Qlusty: Quick and Dirty Generation of Event Videos from Written Media Coverage

Alberto Barrón-Cedeño, Giovanni Da San Martino, Yifan Zhang, Ahmed Ali, and Fahim Dalvi
{albarron, gmartino, yzhang, amali, faimaduddin}@hbku.edu.qa

Qatar Computing Research Institute
HBKU Research Complex, Doha, Qatar

Abstract

Qlusty generates videos describing the coverage of the same event by different news outlets automatically. Throughout four modules it identifies events, de-duplicates notes, ranks according to coverage, and queries for images to generate an overview video. In this manuscript we present our preliminary models, including quantitative evaluations of the former two and a qualitative analysis of the latter two. The results show the potential for achieving our main aim: contributing in breaking the information bubble, so common in the current news landscape.

1 Introduction

Event reporting in digital media spans from the re-use of contents from news agencies to the direct coverage and shaping of a story. The point of view, aspects, and storytelling of the same and related events can be diverse from medium to medium, depending on their editorial line (e.g., left vs right), target audience (e.g., quality vs tabloid), house style, or mere interest in an event. Qlusty aims at presenting consumers with a short video overview of the facts with contrasting coverage of the same news event by different news outlets, the overall aim being to break the information bubble.

Our video-production architecture consists of four modules: event identification, de-duplication, coverage diversification, and image gathering. Such modules can be translated into IR and NLP problems: document clustering, near-duplicate identification, ranking, and query generation. We present a quantitative analysis of the clustering and de-duplication modules, taking advantage of the METER corpus for text re-use analysis. The clustering strategy we use — DBSCAN— outperforms k-means even if in the former one no information about the number of clusters is known in advance: F_1 values in the range of 0.71 vs. 0.60. A qualitative analysis, carried out on the News Corpus G and Signalmedia 1M corpora, shows the potential of our diversification and query generation modules in the generation of attractive videos.

In: D. Albakour, D. Corney, J. Gonzalo, M. Martinez, B. Poblete, A. Vlachos (eds.): Proceedings of the NewsIR'18 Workshop at ECIR, Grenoble, France, 26-March-2018, published at http://ceur-ws.org

2 News Corpora

We use three corpora to tune and test our models both quantitatively and qualitatively.

METER Corpus [CGP02]. It includes documents covering events as published by one news agency and nine newspapers from the British press. This characteristic allows for the tuning of models for event identification. Each newspaper document can be wholly-, partially-, or non-derived out of a news agency report. Therefore METER is useful to test de-duplication models. It is relatively small: $1.7k$ documents. Still it is manually annotated by expert journalists. Twenty-five percent of the newspaper notes are wholly-derived from an agency wire. Either derived or not, in general the notes are modified to stick to editorial focus, style, and readability standards.

News Corpus G [Gas17]. It was originally intended for the development of news recommendation models. G does not contain full articles; only titles. The article's content can be downloaded from the provided URL, pointing to the original publisher. We stick to use only the titles to assess the robustness of our models when dealing with very short texts. G is significantly larger: $423k$ documents covering $7,231$ events. Such events are as provided by Google News and we

do not consider them as ground-truth.

SignalMedia 1M Corpus [CAMM16]. This corpus is significantly more diverse. Beside including documents from major news agencies and papers, 1M contains material from magazines and blog entries, among others. We discard blog entries and focus on items identified as news. This dataset is particularly challenging because it is only lightly curated; it may contain noisy text (e.g., with HTML tags) and even content-less entries.[1] Due to the regular querying of articles, verbatim duplicates also exist in the collection (i.e. the same article may exist a number of times with a different unique id). As stressed in [CAMM16], this real-life dataset prevents from the over-estimation of performance usually obtained on clean data. 1M does not include any event-related information.

3 Architecture and Models

Our architecture consists of four modules plus the video-generation stage, which are described next.

3.1 Clustering for Event Identification

The input to this module is a batch of news articles from a fixed time period. The output is the articles organised within a non-specified number of events. Traditionally, for this task the input data is treated as a continuous stream of documents. Hierarchical [SCK+06] and partitional clustering [AS12, AY10] are popular approaches. Still we use DBSCAN [EKSX96]. The main reasons are that —at this stage— we are not interested in news streams but in temporal batches and, perhaps more important, DBSCAN does not require information related to the expected number of events. As a result, no knowledge is necessary about the distribution of the input documents.

