheduled events. We aim to identify the highlights of pre-scheduled events from tweet streams related to the event and automatically summarize these highlights. Specifically we evaluate our algorithm on tweets we collected around a popular fantasy TV show, Game of Thrones. We will make this dataset available to the research community². This paper makes the following contributions:

- Identify the highlights of pre-scheduled events from tweets streams related to the event and identify the character that had the most mentions in tweets published during the highlight.
- Identify the context in which this character was being discussed in tweets published during the highlight and summarize the highlight by selecting the tweets that discuss this character in a similar context.

2 Related Work

Some approaches to summarizing tweets related to an event adapt or modify summarization techniques that perform well with documents from news articles and apply these adaptations to tweets. In Sharifi et al. (2010a); Shen et al.

²<https://anietieandy.github.io>

(2013) a graph-based phrase reinforcement algorithm was proposed. In Sharifi et al. (2010b) a hybrid TF-IDF approach to extract one-or-multiple-sentence summary for each topic was proposed. In Liu et al. (2011) an algorithm is proposed that explores a variety of text sources for summarizing twitter topics. In Harabagiu and Hickl (2011) an algorithm is proposed that synthesizes content from multiple microblog posts on the same topic and uses a generative model which induces event structures from the text and captures how users convey relevant content. In Marcus et al. (2011), a tool called "Twitnfo" was proposed. This tool used the volume of tweets related to a topic to identify peaks and summarize these events by selecting tweets that contain a desired keyword or keywords, and selects frequent terms to provide an automated label for each peak. In Takamura et al. (2011); Shen et al. (2013) a summarization model based algorithm was proposed based on the facility location problem. In Chakrabarti and Punera (2011) a summarization algorithm based on learning an underlying hidden state representation of an event via hidden Markov models is proposed. Louis and Newman (2012) proposed an algorithm that aggregates tweets into subtopic clusters which are then ranked and summarized by a few representative tweets from each cluster (Shen et al., 2013). In Nichols et al. (2012) an algorithm was proposed that uses the volume of tweets to identify sub-events, then uses various weighting schemes to perform tweet selection. Li et al. (2017) proposed an algorithm for abstractive text summarization based on sequence-to-sequence oriented encoder-decoder model equipped with a deep recurrent generative decoder. Nallapati et al. (2016) proposed a model using attentional endoder-decoder Recurrent Neural Network.

Our algorithm is different from the previous work in that it identifies the character that had the most mentions in tweets published in a highlight and identifies the context in which this character was being discussed in this highlight; it then summarizes the highlight by selecting tweets that discuss this character in a similar context.

3 Dataset

Our dataset consist of tweets collected around 7 episodes of a popular TV show, GOTS7. We algorithmically identify points of elevated drama or

highlights from this dataset and summarize these highlights.

Each episode of GOTS7 lasted approximately an hour. We used the Twitter streaming API to collect time-stamped and temporally ordered tweets containing "#gots7", a popular hashtag for the show, while each episode was going on. We note that filtering by hashtag gives us only some of the tweets about the show—we omit tweets that used other GOTS7 related hashtags or no hashtags at all. Our dataset consists of the tweet streams for seven episodes of GOTS7; we collected the following number of tweets: 32,476, 9,021, 4,532, 8,521, 6,183, 8,971, and 17,360 from episodes 1,2,3,4,5,6, and 7 respectively.

3.1 Highlight Identification

To identify the highlights of each episode, we plot the number of tweets that were published per minute for each minute of an episode. Since data at the minute level is quite noisy and to smooth out short-term fluctuations, we calculated the mean of the number of tweets published every 3 minutes as shown by the red line in Figure 1, which forms peaks in the tweet volume. We observed the following: (1) the spikes in the volume of tweets correspond to some exciting events/scenes during the show and (2) when there is a spike in the volume of tweets, the characters involved in the sub-events (around that time period) spike in popularity as well in the published tweets. For example, in episode 1, when the character Arya Stark wore Walder Freys face and poisoned all of house Frey, there was a spike in the volume of tweets at this time period; also Arya Stark and Walder Frey spiked in popularity in tweets published in this time period.

Several studies have suggested that a peak needs to rise above a threshold to qualify it as a highlight in a given event. Hence, similar to Shamma et al. (2009); Gillani et al. (2017), we identify highlights of the events by selecting the peaks using the mean and standard deviation of the peaks in all the tweets collected around the 7 episodes of GOTS7.

