Summarizing Disasters Over Time

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

Kathleen McKeown
Columbia University
Department of Computer Science
kathy@cs.columbia.edu

Fernando Diaz
Microsoft Research
fdiaz@microsoft.com

ABSTRACT

We have developed a text summarization system that can generate summaries over time from web crawls on disasters. We show that our method of identifying exemplar sentences for a summary using affinity propagation clustering produces better summaries than clustering based on K-medoids as measured using Rouge on a small set of examples. A key component of our approach is the prediction of salient information using event related features based on location, temporal changes in topic, and two different language models.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Summarization

Keywords

Extractive Summarization, Affinity Propagation

1. INTRODUCTION

During crises, information is critical for first responders and those caught in the event. When the event is significant, as in the case of Hurricane Sandy, the amount of information produced by traditional news outlets, government agencies, relief organizations, and social media can vastly overwhelm those trying to monitor the situation. Methods for identifying, tracking, and summarizing events from text based input have been explored extensively (e.g., [1, 8, 27]). However, these experiments were not performed in the large and heterogeneous environment of the modern web.

In this paper, we present an update summarization system to track events across time. Our system predicts sentence salience in the context of a large-scale event, such as a disaster, and integrates these predictions into a clustering based multi-document summarization system. We train a regression model to predict sentence salience and use these predictions to bias the formation of sentence clusters around more salient regions in the input space using affinity propagation (AP) clustering. AP uses the salience predictions as well as pairwise similarities among input sentences to identify *exemplar* sentences, which we use as our summary output. Our approach differs from other methods of summarization that compute salience by pairwise comparisons alone, ignoring features of importance that are intrinsic to the sentences themselves.

WOODSTOCK '97 El Paso, Texas USA

2. RELATED WORK

A principal concern in extractive multi-document summarization is the selection of salient sentences for inclusion in summary output [20]. This has often been approached as a ranking problem. Sentences have been ranked by the average word probability, average tf-idf score, and the number of topically related words (topic-signatures in the summarization literature) [21, 13, 16]. The first two statistics are easily computable from the input sentences, while the third only requires an additional, generic background corpus. Another ranking approach, centroid summarization, involves creating an average bag of words (BOW) vector, the centroid, from the input sentences and ranking sentences by their similarity to the centroid [23]. Graph [7] and clustering [12, 18, 24] based approaches, on the other hand, make use of pair-wise similarity comparisons amongst input sentences. In these models, salient sentences are more central to the input or cluster, respectively.

Supervised learning has also been applied to this task. Model features are usually derived from human generated summaries, and are non-lexical in nature (e.g., sentence starting position, number of topic-signatures, number of unique words, word frequencies). Seminal work in this area has employed naive Bayes and logistic regression classifiers to identify sentences for summary inclusion [14, 5].

Several researchers have recognized the importance of summarization during natural disasters. Guo *et al.* developed a system for detecting novel, relevant, and comprehensive sentences immediately after a natural disaster [10]. The method uses a model of sentence relevance and novelty in order to select appropriate updates. Training data for regression targets is automatically generated from retrospective Wikipedia data. The system is evaluated on news documents related to 197 natural and human disasters from

2009 to 2011 using variants of Rouge modified to capture novelty, relevance, and comprehensiveness [15]. Wang and Li present a clustering-based approach to efficiency detect important updates during natural disasters [26]. The algorithm works by hierarchically clustering sentences online, allowing the system to output a more expressive narrative structure than Guo *et al.*. The method is evaluated on official press releases related to Hurricane Wilma in 2005 using Rouge score between the system summary and a manually generated target summary.

This work uses the TREC Stream Corpus data set that is in use by the TREC Temporal Summarization track [9, 2] . Generally, last year's participants used a pipelined approach to build summaries, generally ranking sentences, filtering out all but the most relevant, and then performing some sort of deduplication/redundancy removal step. Ranking approaches ranged from simple query word match to more sophisticated query expansion and and query based language model scoring [17, 28, 3]. Perhaps most similar to our approach, the system of [28] uses a weighted combination of features (similarity to query, named entity frequency, predicate frequency, presence of numerical values, sentence novelty, etc.) to score sentences; sentences above a threshold are added to the summary. Both the weights and the threshold are selected by hand.

Our system seeks to combine the best of these approaches, using supervised learning to predict salience rankings, and directly incorporate this information in a clustering algorithm to bias the formation of sentence clusters around highly salient regions.