DBSCAN does require to set two hyper-parameters. The first one is the maximum distance under which two elements can be considered as part of the same cluster. The second one is the minimum number of elements in a cluster. Items can belong to no neighbourhood at all, and be considered as noisy entries. We fix the minimum size of a cluster to 2 news articles, thus considering singletons as noise. As for the maximum distance, we use the METER corpus to empirically set it. The experiments are described in Section 4.1.

We opt for *doc2vec* embeddings [LM14] for document representation, pre-trained on articles from the Associated Press [LB16]. The pair-wise distances are computed using 1 minus cosine similarity. The use of *doc2vec* for representing documents looks appealing due to its semantic properties.

[1]For instance, the content of entry f4edd16d-df59-41f9-ae01-d4dee076b0d5 is "Your access to this site has been temporarily blocked. This block will be automatically removed shortly".

3.2 Near-Duplicate Detection for De-Duplication

The input of this module is the articles belonging to a single event, as identified by the clustering module. The output is such articles after discarding near-duplicates. We opt for standard text re-use identification approaches based on word n-grams comparison [LBM04]. We represent the texts as bags of word n-grams after standard pre-processing: casefolding, tokenisation, and stopword removal. Tokens shorter than 2 characters are discarded as well. We use the Jaccard coefficient [Jac01] to compute the similarities.

The value for n as well as the threshold to consider that two documents are near-duplicates are set empirically, once again on the METER corpus. The experiments are discussed in Section 4.2

3.3 Ranking for Diversification

The input of this module is the de-duplicated articles from a specific event as filtered by the de-duplication module. The output is a ranked list of the documents. One of the premises of our system is breaking a user's bubble. We aim at presenting a news event including points of view as diverse as possible. The idea is that those articles which are most dissimilar to the rest covering the event are those which contain the most diverse contents.

In a k-means-like model finding such dissimilarity would be as straight-forward as computing the similarity of each article against the centroid. Nevertheless, no centroid exist in a DBSCAN-generated cluster. Therefore, our ranking function consists of computing the average similarity between an article and the rest of articles in the cluster:

$$score(d) = \frac{-1}{|c|} \sum_{d' \in c | d' \neq d} sim(d, d') \qquad (1)$$

where d (d') is a document in cluster c and $|c|$ represents the size of c. Once again, we use cosine similarity on *doc2vec* representations. The articles will be presented to the user according to this ranking, from top to bottom. We use -1 because we want the most different articles to appear first. The opening article is an exception: it is the last one according to the scoring function (i.e. the most similar to the rest of the cluster members). The reason is that we consider this article is the best one to open the video and give a good overview of the event. This module requires no tuning. Section 4.3 shows a qualitative analysis.

3.4 Query Generation for Image Gathering

Finally, we query a search engine to gather illustrations for each of the articles. The input to this module is

the ranked list of texts from the diversification module and the output is one query per article.

We explore three alternatives to generate the query. Model q_1 uses the news title. Models q_2 and q_3 follow a common mechanism. Firstly, all sub-chunks for all texts are extracted and tf-idf-ranked. For each document in the list, that chunk with the highest score is selected as the query. Once a chunk has been used, it is discarded from the list of candidates to avoid grabbing duplicate images. For model q_2 we use word 2-grams, whereas for q_3 we use named entities (NE). Regardless of the contents in the first article in the ranking, its query consists of the top NE.

The so-generated chunks are queried to a search engine, one at a time, and the top-5 pictures grabbed for integration in the video. In this version we use Google's search engine.

4 Experiments and Results

4.1 Clustering Tuning

Our first experiment intends to tune our event identification model (cf. Section 3.1). Our objective is identifying the best DBSCAN neighbourhood maximum distance (*eps*) for a random number of events and their associated articles. We are interested in two factors: high quality and stability for different document volumes.

