

Indicative Tweet Generation: An Extractive Summarization Problem?

First Author

Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author

Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

Social media such as Twitter have become an important method of communication, with potential opportunities for NLG to facilitate the generation of social media content. We focus on the generation of *indicative tweets* that contain a link to an external web page. While it is natural and tempting to view the linked web page as the source text from which the tweet is generated in an extractive summarization setting, it is unclear to what extent actual indicative tweets behave like extractive summaries. We collect a corpus of indicative tweets with their associated articles and investigate whether they can actually be derived from the articles using extractive methods. We also consider the impact of the formality and genre of the article. Our results demonstrate the limits of viewing indicative tweet generation as extractive summarization, and point to the need for the development of a methodology for tweet generation that is sensitive to genre-specific issues.

1 Introduction

With the rise in popularity of social media, message broadcasting sites such as Twitter and other microblogging services have become an important means of communication, with an estimated 500 million tweets being written every day¹. In addition to individual users, various organizations and public figures such as newspapers, government officials and entertainers have established themselves on social media in order to disseminate information or promote their products.

While there has been recent progress in the development of Twitter-specific POS taggers,

parsers, and other tweet understanding tools (Owoputi et al., 2013; Kong et al., 2014), there has been little work on methods for generating tweets, despite the utility this would have for users and organizations.

In this paper, we study the generation of the particular class of tweets that contain a link to an external web page that is composed primarily of text. This class of tweets, which we call *indicative tweets*, represents a large subset of tweets overall, constituting more than half of the tweets in our data set. Indicative tweets would appear to be the easiest to handle using current methods in text summarization, because there is a clear source of input from which a tweet could be generated. In effect, the tweet would be acting as an indicative summary of the article being linked to, and it would seem that existing methods in summarization can be applied.

There has in fact been some work along these lines, within the framework of extractive summarization. Lofi and Krestel (2012) describe a system to generate tweets from local government records through keyphrase extraction. Lloret and Palomar (2013) compares various extractive summarization algorithms applied on Twitter data to generate tweets from documents.

Lofi and Krestel do not provide a formal evaluation of their model, while Lloret and Palomar compared overlap between system-generated and user-generated tweets using ROUGE (Lin, 2004). Unfortunately, they also show that there is little correlation between ROUGE scores and the perceived quality of the tweets when rated by human users for indicativeness and interest. More scrutiny is required to determine whether the wholesale adoption of methods and evaluation schemes from extractive summarization is justified.

Beyond issues of evaluation measures, it is also unclear whether extraction is the strategy employed by human tweeters. One of the origi-

¹<https://about.twitter.com/company>

nal motivations behind extractive summarization was the observation that human summary writers tended to extract snippets of key phrases from the source text (Mani, 2001). And while it may be true that an automatic tweet generation system need not necessarily follow the same approach to writing as human tweeters, it is still necessary to know what proportion of tweets could be accounted for in an extractive summarization paradigm.

With indicative tweets, an additional issue arises in that the genre of the source text is not constrained. Thus, it may be a news article or an informal blog post. This may be vastly different from the desired formality of tweet itself, and thus, a genre-appropriate extract may not be available.

We begin to address the above issues through a study that examines to what extent tweet generation can be viewed as an extractive summarization problem. We extracted a data set of indicative tweets containing a link to an external article, including the documents linked to by the tweets. We used this data and applied unigram, bigram and LCS (longest common subsequence) matching techniques inspired by ROUGE to determine what proportion of tweets can be found in the linked article. Even with the permissive unigram match measure, we find that well under half of the tweet can be found in the linked article. We also use stylistic analysis on the articles to examine the role that genre differences between the source text and the target tweet play, and find that it is easier to extract tweets from more formal articles than less formal ones.

Our results point to the need for the development of a methodology for indicative tweet generation, rather than to expropriate the extractive summarization paradigm that was developed mostly on news text. Such a methodology will ideally be sensitive to stylistic factors as well as the underlying intent of the tweet.

2 Background and Related Work

There have been studies on a number of different issues related to Twitter data, including classifying tweets and sentiment analysis of tweets. Ghosh et al. (2011) classified the retweeting activity of users based on time intervals between retweets of a single user and frequency of retweets from unique users. ‘Retweet’ here means the occurrence of the same URL in a different tweet. The study was able to classify the retweeting as automatic or

robotic retweeting, campaigns, news, blogs and so on, based on the time-interval and user-frequency distributions. In another study, Chen et al. (2012) were able to extract sentiment expressions from a corpus of tweets including both formal words and informal slang that bear sentiment.

