Automatic 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 formality and genre differences between the article and the tweet. 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 each day[1]. 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 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, 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 from extractive summarization is justified.

Beyond issues of evaluation measure, it is also unclear whether extraction is the strategy employed by human tweeters. One of the original motivations behind extractive summarization was the observation that human summary writers

[1] https://about.twitter.com/company

tended to extract snippets of key phrases from the source text (Mani, 2001). In Twitter data, an additional issue arises in that the genre of the source text, often a news article or other formal text, may be vastly different from the text of the tweet itself. 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 through the tweets. We used this data and applied unigram, bigram and LCS (longest common subsequence) matching techniques inspired by ROUGE to show that we need a more involved approach than directly applying existing extractive summarization algorithms developed for news text. We also use stylistic analysis on the articles to examine the role of genre differences between the source text and the target tweet.

Our results point to the need for the development of a methodology for indicative tweet generation that is sensitive to stylistic factors.

Mention intent of the tweets?

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' in this context was considered as the occurrence of the same URL in a different tweet, and 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 multipost blogs on Twitter.

As described above, we analyze tweet generation using measures inspired by extractive summarization evaluation. There has been one such study comparing different text summarization techniques for tweet generation by Lloret and Palomar (2013). Summarization systems were used to summarize texts to sentences and then were compared against each other, evaluated using the ROUGE metric for evaluation. The ROUGE-1, ROUGE-2 and ROUGE-L metrics were used and the tweets were compared against an ideal summary. ROUGE (Lin, 2004) is a recall based n-gram counting evaluation metric developed for summarization (Nenkova, 2006). ROUGE has been known to work better when multiple reference summaries are used and is not meant to be used at the sentence level. This study uses ROUGE for a single reference summary, which is the reference tweet. However, given the size of a tweet, it can be argued that while generating a reference tweet from a single document, it is difficult to generate multiple reference tweets with largely varying content.

Is this reasoning okay?

The limits of extractive summarization have been studied by He et al. (2000) by comparing user preferences for multiple types of summaries for 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) have defined an oracle score towards the same aim. The oracle score is based on the maximum likelihood probability of words occuring in model summaries and is in turn used to generate summaries that perform better than any extracted and also human-generated summaries. 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 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, since none of the datasets used in these studies focused on tweets and related articles linked to these tweets separated into categories as required for this study, we extracted data directly from the site. The (Lloret and Palomar, 2013) dataset has a similar news article to tweet dataset. However, there are only 200 English tweet to article pairs and aren't separated by search terms on Twitter, 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.

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

Table 1: Table of Hashtags used for extraction. Table shows some examples of search terms chosen from various different categories.

Only English tweets were extracted since the study is limited to English. In the beginning, about 30,000 tweets were extracted, and more than half of these tweets, around 16,000 contained URLs referencing some news articles, photos on photo sharing sites, and videos. The hashtags were chosen to maximise the number of articles related to the tweets. Hence, a lot of topics that were chosen were being tweeted about by news agencies and other popular news sources.

The data from the tweets was cleaned by removing the tweets that were not in English as well as the ones that were retweeted, which is equivalent to re-publishing the same tweet from a different user.

Unique URLs were first extracted from the 16,000 or so URLs in the data. Next, data from these unique URLs was extracted and then preprocessed. The `newspaper` package[2] was used to extract article text and the title from the web page. For the articles obtained from URLs, photos and video links for example, from Instagram and YouTube needed to be removed. For this, the data cleaning was achieved by removing articles by limiting word length of the extracted text to about 150 words. This ensured the removal of photos, videos, advertisements, incorrectly extracted articles from the data. After this preprocessing, the number of useful articles reduced from 6003 to 3066.

The final version of the data consists of all tweets along with all the information of the tweet itself, such as the text of the tweet, links to articles if any, hashtags, and so on. The article links from these tweets are stored as a separate file, with information about the articles themselves, along with some preprocessed data. 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. This processing of the text was done using the CoreNLP toolkit developed at Stanford Manning et al. (2014). These were used later during analysis in Section 4.