First we formalise the problem and describe the performance measures. Let D be a collection of documents covering a set of events E. We refer to the number of events in E as $|E|$. For each $d \in D$, let $e(d)$ be the set of documents belonging to the same event as d. Analogously, let $c(d)$ be the set of documents that the model assigns to the same cluster as d's. For any $E' \subseteq E$, let $D|_{E'}$ be the subset of D whose documents' events are in E'. We use BCubed-F_1 as clustering performance measure [AGAV09]. We define $\delta(s_1, s_2) = 1$ if the sets s_1 and s_2 are identical, 0 otherwise. Let

$$\mathrm{TP}(d) = \sum_{d'} \delta(e(d), e(d')) \cdot \delta(c(d), c(d')) \qquad (2)$$

be a function counting the number of documents belonging to the same event as d which have been put together in the same cluster by DBSCAN. BCubed-F_1 is the harmonic mean between BCubed Precision P and BCubed Recall R:

$$P = \frac{1}{|D|} \sum_{d \in D} \frac{\mathrm{TP}(d)}{|c(d)|}, \quad R = \frac{1}{|D|} \sum_{d \in D} \frac{\mathrm{TP}(d)}{|e(d)|} \qquad (3)$$

We estimate parameter *eps* as follows. For $10 \leq i \leq |c|$: we randomly select $c', |c'| = i$ events and run our clustering algorithm on $D|_{c'}$. We perform 10 random repetitions to assess the stability of the outcome on increasing numbers of gold clusters. Figure 1 shows, for each *eps* value, the BCubed-F_1 measure averaged over the 10 runs together with standard deviation. We include the performance of k-means to give perspective to the results. In principle, k-means has the advantage of including the expected number of clusters as a parameter and we always assign the right number.

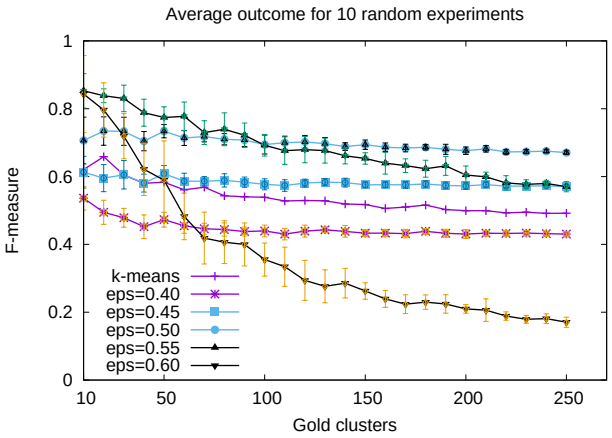

Figure 1: Evolution of BCubed F_1-measure when dealing with increasing numbers of events and documents.

Values of *eps*≥ 0.55 yield the best results when dealing with small numbers of clusters, but drop drastically when facing larger numbers of events. Lower values yield a relatively stable performance, regardless of the number of events in the dataset. An analysis focused on BCubed precision and recall values (not reported) indicate that the drop observed for *eps*≥ 0.55 is pulled by precision; the clusters tend to be larger than they should, including noisier entries. As a compromise between stability and purity, we select *eps*$= 0.55$.

4.2 Near-Duplicate Identification Tuning

Our second experiment intends to tune the model for near-duplicate identification (cf. Section 3.2). The purpose is tuning two parameters: the value of n — the word n-gram level— and the similarity threshold upon which documents are considered near-duplicates and hence one can be discarded from the final output. In the sibling task of text re-use detection, setting n to $\{2, 3\}$ [BCR09] and even 5 [KBK09] is considered standard. As we are interested in discarding whole documents to reduce redundancy to a minimum, we explore low values to allow for a more flexible comparison: $n = \{1, 2\}$.

Once again we use the METER corpus and its text re-use annotation. We adopt two settings. In the simple setting, we consider a pair of documents news agency–newspaper as positive iff the latter is labelled as wholly-derived and both cover the same event. In the complex setting we consider an additional triangu-

Table 1: Evolution of F_1 for different similarity thresholds τ in the near-duplicate identification task.

	simple		complex	
τ	$n=1$	$n=2$	$n=1$	$n=2$
0.10	0.397	0.645	0.422	0.679
0.15	0.581	0.497	0.621	0.518
0.20	0.720	0.373	0.758	0.381
0.25	**0.752**	0.274	**0.787**	0.268
0.30	0.723	0.205	0.755	0.192
0.35	0.657	0.147	0.686	0.133
0.40	0.575	0.107	0.599	0.096
0.45	0.496	0.072	0.511	0.064
0.50	0.422	0.049	0.429	0.043