3.2 Character Identification

To identify the characters involved in GOTS7, we select all the character names listed in the GOTS7 Wikipedia page. It is common for tweets to mention nicknames or abbreviations rather than character full names. For example, in tweets col-

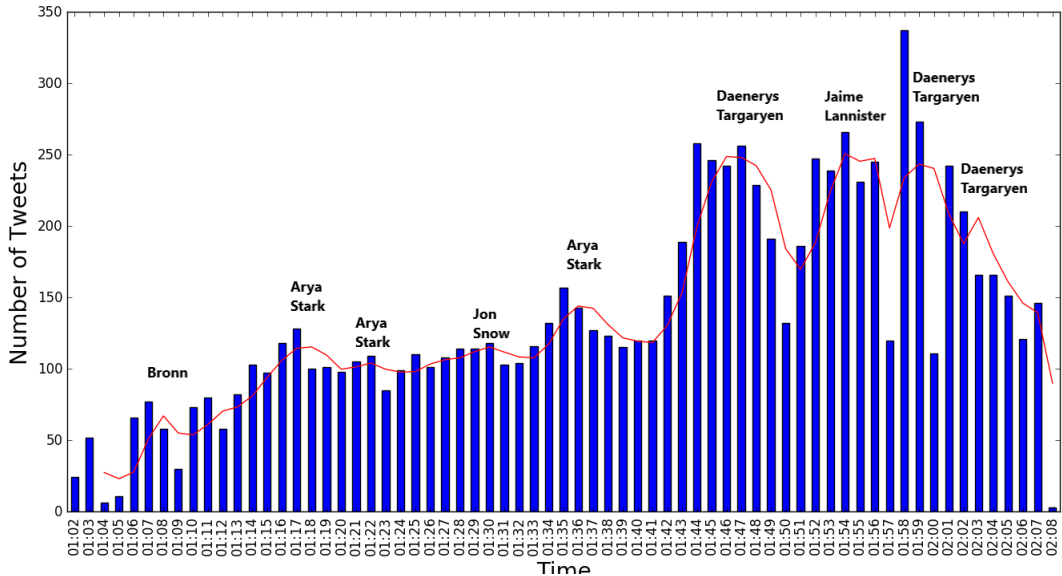


Figure 1: Histogram of number of tweets published per minute during an episode of Game of Thrones. The red line, which forms peaks, shows the mean of the number of tweets published every 3 minutes. The names of the character that had the most mentions during each peak in tweet volume are also shown.

lected around GOTS7 episode 1, the character Sandor Clegane is mentioned 22 times by his full name and 61 times by his nickname “the hound.” Therefore, for each character, we assemble a list of aliases consisting of their first name (which for GOTS7 characters is unique), and the nicknames listed in the first paragraph of the character’s Wikipedia article. All characters end up having at most 2 aliases i.e. their first name and/or a nickname. For example, the nicknames for Sandor Clegane are Sandor and the hound.

To identify the character(s) involved in a highlight from the tweets published during the highlight, we do the following: (1) given the highlight (section 3.1) we count the frequency of mentions of characters in tweets published during the highlight. We select the character with the most mentions. The intuition here is that the character with the most mentions in tweets published in each highlight played a major role in the sub-event that occurred during the highlight.

4 Our Model

We use context2vec [Melamud et al. \(2016\)](#) to create a vector representation for the tweets in each highlight. Context2vec uses an unsupervised neural model, a bidirectional LSTM, to learn sentential context representations that result in comparable or better performances on tasks such as

sentence completion and lexical substitution than popular context representation of averaged word embeddings. Context2vec learns sentential context representation around a target word by feeding one LSTM network with the sentence words around the target from left to right, and another from right to left. These left-to-right and right-to-left context word embeddings are concatenated and fed into a multi-layer perceptron to obtain the embedding of the entire joint sentential context around the target word. Finally, similar to word2vec, context2vec uses negative sampling to assign similar embeddings to this sentential context and its target word. This process indirectly results in sentential contexts, which are associated with similar target words, being assigned similar embeddings.

Given the tweets that mention the character that had the most mentions in tweets published during the time period of a highlight, we want vector representations of tweets that represent the context in which this character is discussed in these tweets. Tweets that discuss the character in a similar context should have similar vector representations. The sentential context representation learned by context2vec is used to find the tweets that best summarize the highlight.

We cluster the context2vec vectors using K-Means. To identify the number of clusters for

Episode	Scene Description	Tweet Summary
Episode 3	Daenerys Targaryen meets Jon Snow	(1) Jon snow telling Daenerys about the white walkers (2) Davos keeping it straight and simple for team Jon snow, the king in the north! (3) Daenerys Targaryen and Jon snow... in the same. room.
Episode 4	Arya displays superb fencing skills as Littlefinger and Sansa watch	(1) Arya got Sansa and Littlefinger shook! (2) Arya and Brienne fighting each other, marvellous piece of telly. (3) so what i'm getting from this is that Arya can beat the hound.

Table 1: Example highlight summaries of Game of Thrones episode 3 and 4

tweets published during the time period of a highlight, we use the elbow method (Kodinariya and Makwana, 2013). For each cluster we choose the five tweets closest to their respective cluster centroids as the tweets that summarize the highlight; these five tweets were concatenated together. We varied the number of tweets we concatenated and five gave the optimal results.

