Adaptive Representations for Tracking Breaking News on Twitter

Igor Brigadir and **Derek Greene** and **Pádraig Cunningham**
Insight Centre for Data Analytics
University College Dublin
igor.brigadir@ucdconnect.ie, derek.greene@ucd.ie, padraig.cunningham@ucd.ie

Abstract

Twitter is often the most up-to-date source for finding and tracking breaking news stories. Therefore, there is considerable interest in developing filters for tweet streams in order to track and summarize stories. This is a non-trivial text analytics task as tweets are short, and standard text similarity metrics often fail as stories evolve over time. In this paper we examine the effectiveness of adaptive text similarity mechanisms for tracking and summarizing breaking news stories. We evaluate the effectiveness of these mechanisms on a number of recent news events for which manually curated timelines are available. Assessments based on the ROUGE metric indicate that an adaptive similarity mechanism is best suited for tracking evolving stories on Twitter.

Introduction

Manually constructing timelines of events is a time consuming task that requires considerable human effort. Twitter has been shown to be a reliable platform for breaking news coverage, and is widely used by established news wire services. While it can provide an invaluable source of user generated content and eyewitness accounts, the terse and unstructured language style of tweets often means that traditional information retrieval techniques perform poorly on this type of content.

Recently, Twitter has introduced the ability to construct *custom timelines*[1] or *collections* from arbitrary tweets. The intended use case for this feature is the ability to curate relevant and noteworthy tweets about an event or topic.

We propose an adaptive approach for constructing *custom timelines* - *i.e.* collections of tweets tracking a particular news event, arranged in chronological order. Our approach incorporates the skip-gram neural network language model introduced by Mikolov et al. (2013b) for the purpose of creating useful representations of terms used in tweets. This model has been shown to capture the syntactic and semantic relationships between words. Usually, these models are trained on large static data sets. In contrast, our approach trains models on relatively smaller sets, updated at frequent intervals. Regularly retraining using recent tweets allows our proposed approach to adapt to temporal drifts in content.

This retraining strategy allows us to track a news event as it evolves, since the vocabulary used to describe it will naturally change as it develops over time. Given a seed query, our approach can automatically generate chronological timelines of events from a stream of tweets, while continuously learning new representations of relevant words, phrases, and entities as the story changes. Evaluations performed in relation to a set of real-world news events indicate that this approach allows us to track events more accurately, when compared to nonadaptive models and traditional "bag-of-words" representations.

Problem Formulation

Custom Timelines, curated tweet collections on *Storify*[2], and *Live Blog* platforms such as *Scribblelive*[3] are conceptually similar and are popular with many major news outlets.

For the most part, live blogs and timelines of events are manually constructed by journalists. Rather than automating construction of timelines entirely, our proposed approach offers editorial support for this task, allowing smaller news teams with limited budgets to use resources more effectively. Our contribution focuses on retrieval and tracking rather than new event detection or verification.

We define a timeline of an event as a timestamped set of tweets relevant to a query, presented in chronological order. The problem of adaptively generating timelines for breaking news events is cast as a topic tracking problem, comprising of two tasks:

Realtime ad-hoc retrieval: For each target query (some keywords of interest), retrieve all relevant tweets from a stream posted after the query. Retrieval should maximize recall for all topics (retrieving as many possibly relevant tweets as available).

Timeline Summarization: Given all retrieved tweets relating to a topic, construct a timeline of an event that includes all detected aspects of a story. Summarization involves removal of redundant or duplicate information while maintaining good coverage.

[1] blog.twitter.com/2013/introducing-custom-timelines

[2] www.storify.com

[3] www.scribblelive.com

Related Work

The problem of generating news event timelines is related to topic detection and tracking, and multi-document summarization, where probabilistic topic modelling approaches are popular. Our contribution attempts to utilise a state-of-the-art neural network language model (NNLM) in order to capitalise on the vast amount of microblog data, where semantic concepts between words and phrases can be captured by learning new representations in an unsupervised manner.