3. MULTI-DOCUMENT SUMMARIZATION FRAMEWORK

A common approach to automatic summarization is to identify sentences with the highest centrality with respect to the input sentences. Intuitively, sentences with a high degree of centrality are more semantically related to the entire set of input sentences. A summary can thus be obtained by returning the k most central sentences. This generally implies the calculation of pairwise distances between all sentences [23, 7]. In these approaches, sentences are evaluated extrinsically by their distance to other sentences, either directly [7] or through an aggregate centroid object [23]. The distance between sentences is most commonly the cosine distance of sentence term-vectors, but in general this can be an arbitrary real-valued similarity function.

For some domains, it is very likely that we will have additional background knowledge that could be predictive of sentence salience for the event being summarized. For example, certain kinds of information extracted from the sentence text (e.g., temporal or geographic proximity) can indicate relevance to a given event. It would be difficult to incorporate this kind of salience into the measure of centrality.

For example, consider a cluster of three sentence vectors $s_1 = (1, 1, 0)$, $s_2 = (1, 1, 1)$, and $s_3 = (0, 1, 1)$. Without any other information, s_2 has the highest degree of centrality, i.e. it has the highest average cosine similarity and smallest average Euclidean distance to the other sentences. Now, if we believe that s_1 is α times more salient than the other sentences, we cannot simply scale s_1 by α–the average cosine similarity will remain unchanged, since the vector magnitude does not affect the angle and s_1 will have an even greater the average Euclidean distance from the rest. Worse still, s_2 will still be the most representative sentence of the three.

In our approach, the system generates clusters using an affinity propagation algorithm and from each cluster an exemplar sentence is selected that is added to the summary. In the following sections, we show how prior information representing salience can easily be incorporated into the affinity propagation algorithm. We believe the incorporation of salience to be useful in noisy environments (e.g., a web crawl), and that it can help the formation of clusters around the most relevant inputs. Our current system is trained using features derived from location, changes in wording across time and language models that characterize the language of disaster to generate summaries at regular intervals across time. As we develop the system further, we will extend it to generate updates across time, penalizing the salience of concepts already selected by the summarizer to encourage the discovery of novel sentences as the event unfolds.

3.1 Data

Our documents for summarization come from the online news portion of the TREC Stream Corpus, a 6.45tb corpus obtained by hourly web crawls from October 2011 through mid February 2013 [9].[1] Summary events come from the TREC Temporal Summarization track, and include natural disasters like Hurricane Sandy as well as man-made events like a 2012 train accident in Buenos Aires.[2] The track organizers also provide a search query for each event [2]. For each event, we collect the documents that contain all query words and stratify them by the hour they were collected.

For evaluation purposes, the track organizers also provided gold nugget information (i.e. important pieces of information, usually the length of a short clause or sentence). These gold nuggets come from the event's related Wikipedia article and also include the timestamp of when they were added to the page.

To create hourly gold summaries to evaluate our system, we simply take the set of gold nugget information from the start of the event up to the current hour.

3.2 Affinity Propagation

Affinity propagation (AP) is a message passing algorithm that identifies both exemplar data points and assignments of each point to an exemplar. This is done iteratively by passing *responsibility* and *availability* messages between data points that quantify the fitness of one data point to represent another, and the fitness of a data point to be represented based on the choices of other data points respectively [6].

AP is parameterized by a $n \times n$ similarity matrix S and a $n \times 1$ preference vector π. S is a real-valued matrix where $S(i, j)$ is the similarity of the i-th data point to the j-th data point. S does not need to be symmetric. π is a real-valued vector where $\pi(i)$ expresses our preference that the i-th data point can serve as an exemplar a priori of other data points.

In our experiments $\pi(i)$ is set to the salience prediction from the Gaussian process regression for the i-th sentence minus an offset. This offset drives most of the preferences negative and reduces the number of returned exemplars to a handful of sentences (around 4-5). For the similarity matrix, we use $S(i, j) = -\text{dist}(i, j)^2$, where dist is the Euclidean

[1] `http://trec-kba.org/kba-stream-corpus-2014.shtml`
[2] `http://trec.nist.gov/data/tempsumm2013.html`

distance between the BOW vectors for sentences i and j. The summary output is the set $\mathcal{E}$ of returned exemplars found after convergence.