A URL could have been tweeted through multiple tweets, all the ids of these tweets are linked to the same URL. It should be noted that the tweet

[2]https://pypi.python.org/pypi/newspaper

to article dataset contains only the articles that are significantly long texts about the subject with a title, and contain no advertisements, other languages, or links to images or videos. 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.

4 Analysis

This section details the analyses performed on the data. The analyses mimic the ROUGE-1,2 and L methods of comparing documents, where we compare the tweet and article text using these methods. This is done to determine whether the tweets promoting the articles could be generated from the document text. The results of the comparison show that the tweet is not extracted from the article text.

We calculate the degree of common words - unigrams and bigrams, between the tweet and the text of the document. We also check least common subsequences between the tweet and the document. These are the ROUGE-1,2 and L style calculations. The hypothesis is that these results give an approximation 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 hashtags, references (@) and URLs from the tweets were also all removed.

4.1 Tweet extracted directly from article

To calculate the position of tweet text as a whole in the text, we 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 23 times. 9 times out of these, the tweet text had been matched against title of the article extracted into the text. The rest of the results are significant, since the text of the tweet appears exactly as is inside the text 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.

Apart from the 9 times that the tweet was matched with text in the article, we also checked to see if the tweet text matched with the article titles that were separately extracted with the newspaper package. We found that it didn't match with the titles. This comparison shows that the tweet is extracted from the article very few times, and does not match with the title of the articles a lot of times either.

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. The order of the words in the tweet or the text did not matter. The results we got seem to suggest that a lot of significant words in the tweet are in fact present in the article. The minimum percentage match obtained was 60%. However, since the order of the words did not matter, this result can be traced back to the fact that tweet is based on the same topic as the document. 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 of the match percentages is 29.53 and standard deviation is 20.2. The graph shows that if compared to the unigrams in a tweet, the number of unigrams matched from that tweet is not consistently high for the tweet describing the article. Figure 2 shows the number of articles with same number of matching unigrams. The graph shows maximum number of articles with 2 unigrams matched. The number of articles with more 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. Let $unigrams(x)$ be the set of unigrams for some text x, then u, the percentage of matching unigrams found between a given tweet, t and a given article, a, can be defined as

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

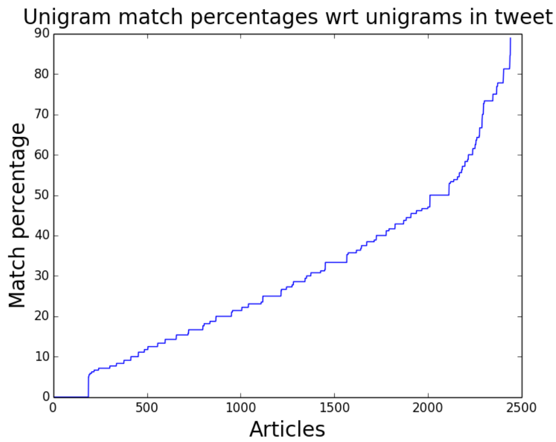

Figure 1: Distribution of unigram match percentage over unique tweets and articles.

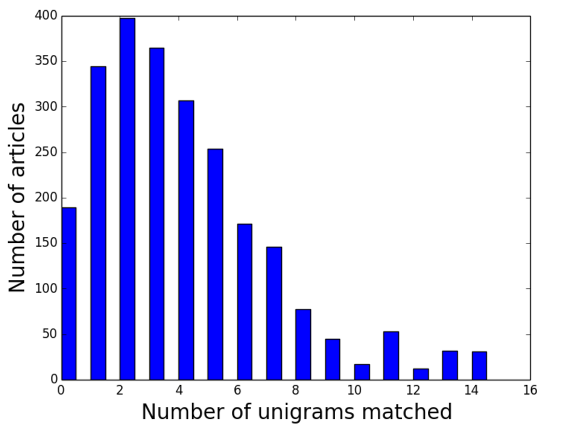

Figure 2: Histogram of number of unique tweet-article pairs vs number of unigrams matched. The mean number of unigrams matched per tweet-article pair is 3.9.