Timeline Generation. An approach by Wang (2013) that deals with longer news articles, employed a Time-Dependent Hierarchical Dirichlet Model (HDM) for generating timelines using topics mined from HDM for sentence selection, optimising coverage, relevance, and coherence. Yan et al. (2011) proposed a similar approach, framing the problem of timeline generation as an optimisation problem solved with an iterative substitution approach, optimising for diversity as well as coherence, coverage, and relevance. Generating timelines using tweets was explored by Li & Cardie (2013). However, the authors solely focused on generating timelines of events that are of a personal interest. *Sumblr* (Shou 2013) uses an online tweet stream clustering algorithm, which can produce summaries over arbitrary time durations, by maintaining snapshots of tweet clusters at differing levels of granularity.

Tracking News Stories. To examine the propagation of variations of phrases in news articles, Leskovec et al. (2009) developed a framework to identify and adaptively track the evolution of unique phrases using a graph based approach. In (Chong and Chua 2013), a search and summarization framework was proposed to construct summaries of events of interest. A Decay Topic Model (DTM) that exploits temporal correlations between tweets was used to generate summaries covering different aspects of events. Osborne & Lavrenko (2012) showed that incorporating paraphrases can lead to a marked improvement on retrieval accuracy in the task of First Story Detection.

Semantic Representations. There are several popular ways of representing individual words or documents in a semantic space. Most do not address the temporal nature of documents but a notable method that does is described by Jurgens and Stevens (2009), adding a temporal dimention to Random Indexing for the purpose of event detection. Our approach focuses on summarization rather then event detection, however the concept of using word co-occurance to learn word representations is similar.

Source Data

The corpus of tweets used in our experiments consists of a stream originating from a set of manually curated "newsworthy" accounts created by journalists[4] as Twitter lists. Such lists are commonly used by journalists for monitoring activity and extracting eyewitness accounts around specific news stories or regions. Our stream collects tweets from a total of 16,971 unique users, segmented into 347 geographical and topical lists. This sample of users offers a reasonable coverage of potentially newsworthy tweets, while reducing the need to filter spam and personal updates from accounts that are not focused on disseminating breaking news events. While these lists of users have natural groupings (by country, or topic), we do not segment the stream or attempt to classify events by type or topic.

As ground truth for our experiments, we use a set of publicly available *custom timelines* from Twitter, relevant content from *Scribblelive* liveblogs, and collections of tweets from *Storify*. Each event has multiple reference sources.

It is not known what kind of approach was used to construct these timelines, but as our stream includes many major news outlets, we expect some overlap with our sources, although other accounts may be missing. Our task involves identifying similar content to event timelines posted during the same time periods. Since evaluation is based on content, reference sources may contain information not in our dataset and vice versa. Where there were no quoted tweets in ground truth, the text was extracted as a separate update instead. Photo captions and other descriptions were also included in ground truth. Advertisements and other promotional updates were removed. For initial model selection and tuning, timelines for six events were sourced from Twitter and other live blog sources: the "BatKid" Make-A-Wish foundation event, Iranian Nuclear proliferation talks, a shooting at LAX, Senator Rob Ford speaking at a Council meeting, multiple tornadoes in US midwest, and an alert regarding a possible gunman at Yale University.

These events were chosen to represent an array of different event types and information needs. Timelines range in length and verbosity as well as content type. See Table 4.

"Batkid" can be characterised as a rapidly developing event, but without contradictory reports. "Yale" is also a rapidly developing event, but one where confirmed facts were slow to emerge. "Lax" is a media heavy event spanning just over 7 hours while "Tornado" spans 9 hours and is an extremely rapidly developing story, comprised mostly of photos and video of damaged property. "Iran" and "Robford" differ in update frequency but are similar in that related stories are widely discussed before the evaluation period.