AP has two useful properties for summarization. First, the number of clusters identified is determined by the preferences – lower overall preference values will result in fewer clusters. Unlike k-means, we do not have to specify how many clusters we would like to find. Determining the number of clusters in a principled way each time we run the clustering algorithm would be difficult in our setup. Secondly, the arbitrary nature of the preferences and similarity function allow us to incorporate a variety of signals for identifying the best exemplars.

3.3 Predicting Sentence Salience

In order to use AP clustering for summarization, we need to assign a preference value to each input sentence. In our approach, we equate a sentence's salience with its preference. A good model of sentence salience should predict higher values for sentences that are more likely to appear in a human generated summary of the event.

To build training data for this regression task, we take a subset of sentences relevant to the TREC events (approximately 1000) and match them to the gold nugget sentence with highest similarity as determined by the sentence similarity system of [11]. We use the real-valued similarity scores as our salience scores for the training sentences.

We want our model to be predictive across different kinds of events so we avoid lexical features. Instead, we extract a variety of features including language model scores, geographic relevance, and temporal relevance from each sentence. These features are used to fit a Gaussian process regression model that can predict the similarity of a sentence to a gold summary [22]. We use the model predicted salience of each sentence as it's preference value in the AP clustering.

3.4 Basic Features

We employ several basic features that have been used previously in supervised models to rank sentence salience [14, 5]. These include sentence length, the number of capitalized words normalized by sentence length, and the number of query words present in the sentence. Query words include the event's type (e.g., *earthquake*) and are expanded with the event type's WordNet [19] synset, hypernyms, and hyponyms. For *earthquake*, e.g., we obtain "quake," "temblor," "seism," "aftershock," etc.

3.5 Language Model Features

We use two trigram language models, trained using the SRILM toolkit [25], taking as features the average log probability (i.e. the sentence's total log probability normalized by sentence length) from each model. This first model is trained on 4 years (2005-2009) of articles from the Gigaword corpus. Specifically, we use articles from the Associated Press and the New York Times. This model is intended to assess the general writing quality (grammaticality, word usage) of an input sentence and helps us to filter out text snippets which are not sentences (e.g., web page titles). The second model is a domain specific language model. We build a corpus of Wikipedia articles for each event type, consisting of documents from a related Wikipedia category. E.g. for earthquakes, we collect pages under the category *Category:Earthquakes*. This model assigns higher probability to sentences that are focused on the given domain.

3.6 Geographic Relevance Features

Locations are identified using a named entity tagger. For each location in a sentence, we obtain its latitude and longitude using the Google Maps API. We then compute its distance to that of the event location. It is possible for a sentence and an event to have multiple locations so we take as features the minimum, maximum, and average distance of all sentence-event location pairs. Distances are calculated using the Vincenty distance.

3.7 Temporal Relevance Features

Our data consists of hourly crawls of online content and so we exploit the temporality of corpus by capturing the burstiness of a sentence, i.e. the change in word frequency from one hour to the next."Bursty" sentences often indicate new and important data.

Let D_t be the set of web pages at time t and let $s = \{w_1, \ldots, w_n\}$ be a sentence from a page $d \in D_t$. We calculate the 1-hour burstiness of sentence s from document d at hour t as

$$\mathrm{b}_1(s,d,t) = \frac{1}{|s|}\sum_{w\in s}\left(\text{tf-idf}_t(w,d) - \frac{\sum_{d'\in D_{t-1}:w\in d'}\text{tf-idf}_{t-1}(w,d')}{|\{d'\in D_{t-1}: w\in d'\}|}\right)$$

where

$$\text{tf-idf}_t(w,d) = \log\left(1+\sum_{w'\in d}1\{w=w'\}\right) \times \log\left(\frac{|D_t|}{1+\sum_{d'\in D_t}1\{w\in d'\}}\right).$$

We similarly find the sentence's 5-hour burstiness. In addition to burstiness, we also include the sentence's average tf-idf and hours since the event in question started as features.

4. EXPERIMENTS

We carried out a small set of initial experiments on one event. We collected a subset of pages from the TREC Stream Corpus that were relevant to a 2012 earthquake off the coast of Guatemala, and further subdivided this collection by the hour they were created. For each hour we generated a summary using the AP clustering algorithm.