4.3 Percentage match for bigrams

Similar to the unigram matching techniques, 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. 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. The number then decreases from one matched bigram till the end, where it increases a little at more than 10 matched bigrams, similar to the unigram frequency graph. For the set of bigrams for a text x, $bigrams(x)$, percentage of matching bigrams b for the tweet t and article a is calculated as

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

Figure 3 shows the percentages of matched bigrams found in every article. 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.

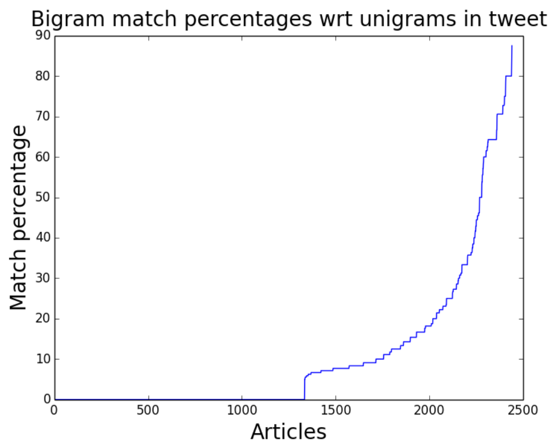

Figure 3: Distribution of bigram match percentage over the tweet-article pair.

4.4 Percentage matching inside a window in the article text

The next analysis was to check 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

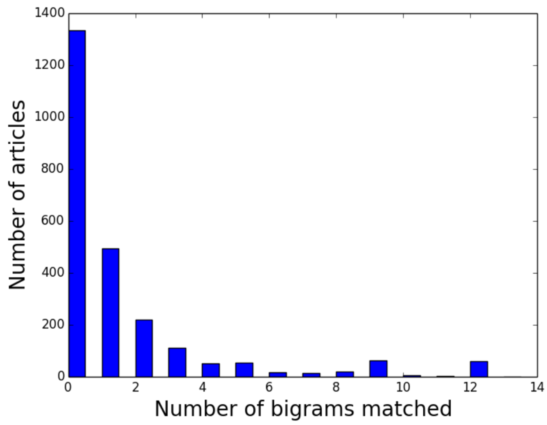

Figure 4: Histogram of number of unique tweet-article pairs vs number of bigrams matched. The mean number of bigrams matched per article is 1.9.

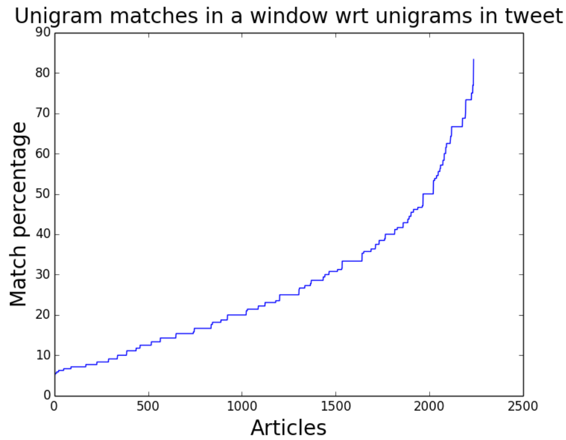

Figure 5: Percentages of common words in tweet and a three sentence window in the aritcle. The maximum match from all percentages is chosen for an article.

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.