In some cases the same tweets present in a human generated timeline appeared in our automatically generated timelines (see Table 1), providing an indication that our data source provides good coverage of newsworthy sources for a variety of events.

Methods

Short documents like tweets present a challenge for traditional retrieval models that rely on "bag-of-words" representations. We propose to use an alternative representation of short documents that takes advantage of structure and context, as well as content of tweets.

Recent work by (Mikolov et al. 2013a) introduced an efficient way of training a Neural Network Language Model (NNLM) on large volumes of text using stochastic gradient descent. This language model represents words as dense vectors of real values. Unique properties of these representations of words make this approach a good fit for our problem.

[4] Tweet data provided by *Storyful* (www.storyful.com)

Event Period	Ground Truth	Adaptive Approach
15:30 to 16:11	Confirmed report of a person w/ gun on/near Old Campus. SHELTER IN PLACE.	NOW: Police responding to reports of a person with a gun at Yale University. Shelter in Place issued on Central Campus (via @Yale)
17:34 to 18:15	New Haven police spokesman says there is no description of a suspect @Yale and "This investigation is in its infancy" #NHV #Yale	New Haven police spokesman says there is no description of a suspect @Yale and "This investigation is in its infancy" #NHV #Yale
18:57 to 19:38	hartman: possibility that witnesses of long guns saw instead law enforcement officers responding to the scene #Yale	RT @NBCConnecticut: Police say witnesses who saw person with long gun at @Yale could have seen law enforcement personnel. #Yalelockdown

Table 1: A manual selection of retrieved tweets for "Yale" event highlighting key developments, and how the adaptive model can handle concept drift with high recall.

The high number of duplicate and near-duplicate tweets in the stream benefits training by providing additional training examples. For example: the vector for the term "LAX" is most similar to vectors representing "#LAX", "airport", and "tsa agent" - either syntactically or semantically related terms. Moreover, retraining the model on new tweets create entirely new representations that reflect the most recent view of the world. In our case, it is extremely useful to have representations of terms where "#irantalks" and "nuclear talks" are highly similar at a time when there are many reports of nuclear proliferation agreements with Iran.

Additive compositionality is another useful property of the these vectors. It is possible to combine several words via an element-wise sum of several vectors. There are limits to this, in that summation of multiple words will produce an increasingly noisy result. Combined with standard stopword removal, and URL filtering, and removal of rare terms, each tweet can be reduced to a few representative words. The NNLM vocabulary also treats mentions and hashtags as words, requiring no further processing or query expansion. Combining these words allows us to compare similarities between whole tweets.

Timeline Generation

We compare three alternative models to generate timelines from a tweet stream. In each case, we initialize the process with a query. For a given event, the tweet stream is then replayed from the event's beginning to end, with the exact dates defined by tweets in the corresponding human generated timelines. Inclusion of a tweet in the timeline is controlled by a fixed similarity threshold. The stream is processed using a fixed length sliding window updated at regular intervals in order to accommodate model training time.

Pre-processing. A modified stopword list was used to remove Twitter specific terms (*e.g.* "MT", "via"), together with common English stopwords. In the case of NNLM models, stopwords were replaced with a placeholder token, in order to preserve word context. This approach showed an improvement when compared with no stopword removal, and complete removal of stopwords. While the model can be trained on any language effectively, to simplify evaluation only English tweets were considered. Language filtering was performed using Twitter metadata.

Bag-of-Words (tf) Model. A standard term frequency-inverse document frequency model is included as a baseline in our experiments, which uses the cosine similarity of a bag-of-words representation of tweets. We use the same pre-processing steps as applied to the other models. Inverse document frequency counts for terms are derived from the same window of tweets used to train the NNLM approaches. The addition of inverse document frequencies did not offer a significant improvement, as most tweets are short and use terms only once. The term frequency model is moderately adaptive in the sense that the seed query can change as the stream evolves. The seed query is updated if it is similar to the current query, while introducing a number of new terms.