lar relationship: a pair newspaper–newspaper is considered as positive iff both are labelled as wholly-derived from the same news agency article. We restrain our similarity comparison to all those articles published in the same day, resulting in $38k$ and $48k$ comparisons in the simple and complex settings, respectively. As a consequence of this volume of comparisons a high imbalance in the dataset exists —most pairs are negative instances. We evaluate this experiment on the basis of the F_1-measure for binary classification: $F_1 = \frac{2 \cdot tp}{2 \cdot tp + fp + fn}$, where tp, fp, and fn stand for number of true positives, false positives, and false negatives. Table 1 shows the results. Firstly, a more flexible comparison based on word 1-grams results in the best performance (this may imply documents which are not complete duplicates are discarded; we prefer this over including very similar notes). In both simple and complex settings the best F_1 is obtained with $\tau = 0.25$ and we select this threshold. This supports the concept of co-derivative and reflects that the threshold is valid for both news agency–newspaper and newspaper–newspaper comparisons.

4.3 Articles Ranking

Now we make a qualitative analysis. Table 2 shows the titles of the articles of three events ranked on the basis of our diversification model (cf. Section 3.3).

Instance A tells the story of Libyan rebels and their impact on oil. The top article does summarise the event, referring to a rebel attack on naval forces. As expected, the topic of article 2 is not as close: it is about the plans to sink a ship transporting illegal oil, currently besieged by the Libyan Navy. Whereas the third article still refers to oil, rebel attacks, and even to the chances for a conflict, the latter two refer to the dismissal of the Libyan PM by the parliament. That is, we are indeed looking at a story from different angles.

Something similar occurs with instances B and C. Instance B is about the listing of a mansion. After an introductory first article, further details appear such as price or location. Instance C tells the story of the decease of a former girlfriend of actor Jim Carrey. It is worth noting article 2, about a different event. Our event detection module got confused because this article is about the girlfriend of an actor. Whether this is relevant for a user is arguable.

4.4 Query Generation

Table 2 also shows the queries as generated by the three variations of our generator: q_1, q_2, and q_3 plus a fourth variation: $q_4 = q_2 + q_3$ (cf. Section 3.4). The NE-based q_3 seems far from perfect when dealing with the titles of instances A and B. The cause is that the camel-casing is confusing the NER. The simple n-grams-based approach seems to produce sensitive queries. When having at hand the full article, the NE-based model works slightly better.

Figure 2 shows the photograms of videos generated with these four kinds of queries for Table 2 instance B. Each subfigure refers to one video and each row to one news article, which can include up to five images. The whole titles from strategy q_1 provide a good visual overview of the event: the listed house and its owners. Still, due to contents overlapping, some images appear more than once: coordinates {1,4; 2,2}, {1,5; 5,3; 7,2}, and {5,1; 7,1}. The chunk-level strategies result in less repetition. Strategy q_3 based on NEs is more varied: focusing on football player *Tom Brady* for the first two titles and moving towards the main event: the listing of a house for *sale* in *Los Angeles*, and finally the second person involved: top-model *Gisele Bündchen*. Something similar occurs with q_2's 2-grams: non-duplicated photograms centred in the couple and the listed house. Still, q_2 has a problem: "The Brady report" is an Arizona radio show and the resulting photograms refer to it. Even with this mistake in mind, it seems like q_2 provides a good balance between relevance and diversity. Combining NEs and 2-grams into q_4 reduces variation (photograms {5,3; 7,3} and {5,5; 7,1} are the same).

5 Final Remarks and Ongoing Work

We presented our first efforts on breaking the news bubble. We integrated a system for the automatic generation of videos consisting of four modules: event identification, de-duplication, diversification, and image gathering. The outcome comes in the form of short illustrated videos aiming at providing a user with different points of view in the coverage of the same event.

Departing from this architecture, we aim at using more sophisticated text representation and event identification technology. We are particularly interested in storyline generation [MSA+15, VCK15].

Table 2: Three examples of diversification-ranked news and the generated queries: $q_{1...3}$.