We also generated baseline summaries using the k-medoids (using the Partitioning Around Medoids algorithm), setting $k = |\mathcal{E}|$, i.e. the number of exemplar sentences returned by the AP. Because k-medoids begins with a random initialization, we took the best (minimum average distance) result of 100 restarts.

Table 1 shows example output of the AP and k-medoids generated summaries. Sentences are ranked by preference score, although preference has no effect on the k-medoids algorithm. Quantitatively, AP exemplar sentences had higher predicted sentence quality scores (preferences) than the cluster medoids. Qualitatively, the AP method appears to select more general details about the earthquake. Looking at the third sentence selection in table 1, we can see that k-medoids

Preference	AP Clustering	Preference	k-Medoids Clustering
9.010	The magnitude-7.5 quake, about 20 miles deep, was centered off the town of Champerico .People fled buildings in Guatemala City , in Mexico City and in the capital of the Mexican state of Chiapas , across the border from Guatemala	9.010	The magnitude-7.5 quake, about 20 miles deep, was centered off the town of Champerico .People fled buildings in Guatemala City , in Mexico City and in the capital of the Mexican state of Chiapas , across the border from Guatemala
9.010	A reporter in the town of San Marcos , about 80 miles north of the epicenter, told local radio station Emisoras Unidas that houses had collapsed onto residents and smashed televisions and other appliances had been scattered into the streets.	3.007	"Things fell in my kitchen ." Perez said more than 2,000 soldiers were deployed from a base in San Marcos to help with disaster relief.
9.007	The local fire department said on its Twitter account that a school had collapsed and eight injured people had been taken to a nearby hospital.	3.007	Ingrid Lopez , who went to the hospital with a 72-year-old aunt whose legs was crushed by a falling wall, said she had waited hours for an X-ray.
7.008	There are three confirmed dead and many missing after the strongest earthquake to hit Guatemala since a deadly 1976 quake that killed 23,000.	1.007	Hundreds of people crammed into the hallways of the small town hospital waiting for medical staff to help out hundreds of injured family members, some complaining they were not getting care quickly enough.

Table 1: Example summary using affinity propagation (left) and k-medoids (right)

Method	ROUGE-1			ROUGE-2			ROUGE-3		
	Recall	Prec.	F-1	Recall	Prec.	F-1	Recall	Prec.	F-1
k-medoids	0.127	0.414	0.181	0.025	0.076	0.035	0.003	0.010	0.005
AP	0.117	0.440	0.173	0.022	0.082	0.033	0.004	0.015	0.006

Table 2: ROUGE scores for k-medoids and affinity propagation methods

selects a personal experience that was reported. This is perhaps less newsworthy or reportable compared to the third sentence in in the AP generated summary which reports a notable structure collapse and injuries related to the quake. We believe AP results in a more readable and informative summary, although we have yet to perform a rigorous human evaluation of the summary output.

We evaluated both algorithms with the ROUGE toolkit [15]. N-ROUGE works by calculating the n-gram recall and precision of an automatically generated summary in reference to a model summary. We created model summaries by taking the gold nugget sentences with timestamps up to and including the current system time as the gold summary for that hour.

Table 2 shows average recall, precision, and F-measure for various orders of ROUGE score. AP demonstrated consistently higher precision than our baseline. While not statistically significant, it is difficult to show significance with Rouge using a small test; we hope further tests will confirm this improvement. On average, the AP summaries were slightly shorter than the baseline, which would partially explain this difference. It is also possible that our language models are biased toward shorter sentences; we are more likely to have seen a shorter sentence in the language model input. We are currently adapting our summarizer to add updates over time, and maintaining precision will be important to prevent topic drift.

5. CONCLUSIONS AND FUTURE WORK

We have developed a summarizer that can generate summaries over time from web crawls on disasters. We show that our method of identifying exemplar sentences for a summary using AP clustering produces summaries with higher precision compared to those based on clustering with K-medoids. A key component of our approach is the prediction of salient information using features based on location, temporal changes in topic, and two different language models.

Currently, we run each hour of summarization independently. In order to avoid repeating information, we would like to incorporate previously chosen exemplars in the preference computation. One possibility would be to down-weight a candidate exemplar's preference based on its similarity to previous exemplars.

Secondly, we would like to do more intelligent inference of missing geographical information since not all sentences contain locations. Currently we are using mean values for missing data.