Nonadaptive NNLM. The nonadaptive version of the NNLM model is a static variant where word vectors are initially trained on a large number of tweets, and no further updates to the model are made as time passes.

Adaptive NNLM. The adaptive version uses a sliding window approach to continuously build new models at a fixed interval. The trade-off between recency and accuracy is controlled by altering two parameters: *window length* (*i.e.* limiting the number of tweets to learn from) and *refresh rate* (*i.e.* controlling how frequently a model is retrained). No updates are made to the seed query in both NNLM approaches, only the representation of the words changes after retraining the model.

Post-processing For all retrieval models, to optimise for diversity and reduce timeline length the same summarization step was applied to remove duplicate and near duplicate tweets. Tweets are considered duplicate or near duplicate if all terms excluding stopwords, mentions and hashtags are identical to a tweet previously included in the timeline.

Skip-Gram Language Model

The skip-gram model described in methods section has a number of hyper parameters. Choices for these are discussed here.

Training:

The computational complexity of the skip-gram model is dependent on the number of training epochs E, total number of words in the training set T, maximum number of nearby words C, dimensionality of vectors D and the vocabulary size V, and is proportional to:

$$O = E \times T \times C \times (D + D \times \log_2(V))$$

The training objective of the skip-gram model, revisited in (Mikolov et al. 2013b), is to learn word representations that are optimised for predicting nearby words. Formally, given a sequence of words $w_1, w_2, \ldots w_T$ the objective is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j}|w_t)$$

In effect, word context plays an important part in training the model.

Pre Processing: For a term to be included in the training set, it must occur at least twice in the set. These words are removed before training the model.

Filtering stopwords entirely had a negative impact on overall accuracy. Alternatively, we filter stopwords while maintaining relative word positions.

Extracting potential phrases before training the model, as described in (Mikolov et al. 2013a) did not improve overall accuracy. In this pre-processing step, frequently occurring bigrams are concatenated into single terms, so that phrases like “trade agreement” become a single term when training a model.

Training Objective: An alternative to the skip-gram model, the continuous bag of words (CBOW) approach was considered. The skip-gram model learns to predict words within a certain range (the context window) before and after a given word. In contrast, CBOW predicts a given word given a range of words before and after. While CBOW can train faster, skip-gram performs better on semantic tasks. Given that our training sets are relatively small, CBOW did not offer any advantage in terms of improving training time. Negative sampling from (Mikolov et al. 2013a) was not used. The context window size was set to 5. During training however, this window size is dynamic. For each word, a context window size is sampled uniformly from 1,...k. As tweets are relatively short, larger context sizes did not improve retrieval accuracy.

Vector Representations:

The model produces continuous distributed representations of words, in the form of dense, real valued vectors. These vectors can be efficiently added, subtracted, or compared with a cosine similarity metric.

The vector representations do not represent any intuitive quantity like word co-occurance counts or topics. Their magnitude though, is related to word frequency. The vectors can be thought of as representing the distribution of the contexts in which a word appears.

Vector size is also a tunable parameter. While larger vector sizes can help build more accurate models in some cases, in our retrieval task, vectors larger than 200 did not show an improvement in scores.

Event:	ROUGE-1 Scores							
	Recall				Precision			
	Max	**Adap.**	Static	tf	Max	**Adap.**	Static	tf
batk.	**0.41**	**0.22**	0.20	0.05	**0.44**	0.27	**0.28**	0.10
iran	**0.89**	**0.42**	0.39	0.16	**0.60**	0.16	**0.18**	0.14
lax	**0.87**	**0.19**	0.15	0.09	**0.26**	**0.25**	0.21	0.21
robf.	**0.56**	**0.13**	0.09	0.03	**0.34**	**0.41**	0.40	0.15
torn.	**0.58**	**0.14**	0.12	0.03	**0.17**	0.17	**0.19**	0.10
yale	**0.53**	**0.24**	0.15	0.02	**0.75**	**0.24**	0.18	0.04