Other studies using Twitter data include O’Connor et al. (2010), who use topic summarization for a given search for better browsing. Chakrabarti and Punera (2011) generate an event summary by learning about the event using a Hidden Markov Model over the tweets describing it. Wang et al. (2014) generate a coherent event summary by treating summarization as an optimization problem for topic cohesion. Inouye and Kalita (2011) compare multiple summarization techniques to generate a summary of multi-post blogs on Twitter.

As described in Section 1, we analyze tweet generation using measures inspired by extractive summarization evaluation. There has been one study comparing different text summarization techniques for tweet generation by Lloret and Palomar (2013). Summarization systems were used to summarize texts into sentences, where were then taken to be tweets. The system-generated tweets were evaluated using ROUGE measures (Lin, 2004). The ROUGE-1, ROUGE-2 and ROUGE-L measures were used, and a human-written reference tweet was taken to be the gold standard.

The limits of extractive summarization have been studied by He et al. (2000). They compare user preferences for various types of summaries of an audio-visual presentation. They demonstrate that the most preferred method of summarization is highlights and notes provided by the author, rather than transcripts or slides from the presentation. Conroy et al. (2006) computed an oracle ROUGE score to investigate the same issue of the limits of extraction.

These studies show that extractive summarization algorithms may not generate good quality summaries despite giving high ROUGE evaluation scores. Cheung and Penn (2013) show that for the news genre, extractive summarization systems that are optimized for centrality—that is, getting the core parts of the text into the summary—cannot perform well when compared to model summaries, since the model summaries are abstracted from the document to a large extent.

3 Data Extraction and Preprocessing

3.1 Using Twitter for Data Extraction

As mentioned earlier, there have been numerous studies that used data from the public Twitter feeds. However, none of the datasets in those studies focused on tweets and related articles linked to these tweets. The dataset of Lloret and Palomar (2013) is one exception, as it contains tweets and the news articles they link to, but it only contains 200 English tweet-article pairs. We chose to build our own dataset. This section describes extraction, cleaning and other preprocessing of the data.

3.2 Extracting Data

Data was extracted from Twitter using the Twitter REST API using 51 search terms, or hashtags. These hashtags were chosen from a range of topics including pop culture, international summit meetings discussing political issues, lawsuits and trials, social issues and health care issues. All these hashtags were trending (being tweeted about at a high rate) at the time of extraction of the data. To get a broader sample, the data was extracted over the course of 15 days in November, 2014, which gave us multiple news stories to choose from for the search terms. The search terms were chosen so that there would be broad representation in terms of various stylistic properties of text like formality, subjectivity, etc. For example, searches related to politics would be more formal, while those related to films would be informal, and would also have a lot more opinion pieces about them. A few examples of the search terms and their distribution in genre are shown in Table 1.

We extracted about 30,000 tweets, of which more than half, or around 16,000, contained URLs to an external news article, photo on photo sharing sites, or video.

The data from the tweets was cleaned by removing the tweets that were not in English as well as the retweets; i.e., re-publications of a tweet by a different user.

We deduplicated the 16,000 extracted URLs into 6,003 unique addresses, then extracted and preprocessed their contents. The `newspaper` package² was used to extract article text and the title from the web page. Since we are interested in text articles that can serve as the source text for summarization algorithms, we needed to re-

Politics	Science & Technology
#apec2014	#rosetta
#G20	#lollipop
#oscarpistorius	#mangalayan
Events	Films and Pop culture
#haiyan	#TaylorSwift
#memorialday	#theforceawakens
#ottawashootings	#johnoliver
International	Sports
#berlinwall	#ausvssa
#ebola	#playingitmyway
#erdogan	#nycmarathon

Table 1: Examples of the hashtags used for extraction, grouped into various categories.

move photos and video links such as those from Instagram and YouTube. To do so, we removed those links that contained fewer than a threshold of 150 words. After this preprocessing, the number of useful articles was reduced from 6003 to 3066. There were some further tweet-article pairs where the text of the tweets was identical, these were removed by further preprocessing and the number of unique tweet-article pairs came down to 2471.