	News title (q_1)	2-grams (q_2)	NE (q_3)
A	**News Corpus G–11 March, 2014**		
1	Rebels Fired at Libyan Naval Forces: Minister	Libyan	Libyan
2	Brincat gives no details as Libya tanker standoff continues	brincat gives	Brincat
3	Rebel group manoeuvres over Libya's oil could lead to renewed civil conflict	rebel group	Rebel
4	URGENT - Libya PM	Libya PM	URGENT
5	Libyan parliament dismisses PM	libyan parliament	–
B	**News Corpus G–20 April, 2014**		
1	Gisele Bundchen and Tom Brady sell home for £30million	Tom Brady	Tom Brady
2	Tom Brady wants $50m - for his mega mansion in LA	brady wants	Brady
3	For Sale: Gisele's $50 Million LA Chateau	sale gisele	Sale
4	Tom and Gisele SELLING Mega-Estate in LA!!!!!	gisele selling	Tom
5	Tom and Gisele's LA home is listed at $50m	gisele home	LA
C	**SignalMedia 1M–29 September, 2015**		
1	Cathriona White, ex-girlfriend of actor Jim Carrey, committed suicide on Monday.	Cathriona White	Cathriona White
2	Twilight star Robert Pattinson says that those who send negative comments [...]	twilight star	Twilight
3	A former girlfriend of actor Jim Carrey has died of an apparent suicide, the [...]	girlfriend actor	Ed Winter
4	Jim Carrey and Cathriona White are spotted hand in hand leaving their [...]	posted photo	N.Y.
5	Jim Carrey's Irish ex-girlfriend Cathriona White has been found dead.	2015 /raw	Cathriona Cappawhite

Figure 2: Photograms of videos as generated with queries $q_{[1,..4]}$. One row corresponds to one article, for which up to 5 images are integrated. The top and left numbers are the photograms coordinates, for easy location.

References

[AGAV09] Enrique Amigo, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constrains. *Inf Retrieval*, 12(4):1–32, 2009.

[AS12] Joel Azzopardi and Christopher Staff. Incremental Clustering of News Reports. *Algorithms*, 5(4):364–378, aug 2012.

[AY10] Charu C Aggarwal and Philip S Yu. On Clustering Massive Text and Categorical Data Streams. *Knowledge and Information Systems*, pages 171–196, 2010.

[BCR09] Alberto Barrón-Cedeño and Paolo Rosso. On Automatic Plagiarism Detection based on n-grams Comparison. *Advances in Information Retrieval. Proceedings of the 31st European Conference on IR Research*, LNCS (5478):696–700, 2009. Springer-Verlag.

[CAMM16] David Corney, Dyaa Albakour, Miguel Martinez, and Samir Moussa. What do a Million News Articles Look like? In *NewsIR 2016 Recent Trends in News Information Retrieval*, pages 42–47, Padua, Italy, 2016.

[CGP02] Paul Clough, Robert Gaizauskas, and Scott Piao. Building and Annotating a Corpus for the Study of Journalistic Text Reuse. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, volume V, pages 1678–1691, Las Palmas, Spain, 2002. ELRA.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press, 1996.

[Gas17] Fabio Gasparetti. Modeling User Interests from Web Browsing Activities. *Data Mining and Knowledge Discovery*, 31(2):502–547, 2017.

[Jac01] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[KBK09] Jan Kasprzak, Michal Brandejs, and Miroslav Kriřpač. Finding Plagiarism by Evaluating Document Similarities. volume 502, pages 24–28, San Sebastian, Spain, 2009. CEUR-WS.org.

[LB16] Jey Han Lau and Timothy Baldwin. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 78–86, 2016.

[LBM04] Caroline Lyon, Ruth Barret, and James Malcolm. A Theoretical Basis to the Automated Detection of Copying Between Texts, and its Practical Implementation in the Ferret Plagiarism and Collusion Detector. In *Plagiarism: Prevention, Practice and Policies Conference*, Newcastle upon Tyne, UK, 2004.

[LM14] Quoc Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. In *Proceedings of the 31 st International Conference on Machine Learning*, Beijing, China, 2014.

[MSA+15] Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, and Ruben Urizar. Semeval-2015 task 4: Timeline: Cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 778–786. ACL, 2015.

[SCK+06] Nachiketa Sahoo, Jamie Callan, Ramayya Krishnan, George Duncan, and Rema Padman. Incremental hierarchical clustering of text documents. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 357–366, New York, NY, 2006. ACM.

[VCK15] Piek Vossen, Tommaso Caselli, and Yiota Kontzopoulou. Storylines for structuring massive streams of news. In *Proceedings of the First Workshop on Computing News Storylines*, pages 40–49, Beijing, China, 2015. ACL.