Extractive and Abstractive Event Summarization over Streaming Web Text

Chris Kedzie
Dept. of Computer Science
Columbia University
kedzie@cs.columbia.edu

Adviser: Kathleen McKeown
Dept. of Computer Science
Columbia University
kathy@cs.columbia.edu

During crises, information is critical for responders and victims. When the event is significant, as in the case of hurricane Sandy, the amount of content produced by traditional news outlets, relief organizations, and social media vastly overwhelms those trying to monitor the situation.

The ensuing digital overload that accompanies large scale disasters suggests an opportunity for automatic summarization – the implied task here is to monitor an event as it unfolds over time by processing an associated stream of documents and producing a rolling update summary containing the most salient information with respect to the event (which we also refer to as the query).

This general task is found in a variety of fields including journalism, finance, and especially crisis informatics, where there is a dire need at all times for situational awareness (i.e. what is happening now) that is largely achieved manually [Starbird and Palen, 2013]. This should be a major use case for the decades-long research on automatic multi-document summarization (MDS) systems. Such systems could deliver relevant and salient information without interruption, even when humans are unable to. Perhaps more importantly, they could help filter out unnecessary and irrelevant detail when the volume of incoming information is large.

Frustratingly, classic MDS approaches are not robust enough to handle streaming data. Their reliance on unsupervised clustering and nearest neighbors techniques leans heavily on lexical redundancy to determine the salience of a text [Erkan and Radev, 2004]. In the streaming scenario we focused on recovering novel information which is often not detected by these algorithms.

In addition, most MDS methods assume a fixed input set to which they have full retrospective access which is clearly not the case with streaming data and may not be feasible for most large web text corpora. The streaming or time component of the summarization task also brings with it the notion of timeliness – information may become stale or outdated. Managing this has not been extensively studied in the context of MDS.

In [Kedzie *et al.*, 2015] we were able to significantly reduce reliance on redundancy by explicitly predicting salience with a Gaussian process regression model. We ran experiments in a crisis informatics type scenario, where our summarization system was given an event query (e.g. "boston marathon bombing") and was expected to filter a multi-terabyte stream of online news articles while producing a brief but comprehensive extract summary about the query.

The stream was processed in hourly batches. At each hour, we predicted how salient each sentence in the current batch was. Then, summary sentence selection was performed by running affinity propagation (AP) clustering over the current batch, adding the sentence at the center of each cluster to the summary. The clustering process was biased by the salience predictions, causing the clusters to form around only the most salient inputs, increasing the likelihood that sentences nearer to the cluster center would be highly relevant to the query.

We defined the salience of a sentence (the response or target variable to be predicted by the regression) as its similarity to a gold standard summary authored by a human annotator. In practice, this similarity was computed under the semantic similarity method of [Guo and Diab, 2012].

We used a variety of features to predict salience. The simplest features included sentence length, number of query term matches, and ratio of capitalized to non-capitalized words, while more complex features included the average word probability under a pair of n-gram language models. The first language model was built from generic newswire text and was intended to identify sentences that typify this style of writing. The second model was specific to the query event type (e.g. bombing, hurricane, etc.) and was built from domain related Wikipedia documents. This model captured query specific relevance. Other features captured the distance of the event to the location discussed (implicitly or explicitly) in the sentence; salient sentences are more likely to discuss locations closer to the event query.

This model, which we referred to as APSALIENCE outperformed several clustering baselines, especially toward the beginning of the stream where, as we hypothesized, clustering approaches were not able to effectively exploit redundancy. This model achieved the best results not only in our evaluation but was a top system in an independent evaluation at the 2014 Text Retrieval Conference [Aslam *et al.*, 2015].

Unfortunately, the clustering component forces the APSALIENCE model to process the stream in hourly buckets which negatively affects the timeliness, or latency, of the summary. In our most recent work [Kedzie *et al.*, 2016], we model the streaming summarization task as a form of greedy search over possible extractive summaries. In this formulation, each search state corresponds to a decision to add the

current stream sentence to the summary or to skip it.

We train the model to make this decision using a similar feature representation to the salience component of our previous paper. However, we are able to take advantage of more expressive features in the current setup – for example, we construct several features using current state of the summary, something that was not possible in the previous model.