The final version of the data consists of tweets along with other information about the tweet, such as links to articles, hashtags, time of publication, etc. We also retain the linked article text and preprocessed it using the CoreNLP toolkit (Manning et al., 2014). This includes the URL itself and the text extracted from the article, as well as some extracted information such as sentence boundaries, POS tags for tokens, parse trees and dependency trees. These annotations are used later during our analysis in Section 4. Table 2 shows an example of an entry in the dataset.

Tweet	‘#RiggsReport: #CA as the #Election-Night exception. Voters rewarded #GOP nationally, but not in the #GoldenState. http://t.co/K542wvSNVz ’
Title	‘The Riggs Report: California as the Election Night exception’
Text	‘When the dust settled on Election Night last week...’

Table 2: Example of a tweet, title of the article and the text.

²<https://pypi.python.org/pypi/newspaper>

4 Analysis

We now describe the analyses we performed on the data. Our goal is to investigate what proportion of the indicate tweets that we extracted can be found in the articles that they link to, in order to determine whether indicative tweet generation can be viewed as an extractive summarization problem. Table 3 gives an example of data where the tweet that was shared about the article does not come directly from the article text, while Table 4 shows a tweet that was almost entirely extracted from the text of the article, but changed a little for the purpose of readability.

Tweet	Are #Airlines doing enough with #Ebola? http://t.co/XExWwxmjnk #travel
Title	Could shortsighted airline refund policies lead to an outbreak?
Text	The deadly Ebola virus has arrived in the United States just in time for the holiday travel season, carrying fear and uncertainty with it...

Table 3: Example of a tweet, title of the article and the text when tweet cannot be extracted from text.

Tweet	Officer Wilson will be returned to active duty if no indictment , says #Ferguson Police Chief http://t.co/zrIBxMUYJ
Title	Jackson clarifies comments on Wilson’s future status
Text	... Chief Jackson said if the grand jury does not indict Wilson , he will immediately return to active duty

Table 4: Example of a tweet, title of the article and the text when tweet can be extracted from text. The matched portions of the tweet and article are in bold.

We first compute the proportion of tweets that can be recovered directly from the article in its entirety (Section 4.1). Then, we calculate the degree of overlap in terms of unigrams and bigrams between the tweet and the text of the document (Sections 4.2, 4.3).

In addition, we consider locality within the article when computing the overlap. For the unigram analysis, we performed a variant of the analysis,

in which we computed the overlap within three-sentence windows in the source article (Section 4.4). We also compute the least common subsequences between the tweet and the document (Section 4.5). This was done to investigate whether sentence compression techniques could be applied to local context windows to generate the tweet.

These calculations are analogous the ROUGE-1, -2 and -L style calculations. These results give an indication of the degree to which the tweet is extracted from the document text.

For all these analyses, the stop words have been eliminated from the tweet as well as the document, so that only the significant words are taken into consideration. The comparisons were made without lemmatization, and the hashtags, references (@) and URLs from the tweets were all removed.

4.1 Exact Match Calculations

We first checked for a complete substring match of the tweet in the text. Out of the 2471 unique instances of tweet and article pairs, a complete match was found only 23 times. In 9 cases out of these, the tweet text matched the title of the article, which our preprocessing tool did not correctly separate from the body of the article. In the other cases, the text of the tweet appears in its entirety inside the body of the article. This suggests that the user chose the sentence that either seemed to be the most conclusive contribution of the article, or expressed the opinion of the user to be tweeted. An example for this is detailed in Table 5.

Apart from the 9 times where the tweet was matched with title in the article, we also checked to see if the tweet text matched with the article titles that were separately extracted by the `newspaper` package in order to determine if tweets could be generated using the headline generation methods. We found that it did not match with the titles. However, even though there are no exact matches, there might still be some matches where the tweet is a slight modification of the headline of the article, and can be measured using a partial match measure.

4.2 Percentage Match for Unigrams

Next, we did a percentage match with the text of the article. This was a bag-of-words check using unigrams from the tweet and the document. Let $unigrams(x)$ be the set of unigrams for some text x , then u , the percentage of matching unigrams

Tweet	@PNHP: 6. Renounce punitive and counterproductive measures such as sealing the borders, http://t.co/LRLS2MhPRE #Ebola
Title	Physicians for a National Health Program
Text	As health professionals and trainees, we call on President Obama to take the following immediate steps to address the Ebola crisis... 6. Renounce punitive and counterproductive measures such as sealing the borders, and take steps to address the...

Table 5: Example where tweet is extracted as is from the text, matched portion in bold.