After the text of the window was extracted, we performed a similar analysis as the last one, except on a smaller set of sentences. Again, the order of the unigrams didn't matter. Next, the matching percentages from all such windows in the articles were compared and the maximum out of these was considered for the highest match percentage and match position for the final results. The result from this experiment is shown in Figure 5. Here, the mean of the values is 26.6% and deviation 17%. 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

$$uw = max\left(\bigcup_{w_i \in S} u\left(t, w_i\right)\right) \tag{3}$$

4.5 Longest Common Subsequence match inside a window for the text

The percentage match analyses were a bag-of-words approach disregarding 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 against the number of words in the tweet, as found in the least common subsequence calculated between the two texts. These numbers are shown in Figure 6. The mean here is 44.6% and the standard deviation is 22.7%. 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

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

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 related to. The analyses are based on the ROUGE-1,2 and L like calculations. This shows that extractive summarization algorithms cannot be directly applied to articles to generate 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

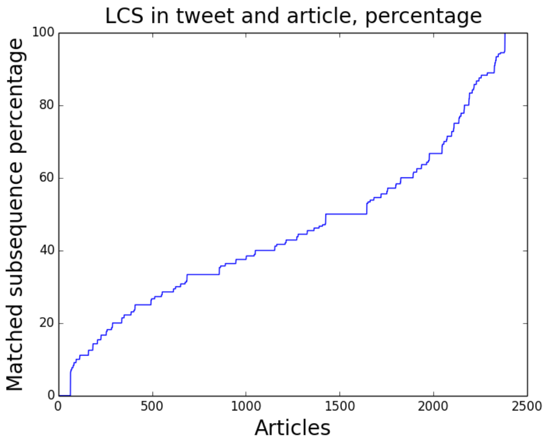

Figure 6: Percentages of words matching in tweet and document text using an LCS algorithm.

subsequence. To achieve this, the degree of formality of the text was calculated with the help of Brooke and Hirst (2013). The formality lexicon was generated by analyzing the stylistics of text and can be used to measure formality of a given text. 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 the degree of formality for their occurrence. 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. The degree of formality was calculated using this lexicon. Let the set of formality expressions from the lexicon be L, where L contains the formality expression, e, and the formality score for this expression is $L(e)$. Let the set of all substrings from the article *substrings*(a) be S. Then, the formality score f for a tweet t and article a is the number of formality expressions per 10 words in article, described as

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

The formality lexicon gave positive weights for formal expressions and negative for informal expressions. After calculating the formality weights for all articles, it was observed that they all had a total negative normalized weight, meaning a lot more informal expressions were getting matched. Hence, we used just the formal word occurrences for calculating the weight. Thus, above a certain cut-off weight, the article could be considered formal, else would be considered informal. To make sure these formality scores intuitively made sense, we calculated the average formality score for each hashtag used in the search during data extraction and ordered them, shown in Table 3

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

Table 3: Table of hashtags(broadly, topics) with highest and lowest formality according to the lexicon.

This formality score for each article was then 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. The p-value justifies that we can reject the null hypothesis, and say with confidence that there is a correlation between the formality scores and the ROUGE-L scores of the tweets and articles. Hence, we can say that the more formal the subject or the article, there are higher chances of the tweet being extracted directly from the article.

Give an example.

6 Discussions

Put discussion on the need to model user intents here. Have examples to highlight need to model intent. Citing these papers is good; say also how their definition of intent is still not sufficient for tweet generation.

A way to get around extractive summarizations for tweet generation and move towards more abstractive solutions is to learn to be able to classigy intent, or the purpose of sharing the tweets. Studies on classifying user intents in tweets are interpreted in different ways. Banerjee et al. (2012) analyze real time data to detect presence of intents in tweets. 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. 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 humor, support and so on.

However, these definitions of intent will still not be sufficient for tweet generation. For this purpose, intent would be the reason the user chose to share the article with that particular 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, which can then be used towards abstractive summarization.

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 more 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.

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 Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 298–306. IEEE.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna 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.

Ani Nenkova. 2006. Summarization evaluation for text and speech: issues and approaches. In *INTERSPEECH*.

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*.