Finally, we would like to experiment with non-symmetric similarity matrices, specifically using narrative chains[4]. Under this model $S(i, j)$ would express the likelihood that the events in sentence j precede the events in sentence i. We hope such a parameterization would promote more causally motivated sentences into exemplar positions, which would better describe the disaster event domain.

6. REFERENCES

[1] J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study final report. 1998.

[2] J. Aslam, M. Ekstrand-Abueg, V. Pavlu, F. Diaz, and T. Sakai. Trec 2013 temporal summarization. In *Proceedings of the 22nd Text Retrieval Conference*

(TREC), November, 2013.

[3] G. Baruah, R. Guttikonda, and O. Vechtomova. University of waterloo at the trec 2013 temporal summarization track. In *Proceedings of the 22nd Text Retrieval Conference (TREC), November*, 2013.

[4] N. Chambers and D. Jurafsky. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 602–610. Association for Computational Linguistics, 2009.

[5] J. M. Conroy, J. D. Schlesinger, P. Dianne, M. E. Okurowski, et al. Using {HMM} and logistic regression to generate extract summaries for {DUC}. 2001.

[6] D. Dueck and B. J. Frey. Non-metric affinity propagation for unsupervised image categorization. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[7] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.(JAIR)*, 22(1):457–479, 2004.

[8] E. Filatova and V. Hatzivassiloglou. Event-based extractive summarization. In *ACL Workshop on Summarization, Barcelona, Spain*, 2004.

[9] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Ré, and I. Soboroff. Building an entity-centric stream filtering test collection for trec 2012. Technical report, DTIC Document, 2012.

[10] Q. Guo, F. Diaz, and E. Yom-Tov. Updating users about time critical events. In P. Serdyukov, P. Braslavski, S. Kuznetsov, J. Kamps, S. Rüger, E. Agichtein, I. Segalovich, and E. Yilmaz, editors, *Advances in Information Retrieval*, volume 7814 of *Lecture Notes in Computer Science*, pages 483–494. Springer Berlin Heidelberg, 2013.

[11] W. Guo and M. Diab. A simple unsupervised latent semantics based approach for sentence similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 586–590. Association for Computational Linguistics, 2012.

[12] V. Hatzivassiloglou, J. L. Klavans, M. L. Holcombe, R. Barzilay, M.-Y. Kan, and K. McKeown. Simfinder: A flexible clustering tool for summarization. Proceedings of the NAACL Workshop on Automatic Summarizatio, 2001.

[13] E. Hovy and C.-Y. Lin. Automated text summarization and the summarist system. In *Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998*, pages 197–214. Association for Computational Linguistics, 1998.

[14] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73. ACM, 1995.

[15] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, 2004.

[16] C.-Y. Lin and E. Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 495–501. Association for Computational Linguistics, 2000.

[17] Q. Liu, Y. Liu, D. Wu, and X. Cheng. Ictnet at temporal summarization track trec 2013. In *Proceedings of the 22nd Text Retrieval Conference (TREC), November*, 2013.

[18] K. McKeown, J. Klavans, V. Hatzivassiloglou, R. Barzilay, and E. Eskin. Towards multidocument summarization by reformulation: Progress and prospects. In *AAAI/IAAI*, pages 453–460, 1999.

[19] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[20] A. Nenkova and K. McKeown. A survey of text summarization techniques. In *Mining Text Data*, pages 43–76. Springer, 2012.

[21] A. Nenkova and L. Vanderwende. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*, 2005.

[22] D. Preotiuc-Pietro and T. Cohn. A temporal model of text periodicities using gaussian processes. In *EMNLP*, pages 977–988, 2013.

[23] D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[24] A. Siddharthan, A. Nenkova, and K. McKeown. Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of the 20th international conference on Computational Linguistics*, page 896. Association for Computational Linguistics, 2004.

[25] A. Stolcke et al. Srilm-an extensible language modeling toolkit. In *INTERSPEECH*, 2002.

[26] D. Wang and T. Li. Document update summarization using incremental hierarchical clustering. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 279–288, New York, NY, USA, 2010. ACM.

[27] W. Y. Wang, K. Thadani, and K. McKeown. Identifying event descriptions using co-training with online news summaries. In *proceedings of IJCNLP*, Chiang-Mai, Thailand, Nov 2011.

[28] T. Xu, P. McNamee, and D. W. Oard. Hltcoe submission at trec 2013: Temporal summarization. In *Proceedings of the 22nd Text Retrieval Conference (TREC), November*, 2013.