Training the model in this scenario is challenging however. We adapt techniques from reinforcement learning search-based learning [Chang *et al.*, 2015] to train our model to mimic the actions of a clairvoyant oracle system. Our results show an improvement of at least 28.3% over the APSALIENCE and other baseline models in summary F_1 performance and 43.8% when accounting for the timeliness of the summary content.

In our remaining work, we will focus on the problem of non-extractive summarization, commonly referred to as *abstractive* summarization. Most summarization systems, including our own discussed thus far, perform extractive summarization – the summary is generated by copying existing sentences without modification from the input documents. In abstractive summarization, the summary is written in whole or in part by an abstractive text generation algorithm.

Common approaches to abstractive text generation include sentence fusion, combining phrases taken from several sentences to form a new sentence, and sentence compression, selectively deleting less essential words and phrases to produce a shorter sentence. These methods, when applied to MDS, are generally the final part of a pipeline that involves content aggregation, selection, and ordering [Barzilay and McKeown, 2005].

Fusion and compression most often begin with an input sentence as a baseline structure, and then prune or substitute less important phrases, while preserving the overall meaning. E.g., the sentence, "A second suspect is still on the loose, the police said Friday," would convey roughly the same information with or without the reporting clause "the police said Friday." Fusion and compression are generally able to perform these kinds of operations.

In practice, non-essential phrases are determined by the term frequency or weight that is derived from the input documents. However, the absence of explicit event semantics prevents an abstractive pipeline from compressing several small and thematically similar sentences into a general gist.

In our current work, we are exploring methods for jointly modeling the aggregation, selection, and planning phases along with language generation, while simultaneously learning semantic representations that work across all four tasks.

We have begun initial experiments to predict word distributions of gold standard summaries by predicting the term frequency of individual summary terms based on their frequency in the input documents. Initial results have been encouraging and we are working to expand this model to account for correlations between words, scaling up to predicting higher order n-gram distributions.

We plan on embedding the estimated n-gram distributions in a neural network architecture for generation. We are interested in two particular architectures, neural attentional models [Rush *et al.*, 2015] and memory networks [Sukhbaatar *et al.*, 2015]. Under these models, generation can be performed using a similar greedy or approximate search as in our most recent extractive summarization system. The attentional model allows us to condition the generation on soft alignments with the input sentences (i.e. perform content selection) while the memory network can capture longer term dependencies between the inputs and previous summary content (i.e. perform content planning).

To conclude, we have developed two extractive summarization systems for streaming text data. Both systems explicitly predict the salience of input stream text to create a rolling summary. Finally, we discussed our proposed work for combining these systems with an abstractive text generation model. In future work, we would also like to apply these models to multi-modal stream summarization perhaps where news is incorporated with social media and microblog text.

References

[Aslam *et al.*, 2015] Javed Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Tetsuya Sakai. Trec 2014 temporal summarization track overview. Technical report, DTIC Document, 2015.

[Barzilay and McKeown, 2005] Regina Barzilay and Kathleen R McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005.

[Chang *et al.*, 2015] Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume, and John Langford. Learning to search better than your teacher. In *ICML*. JMLR Workshop and Conference Proceedings, 2015.

[Erkan and Radev, 2004] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *JAIR*, 22(1):457–479, 2004.

[Guo and Diab, 2012] Weiwei Guo and Mona Diab. A simple unsupervised latent semantics based approach for sentence similarity. In *Proc. of Lexical and Computational Semantics*, pages 586–590. ACL, 2012.

[Kedzie *et al.*, 2015] Chris Kedzie, Kathleen McKeown, and Fernando Diaz. Predicting salient updates for disaster summarization. In *Proc. of ACL-IJCNLP*, pages 1608–1617. ACL, July 2015.

[Kedzie *et al.*, 2016] Chris Kedzie, Fernando Diaz, and Kathleen McKeown. Real-time web scale event summarization using sequential decision making. In *IJCAI*, 2016.

[Rush *et al.*, 2015] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[Starbird and Palen, 2013] Kate Starbird and Leysia Palen. Working and sustaining the virtual disaster desk. In *Proc. of Computer supported cooperative work*, pages 491–502. ACM, 2013.

[Sukhbaatar *et al.*, 2015] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. Weakly supervised memory networks. *arXiv preprint arXiv:1503.08895*, 2015.