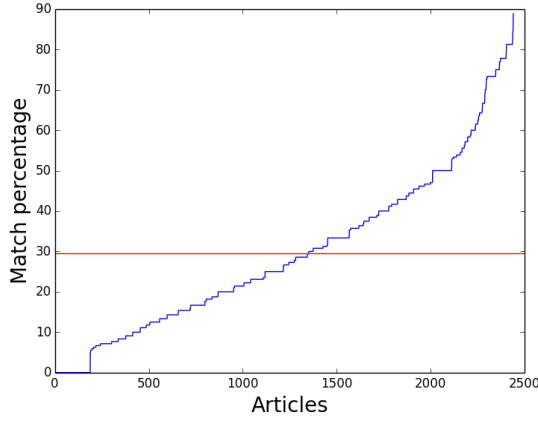


Figure 1: Distribution of unigram match percentage over unique tweets and articles. Mean is 29.53%, indicated by the red horizontal line, with a standard deviation of 20.2%

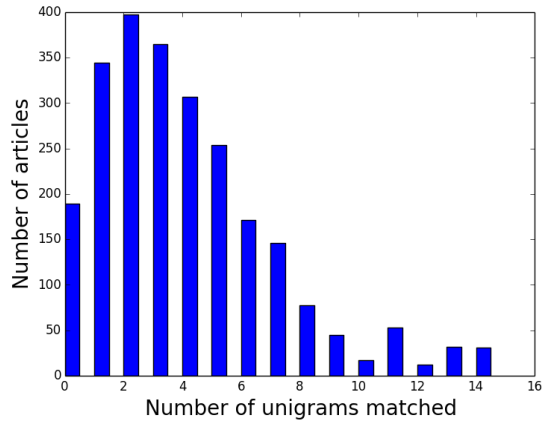


Figure 2: Histogram of number of unique tweet-article pairs vs number of unigrams matched. The mean number of unigrams matched per tweet-article pair is 3.9.

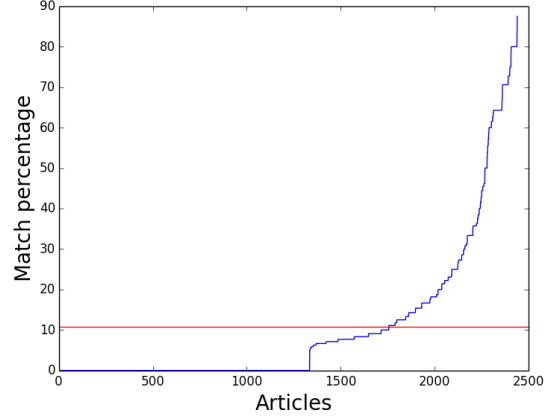


Figure 3: Distribution of bigram match percentage over the tweet-article pair. The mean here is 10.73% shown by the red horizontal line, with a standard deviation of 18.5%

found between a given tweet, t and a given article, a , can be defined as

$$u = \frac{|unigrams(t) \cap unigrams(a)|}{|unigrams(t)|} * 100 \quad (1)$$

Figure 1 shows the percentage of matches in the tweet and the article text as compared to the number of unigrams in the tweet. The mean match percentage is 29.53% and standard deviation is 20.2%. The mean of this distribution shows that the number of matched unigrams from a tweet in the article are fairly low. Figure 2 shows the number of articles with a certain number of matching unigrams. The graph shows that the most common number of unigrams matched was 2. The number of articles with higher unigrams matched goes on decreasing. The slight rise at the end - more than 10 matched unigrams - is accounted for by the completely matched tweets described above.

4.3 Percentage Match for Bigrams

Similar to the unigram matching techniques, the bigram percentage matching was also calculated. The text of the tweet was converted into bigrams and we then looked for those bigrams in the article text. The percentage was calculated similar to the unigram matching done earlier. For the set of bigrams for a text x , $bigrams(x)$, percentage of matching bigrams b for the tweet t and article a is:

$$b = \frac{|bigrams(t) \cap bigrams(a)|}{|bigrams(t)|} * 100 \quad (2)$$

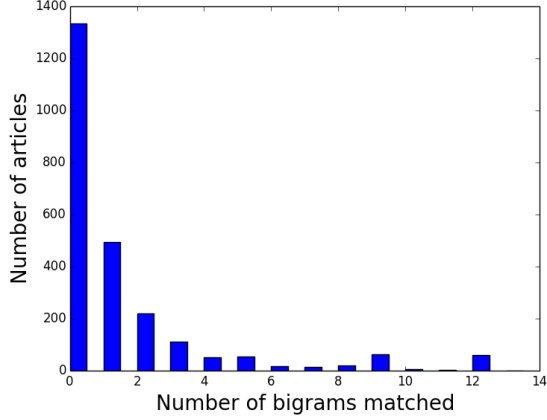


Figure 4: Histogram of number of unique tweet-article pairs vs number of bigrams matched. The mean number of bigrams matched per article is 1.9.