Table 2: ROUGE-1 Scores for “Tuning” Events

Event:	ROUGE-2 Scores							
	Recall				Precision			
	Max	**Adap.**	Static	tf	Max	**Adap.**	Static	tf
batk.	**0.24**	**0.09**	0.04	0.01	**0.34**	0.06	**0.06**	0.01
iran	**0.85**	0.12	**0.12**	0.06	**0.58**	0.04	**0.05**	0.04
lax	**0.72**	**0.05**	0.02	0.03	**0.25**	0.06	0.03	**0.08**
robf.	**0.27**	**0.04**	0.03	0.01	**0.29**	**0.11**	0.10	0.03
torn.	**0.43**	**0.04**	0.03	0.01	**0.14**	0.03	**0.04**	0.02
yale	**0.43**	**0.06**	0.03	0.00	**0.79**	**0.09**	0.04	0.00

Table 3: ROUGE-2 Scores for “Tuning” Events

Parameter Selection

Our system has a number of tuneable parameters that suit different types of events. When generating timelines of events retrospectively, these parameters can be adapted to improve accuracy. For generating timelines in real-time, parameters are not adapted to individual event types. For initial parameter selection, a number of representative events was chosen, detailed in Table 2.

For all models, the *seed query* (either manually entered, or derived from a tweet) plays the most significant part. Overall, for the NNLM models, short event specific queries with few terms perform better than longer, expanded queries which benefit term frequency (TF) models. In our evaluation, the same queries were used while modifying other parameters. Queries were adapted from the first tweet included in an event timeline to simulate a lack of information at the beginning of an event.

The *refresh rate* parameter controls how old the training set of tweets can be for a given model. In the case of TF models, this affects the IDF calculations, and for NNLM models, the window contains the preprocessed text used for training. As such, when the system is replaying the stream of tweets for a given event, the model used for similarity calculations is *refresh rate* minutes old.

Window length effectively controls how many terms are considered in each model for training or IDF calculations. While simpler to implement, this fixed window approach does not account for the number of tweets in a window, only the time range is considered. The volume of tweets is not constant over time - leading to training sets of varying sizes. However, since the refresh rate is much shorter than the window length, the natural increase and decrease in tweet volume is smoothed out. On average, there are 150k-200k unique terms in each 24 hour window. Figure 1 shows how varying window size can improve or degrade retrieval performance of different events.

Updating the sliding window every 15 minutes and retraining on tweets posted in the previous 24 hours was found

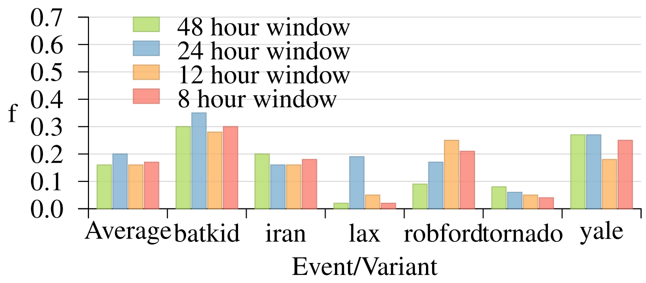

Figure 1: F1 scores for Adaptive model accuracy in response to changing window size

Event Name:	References:	Duration: (Hrs:min)	Total Updates, (Tweets)	Per Window
Batkid	3	5:30	311,(140)	14.14
Iran	4	4:15	198,(190)	11.65
Lax	6	7:15	1236,(951)	42.62
RobFord	4	6:45	1219,(904)	45.15
Tornado	6	9:00	2273,(1666)	63.14
Yale	1	7:15	124,(124)	4.28

Table 4: Details for events used for parameter fitting

to provide a good balance between adaptivity and quality of resulting representations. Larger window sizes encompassing more tweets were less sensitive to rapidly developing stories, while smaller window sizes produced noisier timelines for most events.