5 Experiments

We evaluate our algorithm on tweets collected around 2 episodes of GOTS7 i.e. episodes 3 and 4. The plot summaries of these GOTS7 episodes are available on the IMDB website. We collected the plot summaries for GOTS7 episodes 3 and 4 from the IMDB website. We compared the summaries from our model to the plot summaries of these episodes from IMDB using the ROUGE metric (Lin, 2004). We compared our model to 5 competitive summarization algorithms and our model performed better than all the baselines in both episodes 3 and 4 as shown in tables 2 and 3. Table 1 shows some of the summaries from our model for a highlight in both episodes 3 and 4.

6 Baselines

LexRank: Computes the importance of textual units using eigenvector centrality on a graph representation based on the similarity of the units (Erkan and Radev, 2004).

TextRank: A graph-based extractive summarization algorithm (Mihalcea and Tarau, 2004).

LSA: Constructs a terms-by-units matrix, and estimates the importance of the textual units based on SVD on the matrix (Gong and Liu, 2001)

Luhn: Derives a significance factor for each textual unit based on occurrences and placements of frequent words within the unit (Luhn, 1958)

Most Retweeted: We select the tweet with the most number of re-tweets in an highlight as a summary of the highlight.

Algorithms	Rouge-1	Rouge-2	Rouge-L
Our Model	38	10	31.5
Luhn	14.4	5	12.3
Most Re-tweets	10	3	10
TextRank	15	2	11.7
LexRank	10	2	10
LSA	10	0	10

Table 2: ROUGE-F1 scores on tweets from highlights in Episode 3

Algorithms	Rouge-1	Rouge-2	Rouge-L
Our Model	33	19.4	25.2
Luhn	16.5	0	13
Most Re-tweets	10	4	10
TextRank	10.5	4	10
LexRank	10	5	10
LSA	11.3	5	10

Table 3: ROUGE-F1 scores on tweets from highlights in Episode 4

7 Conclusion and Future Work

We proposed a model to summarize highlights of events from tweet streams related to the events and showed that our model outperformed several baselines. In the future, we will test our model on tweets collected around other events such as Presidential debates. There were a few cases where our algorithm generated summaries that are somewhat similar, for example: "Arya got Sansa and Littlefinger shook!" and "Littlefinger is shook by Aryas fighting skills". In the future, we will improve the diversity of the generated summaries.

References

- Deepayan Chakrabarti and Kunal Punera. 2011. Event summarization using tweets. *ICWSM*, 11:66–73.
- Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.
- Mehreen Gillani, Muhammad U Ilyas, Saad Saleh, Jalal S Alowibdi, Naif Aljohani, and Fahad S Alotaibi. 2017. Post summarization of microblogs of sporting events. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 59–68. International World Wide Web Conferences Steering Committee.
- Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25. ACM.
- Sanda M Harabagiu and Andrew Hickl. 2011. Relevance modeling for microblog summarization. In *ICWSM*.
- Trupti M Kodinariya and Prashant R Makwana. 2013. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95.
- Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. *arXiv preprint arXiv:1708.00625*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.
- Fei Liu, Yang Liu, and Fuliang Weng. 2011. Why is sxsw trending?: exploring multiple text sources for twitter topic summarization. In *Proceedings of the Workshop on Languages in Social Media*, pages 66–75. Association for Computational Linguistics.
- Annie Louis and Todd Newman. 2012. Summarization of business-related tweets: A concept-based approach. *Proceedings of COLING 2012: Posters*, pages 765–774.
- Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.
- Adam Marcus, Michael S Bernstein, Osama Badar, David R Karger, Samuel Madden, and Robert C Miller. 2011. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 227–236. ACM.
- Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *CoNLL*, pages 51–61.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.
- Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.
- Jeffrey Nichols, Jalal Mahmud, and Clemens Drews. 2012. Summarizing sporting events using twitter. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pages 189–198. ACM.
- David A Shamma, Lyndon Kennedy, and Elizabeth F Churchill. 2009. Tweet the debates: understanding community annotation of uncollected sources. In *Proceedings of the first SIGMM workshop on Social media*, pages 3–10. ACM.
- Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. 2010a. Summarizing microblogs automatically. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 685–688. Association for Computational Linguistics.
- Beaux Sharifi, Mark-Anthony Hutton, and Jugal K Kalita. 2010b. Experiments in microblog summarization. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 49–56. IEEE.
- Chao Shen, Fei Liu, Fuliang Weng, and Tao Li. 2013. A participant-based approach for event summarization using twitter streams. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1162.
- Hiroya Takamura, Hikaru Yokono, and Manabu Okumura. 2011. Summarizing a document stream. In *European conference on information retrieval*, pages 177–188. Springer.