Figure 3 shows the percentages of matched bigrams found. The mean is 10.73 with a standard deviation of 18.5. As seen in the figure, most of the tweet-article pairs have no matched bigrams. The percentage then increases to reflect the complete matches found above.

Figure 4 shows frequency of the number of tweet-article pairs for the number of bigrams matched. There are no matched bigrams for most of the pairs. A smaller number of articles had one matched bigram, and the number decreased until the end, where it increases a little at more than 10 matched bigrams because of exact tweet matches.

4.4 Percentage Match Inside a Window in the Article Text

The next analysis checks for a significant word matching inside a three-sentence window inside the article text. We used a three sentence long window using the sentence boundary information obtained during preprocessing. A window of three sentences was chosen to give a smaller context for the tweet to be extracted from than the entire article. The number was chosen as a moderate context window size as not too small to reduce it to a sentence level, and not too big for the context to be diluted. This was done to investigate whether a pseudo-extractive sentence compression approach could convert a small number of sentences into a tweet.

After the text of the window was extracted, we performed a similar analysis as the last one, except on a smaller set of sentences. The match-

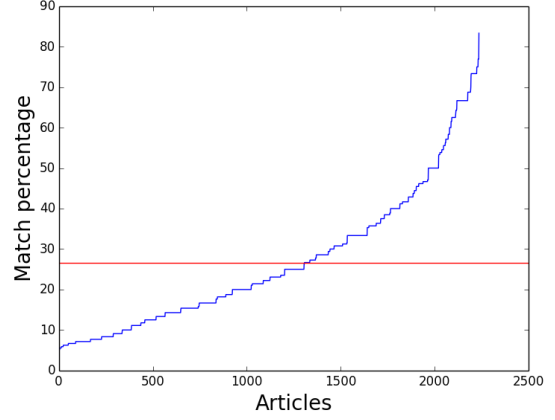


Figure 5: Percentages of common words in tweet and a three sentence window in the article. The maximum match from all percentages is chosen for an article. The red horizontal line is the mean is 26.6%, and standard deviation 17%.

ing percentages from all three-sentence windows in the articles were computed and the maximum out of these was taken for the final results. Let a sentence window w_i be the set of three consecutive sentences starting from the sentence number i . For this window, the unigram match in the tweet t , and the window is the unigram match u calculated above. Then, the maximum match from all the windows, uw is

$$uw = \max_{w_i \in S} u(t, w_i) \quad (3)$$

The result from this experiment is shown in Figure 5. Here, the mean of the values is 26.6% and deviation 17%. Again this shows that only a small proportion of tweets can be generated even with an approach that combines unigrams from multiple sentences in the article.

4.5 Longest Common Subsequence Match Inside a Window for the Text

The percentage match analyses were a bag-of-words approach that disregarded the order of the words inside the texts and tweets. To respect the order of the words in the sentence of the tweet, we also used the least common subsequence algorithm between the tweet text and the document text. This subsequence matching was done inside a sentence window of 5 sentences. Again, the final result for the article was the window in which the maximum percentage was recorded among all windows. The percentage match was calculated

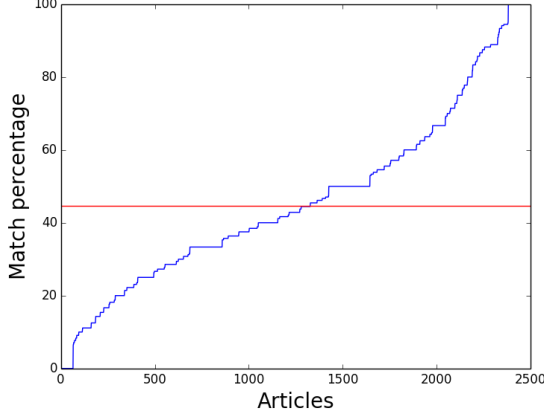


Figure 6: Percentages of words matching in tweet and document text using an LCS algorithm. Mean is 44.6%, which is shown by the red horizontal line, and standard deviation is 22.7%.

using the number of words in the tweet as the denominator.