Evaluation

In order to evaluate the quality of generated timelines, we use the popular ROUGE set of metrics (Lin 2004), which measure the overlap of ngrams, word pairs and sequences between the ground truth timelines, and the automatically generated timelines. ROUGE parameters are selected based on (Owczarzak et al. 2012). ROUGE-1 and ROUGE-2 are widely reported and were found to have good agreement with manual evaluations. In all settings, stemming is performed, and no stopwords are removed. Text is not preprocessed to remove tweet entities such as hashtags or mentions but URLs, photos and other media items are removed.

To take into account the temporal nature of an event timeline, we average scores across a number of event periods for each variant of the model. This ensures that scores are penalised if the generated timeline fails to find relevant tweets for different time periods as a story evolves. The number of evaluation periods is dependent on the event duration, and selected refresh rate parameter.

“Max” Baseline The “Max” baseline is an illustrative retrieval model, having perfect information about the ground truth and source data. It is designed to represent the maximum achievable score on a metric, given our limited data set and ground truth. For every evaluation period, for each ground truth update, this baseline will select the highest scoring tweet from our stream. This method gives an upper bound on performance for each test event, as it will find the set of tweets that maximise the target ROUGE score directly.

#	Event Name:	References:	Duration: (Hrs:min)	Total Updates, (Tweets)	Updates Per Window
1	Metronorth	3	10:0	483,(480)	12.08
2	Floods	2	10:30	25,(25)	0.60
3	Westgate	4	18:15	73,(62)	1.00
4	MH370	4	7:00	43,(8)	1.54
5	Crimea	1	7:00	34,(34)	1.21
6	Bitcoin	2	4:15	157,(149)	9.24
7	Mandela	2	4:45	89,(51)	4.68
8	WHCD	2	8:00	617,(440)	19.28
9	P.Walker	2	5:45	152,(106)	6.61
10	WWDC14	2	3:30	1069,(81)	76.36

Table 5: Details for events used for evaluation

Event:	ROUGE-1 Scores							
	Recall				Precision			
	Max	**Adap.**	Static	tf	Max	**Adap.**	Static	tf
Met.	**0.55**	**0.24**	0.21	0.10	**0.49**	**0.28**	0.18	0.16
Flo.	**0.32**	0.09	0.01	**0.09**	**0.43**	0.08	0.00	**0.10**
West	**0.63**	0.19	**0.21**	0.03	**0.58**	**0.09**	0.09	0.03
MH3.	**0.46**	**0.35**	0.32	0.09	**0.79**	0.24	**0.28**	0.11
Ukr.	**0.89**	0.16	**0.24**	0.00	**0.91**	0.14	**0.15**	0.00
Bitc	**0.38**	**0.13**	0.07	0.10	**0.44**	0.30	0.20	**0.31**
Man.	**0.35**	**0.16**	0.07	0.08	**0.24**	0.07	**0.07**	0.03
Whc.	**0.18**	0.02	**0.03**	0.02	**0.15**	**0.15**	0.14	0.12
PWa.	**0.61**	**0.30**	0.05	0.09	**0.63**	**0.20**	0.14	0.07
Wwd.	**0.37**	**0.13**	0.06	0.01	**0.54**	**0.50**	0.49	0.13

Table 6: ROUGE-1 Scores for evaluation events, Adaptive approach best on 6/10 events

Performance on unseen Events For initial parameter selection, a number of representative events were selected. We evaluate the system on several new events, briefly described here.

Table 5 gives an overview of the durations, total length, number of reference sources and average number of updates per evaluation period for each event. “Metronorth” timeline describes a Metronorth train derailment, “Floods” deals with flooding in the Solomon Islands, and is characterised by having a low number of potential sources, and sparse updates. “Westgate” follows the Westgate Mall Siege, MH370 details the initial reports of the missing flight, “Crimea” follows an eventful day during the annexation of the Crimean peninsula, “Bitcoin” follows reporters chasing after the alleged creator of Bitcoin, “Mandela” and “P. Walker” are reactions to celebrity deaths, “WHCD” follows updates from the White House Correspondents Dinner, and “WWDC14” follows the latest product launches from Apple - characterised by a very high number of updates and rapidly changing context.