If $lcs(t, a)$ is the longest common subsequence between the tweet t and article a , $unigrams(x)$ is the set of unigrams for a text x , then the percentage of match for the lcs as compared to the tweet, l is

$$l = \frac{|lcs(t, a)|}{|unigrams(t)|} * 100 \quad (4)$$

These numbers are shown in Figure 6. The mean here is 44.6% and the standard deviation is 22.7%.

5 Interaction with Formality

As seen in the results of the analyses performed in Section 4, the tweets have little in common with the articles they are linked to. This shows that extractive summarization algorithms can only recover a small proportion of the indicative tweets.

To tie in the results of the findings above with some intuitive notions about the text and see how formality interacts with the results, we also calculated the formality of the articles. This formality score was correlated with the longest common subsequence measure that we defined above.

We assume that the formality of an article can be estimated by the formality of the words and phrases in the article. We used the formality lexicon of Brooke and Hirst (2013). They calculate formality scores for words and sentences by training a model on a large corpus based on the appearance of words in specific documents. Their model represents words as vectors and the formal

and informal seeds appear in opposite halves of the graphs, suggesting that we can use these seeds to determine if an article is formal or informal. The lexicon consists of words and phrases and their degree of formality. Thus, more formal words are marked on a positive scale and informal words like those occurring in colloquial language are marked on a negative scale.

Let the set of formality expressions from the lexicon be L , and the formality score for an expression e be $score(e)$. Let the set of all substrings from the article $substrings(a)$ be S . Then, the formality score f for an article a is the number of formal expressions per 10 words in article is

$$f = \frac{\sum_{e \in L \& e \in S} score(e)}{|unigrams(a)|} * 10 \quad (5)$$

The formality lexicon gave positive weights for formal expressions and negative for informal expressions. When we computed f using both formal and informal expressions, we found that the informal words predominated and “swamped” the signal of the formal words, leading to incomprehensible results. Thus, we discarded the informal words and used only the weights from the formal words in our final calculations. To check that these formality scores made sense intuitively, we calculated the average formality score for the articles belonging to each hashtag and ordered them, as shown in Table 7.

Lowest	Highest
#theforceawakens	#KevinVickers
#TaylorSwift	#erdogan
#winteriscoming	#apec

Table 6: Table of hashtags(broadly, topics) with highest and lowest formality according to the lexicon.

This formality score for each article was correlated with the percentage of match obtained using the longest common subsequence algorithm. The Pearson correlation value was 0.41, with a p-value of 7.08e-66, indicating that the interaction between formality and overlap was highly significant. Hence, we can say that the more formal the subject or the article, the better that the tweet can be extracted from the article. Table 7 gives an example of the formality of the article, which is a low 4.2 formality words per 10 words, where the tweet

is not extracted from the article, but rephrased from the article instead.

Tweet	@globetoronto: Why Buffalo got clobbered with snow and Toronto did not. #weather #snow-storm http://t.co/gcwwDPZmX... http://t.co/BXY7EH6F3u "
Title	What caused Buffalos massive snow and why Toronto got lucky
Text	Torontonians have long been the butt of jokes about calling in the army every time a few snow flurries whip by...

Table 7: Example of a tweet, title of the article where the formality of the article is over the mean, and the tweet is extracted from the article.

6 Discussions

Having presented the above statistics showing that only a small portion of indicative tweets can be recovered from the article they link to if viewed as an extractive summarization problem, the question then becomes, how should we view the process of tweet generation?

We think that one promising direction is to model more explicitly the intent, or the purpose of the tweets. There have been several studies on classifying intents in tweets, but in many cases the intents are general, high-level intents of the tweets, more akin to classifying the topic or genre of the tweet than the intent. Wang et al. (2015) classify intents as food and drink, travel, career and so on, ones that can directly be used as intents for purchasing and can be utilized for advertisements. They also focus on finding tweets with intent and then classifying those. Banerjee et al. (2012) analyze real time data to detect presence of intents in tweets. Gómez-Adorno et al. (2014) use features from text and stylistics to determine user intentions, which are classified as news report, opinion, publicity and so on. Mohammad et al. (2013) study the classification of user intents specifically for tweets related to elections. They study one election and classify tweets as ones that agree or disagree with the candidate, ones that are meant for humour, support and so on.

These definitions of intent, while a promising start, will not be sufficient for tweet generation. For this purpose, intent would be the reason the user chose to share the article with that particu-

lar text. This would include reasons like support some cause, promote a product or an article, agree or disagree with an event, or express an opinion about it. Identifying these intents will help provide parameters for generating tweets.

7 Conclusion

We have described a study on investigating whether indicative tweet generation can be viewed as an extractive summarization problem. By analyzing a collection of indicative tweets that we collected according to measures inspired by extractive summarization evaluation measures, we find that most tweets cannot be recovered from the article that they link to, demonstrating a limit to the effectiveness of extractive methods.

We further performed an analysis to determine the role of formality differences between the source article and the Twitter genre. We find evidence that formality is an important factor, as the less formal the source article is, the less extractive the tweets seem to be. Future methods that can change the level of formality of a piece of text without changing the contents will be needed, as will those that explicitly consider the intended use of the tweet.

References

- Nilanjan Banerjee, Dipanjan Chakraborty, Anupam Joshi, Sumit Mittal, Angshu Rai, and Balaraman Ravindran. 2012. Towards analyzing micro-blogs for detection and classification of real-time intentions. In *ICWSM*.
- Julian Brooke and Graeme Hirst. 2013. A multi-dimensional bayesian approach to lexical style. In *HLT-NAACL*, pages 673–679.
- Deepayan Chakrabarti and Kunal Punera. 2011. Event summarization using tweets. *ICWSM*, 11:66–73.
- Lu Chen, Wenbo Wang, Meenakshi Nagarajan, Shaojun Wang, and Amit P Sheth. 2012. Extracting diverse sentiment expressions with target-dependent polarity from twitter. In *ICWSM*.
- Jackie Chi Kit Cheung and Gerald Penn. 2013. Towards robust abstractive multi-document summarization: A caseframe analysis of centrality and domain. In *ACL (1)*, pages 1233–1242.
- John M Conroy, Judith D Schlesinger, and Dianne P O’Leary. 2006. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 152–159. Association for Computational Linguistics.

- Rumi Ghosh, Tawan Surachawala, and Kristina Lerman. 2011. Entropy-based classification of 'retweeting' activity on twitter. *arXiv preprint arXiv:1106.0346*.
- Helena Gómez-Adorno, David Pinto, Manuel Montes, Grigori Sidorov, and Rodrigo Alfaro. 2014. Content and style features for automatic detection of users intentions in tweets. In *Advances in Artificial Intelligence-IBERAMIA 2014*, pages 120–128. Springer.
- Liwei He, Elizabeth Sanocki, Anoop Gupta, and Jonathan Grudin. 2000. Comparing presentation summaries: slides vs. reading vs. listening. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 177–184. ACM.
- David Inouye and Jugal K Kalita. 2011. Comparing twitter summarization algorithms for multiple post summaries. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (Social-Com), 2011 IEEE Third International Conference on*, pages 298–306. IEEE.
- Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archana Bhatia, Chris Dyer, and Noah A. Smith. 2014. A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1001–1012, Doha, Qatar, October. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81.
- Elena Lloret and Manuel Palomar. 2013. Towards automatic tweet generation: A comparative study from the text summarization perspective in the journalism genre. *Expert Systems with Applications*, 40(16):6624–6630.
- Christoph Lofi and Ralf Krestel. 2012. iparticipate: Automatic tweet generation from local government data. In *Database Systems for Advanced Applications*, pages 295–298. Springer.
- Inderjeet Mani. 2001. *Automatic summarization*, volume 3. John Benjamins Publishing.
- Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.
- Saif M Mohammad, Svetlana Kiritchenko, and Joel Martin. 2013. Identifying purpose behind electoral tweets. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, page 1. ACM.
- Brendan O'Connor, Michel Krieger, and David Ahn. 2010. Tweetmotif: Exploratory search and topic summarization for twitter. In *ICWSM*.
- Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390, Atlanta, Georgia, June. Association for Computational Linguistics.
- Lu Wang, Claire Cardie, and Galen Marchetti. 2014. Socially-informed timeline generation for complex events. *constitution*.
- Jinpeng Wang, Gao Cong, Xin Wayne Zhao, and Xiaoming Li. 2015. Mining user intents in twitter: A semi-supervised approach to inferring intent categories for tweets. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.