In most cases, shown in Figure 2, our adaptive approach performs well on a variety of events, capturing relevant tweets as the event context changes. This is most notable in the “WWDC14” story, where there were several significant changes in the timeline as new products were announced for the first time. While the adaptive approach can follow concept drift in a news story, it cannot understand or disambiguate between verified and unverified developments, even though relevant tweets are retrieved as the news story

Event	: ROUGE-2 Scores							
	Recall				Precision			
	Max	**Adap.**	Static	tf	Max	**Adap.**	Static	tf
Met.	**0.43**	**0.13**	0.09	0.06	**0.38**	**0.16**	0.08	0.09
Flo.	**0.32**	0.09	0.02	**0.09**	**0.42**	0.08	0.02	**0.10**
West	**0.53**	0.07	**0.08**	0.01	**0.48**	**0.02**	0.02	0.01
MH37	**0.33**	**0.10**	0.10	0.05	**0.48**	0.05	**0.06**	0.01
Ukr.	**0.87**	0.15	**0.22**	0.00	**0.89**	**0.12**	0.11	0.00
Bit.	**0.29**	**0.08**	0.04	0.06	**0.32**	0.16	0.07	**0.22**
Man.	**0.29**	**0.06**	0.03	0.02	**0.18**	0.03	**0.03**	0.00
Whc.	**0.12**	0.01	**0.01**	0.01	**0.10**	**0.09**	0.08	0.09
PWa.	**0.47**	**0.13**	0.02	0.03	**0.45**	**0.07**	0.04	0.02
Wwd.	**0.11**	**0.02**	0.01	0.00	**0.17**	**0.08**	0.07	0.02

Table 7: ROUGE-2 Scores for evaluation events, NNLM approaches best on 6/10 events

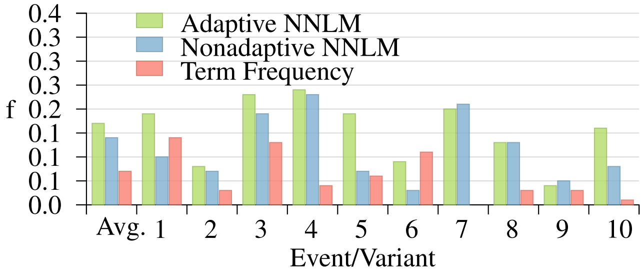

Figure 2: F1 Scores for each Model

evolves, incorrect or previously debunked facts are still seen as relevant, and included in the generated timeline.

Overall the adaptive NNLM approach performs much more effectively in terms of recall rather than precision. A more effective summarization step could potentially improve accuracy further. This property makes this model suitable for use as a supporting tool in helping journalists find the most relevant tweets for a timeline or liveblog.

The Nonadaptive approach performs well in cases where the story context does not change much, tracking reactions of celebrity deaths for example. Timelines generated with this variant tend to be more general.

While the additive compositionality of learnt word representations works well in most cases, there are limits to this usefulness. Short, focused seed queries tend to yield better results. Longer queries benefit baseline term frequency models but hurt performance of the NNLM approach.

Future and Ongoing Work

Currently, there is a lack of high quality annotated Twitter timelines available for newsworthy events. This is perhaps unsurprising, as current methods provided by Twitter for creating custom timelines are limited to either manual construction, or through a private API. Other forms of liveblogs and curated collections of tweets are more readily available, but vary in quality. As new timelines are curated, we expect that the available set of events to evaluate will grow. In the interest of reproducibility, we make our dataset of our reference timelines and generated timelines available[5].

We adopted an automatic evaluation method for assessing timeline quality. A more qualitative evaluation involving potential users of this set of tools is currently in progress.

We have compared one unsupervised way of generating word vectors in a semantic space against a Term Frequency based approach, but other techniques may provide a better baseline to compare against (Widdows and Cohen 2010). There is also room for improving the model retraining approach. Rather than updating the model training data with a fixed length moving window over a tweet stream, the model could be retrained in response to tweet volume or another indicator, such as the number of "out of bag" words, *i.e.* words for which the model does not have embeddings for.

The quality of the word embeddings depends mostly on the size of the training set, and our approach requires a relatively fast training time in order to generate a timely model. There is a trade-off between quality and ability to adapt, that is dependent on the number of unique words, so a further improvement might be gained by maintaining a foreground model that's updated frequently, as well as a background model built on a much larger data set. Compositionality of word embeddings no longer applies when using representations from different models, so an alternative method of calculating term similarities across a foreground and background model would need to be developed. Retrieval accuracy is also bound by the quality of our curated tweet stream, expanding this data set would also improve retrieval accuracy.

Conclusion

The continuous skip-gram model trained on Twitter data has the ability to capture both the semantic and syntactic similarities in tweet text. Creating vector representations of all terms used in tweets enables us to effectively compare words with account mentions and hashtags, reducing the need to pre-process entities and perform query expansion to maintain high recall. The compositionality of learnt vectors lets us combine terms to arrive at a similarity measure between individual tweets.

Retraining the model using fresh data in a sliding window approach allows us to create an adaptive way of measuring tweet similarity, by generating new representations of terms in tweets and queries at each time window.

Experiments on real-world events suggest that this approach is effective at filtering relevant tweets for many types of rapidly evolving breaking news stories, offering a useful supporting tool for journalists curating liveblogs and constructing timelines of events.

Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

We thank Storyful for providing access to data, and early adopters of custom timelines who unknowingly contributed ground truth used in the evaluation.

[5] http://mlg.ucd.ie/timelines

References

Chong, F., and Chua, T. 2013. Automatic Summarization of Events From Social Media. In *Proc. 7th International AAAI Conference on Weblogs and Social Media (ICWSM'13)*.

Jurgens, D., and Stevens, K. 2009. Event detection in blogs using temporal random indexing. *Proceedings of the Workshop on Events in ...* 9–16.

Leskovec, J.; Backstrom, L.; and Kleinberg, J. 2009. Memetracking and the dynamics of the news cycle. *Proc. 15th ACM SIGKDD international conference on Knowledge discovery and data mining* 497.

Li, J., and Cardie, C. 2013. Timeline Generation : Tracking individuals on Twitter. *arXiv preprint arXiv:1309.7313*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In Marie-Francine Moens, S. S., ed., *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS'13*, 1–9.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013b. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Owczarzak, K.; Conroy, J. M.; Dang, H. T.; and Nenkova, A. 2012. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, 1–9. Stroudsburg, PA, USA: Association for Computational Linguistics.

Petrović, S.; Osborne, M.; and Lavrenko, V. 2012. Using paraphrases for improving first story detection in news and Twitter. In *Proc. Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 338–346.

Shou, L. 2013. Sumblr: Continuous Summarization of Evolving Tweet Streams. In *Proc. 36th SIGIR conference on Research and Development in Information Retrieval*, 533–542.

Wang, T. 2013. Time-dependent Hierarchical Dirichlet Model for Timeline Generation. *arXiv preprint arXiv:1312.2244*.

Widdows, D., and Cohen, T. 2010. The Semantic Vectors Package: New Algorithms and Public Tools for Distributional Semantics. *Proc. 4th IEEE International Conference on Semantic Computing* 9–15.

Yan, R.; Wan, X.; Otterbacher, J.; Kong, L.; Li, X.; and Zhang, Y. 2011. Evolutionary Timeline Summarization : a Balanced Optimization Framework via Iterative Substitution. In *Proc. 34th SIGIR Conference on Research and development in Information Retrieval*, 745–754.