POLITECNICO DI TORINO
Repository ISTITUZIONALE

Summarization of emergency news articles driven by relevance feedback

Original
Summarization of emergency news articles driven by relevance feedback / Cagliero, Luca. - ELETTRONICO. - (2017), pp. 3713-3721. (Intervento presentato al convegno 2017 IEEE International Conference on Big Data (BigData 2017) tenutosi a Boston (MA, USA) nel 11-14 Dicembre 2017) [10.1109/BigData.2017.8258368].

Availability:
This version is available at: 11583/2708504 since: 2018-05-24T01:02:54Z

Publisher:
IEEE

Published
DOI:10.1109/BigData.2017.8258368

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

24 July 2024

Summarization of emergency news articles driven by relevance feedback

Luca Cagliero
Dipartimento di Automatica e Informatica
Politecnico di Torino
Torino, Italy
luca.cagliero@polito.it

***Abstract*—Many articles on the same news are daily published by online newspapers and by various social media. To ease news article exploration sentence-based summarization algorithms aim at automatically generating for each news a summary consisting of the most salient sentences in the original articles. However, since sentence selection is error-prone, the automatically generated summaries are still subject to manual validation by domain experts. If the validation step not only focuses on pruning less relevant content but also on enriching summaries with missing yet relevant sentences this activity may become extremely time consuming.**

The paper focuses on summarizing news articles by means of an itemset-based technique. To tune summarizer performance a relevance feedback given on sentences is exploited to drive the generation of a new, more targeted summary. The feedback indicates the pertinence of the sentences that are already in the summary. Among the words or the word combinations selected by the summarization model, those occurring in sentences with high feedback score represent concepts that may be deemed as particularly relevant. Therefore, they are exploited to drive the new sentence selection process.

The proposed approach was tested on collections of news articles reporting emergency situations. The results show the effectiveness of the proposed approach.

***Keywords*-Multi-document summarization; Text mining; Frequent itemset mining; Emergency management**

I. Introduction

News articles in electronic form are daily published by online newspapers, by Web portals of public authorities, and by various social media platforms. They report significant political/economic/social events or they cover topics that are currently matter of contention between sections of public opinion. For example, emergency situations are typically reported by most news providers [1].

Since news retrieval and exploration is time consuming, among the large number of articles related to the same news readers typically select and explore only a small subset of them. For example, they read only the articles published by most renowned newspapers. However, in this way some relevant news facets or some interesting viewpoints can be missed. To overcome this issue, readers may be interested in reading through a summary per news, which summarizes the salient content of all the related articles.

News article summarization is an established text mining problem, which entails automatically generating summaries of potentially large collections of news articles. In our context of analysis, the input articles are assumed to be homogeneous, i.e., all the articles in the collection range over the same news. Based on the type of generated summary, summarizers can be categorized as sentence-based (e.g., [2], [3], [4], [5]), if the resulting summaries consist of a subset of article sentences, or keyword-based (e.g., [6]), if they consist of a set of keywords. This paper specifically addresses the sentence-based news article summarization problem.

Sentence-based summarization algorithms commonly rely on data mining or information retrieval techniques. Depending on the type of model they generate on top of the analyzed articles, summarizers can be classified as:
(i) Clustering-based (e.g., [4], [5], [7]), if they group sentences into homogeneous clusters by means of clustering algorithms.
(ii) Graph-based (e.g., [8], [9], [10]), if they rely on graph indexing strategies (e.g., PageRank [11], HITS [12]).
(iii) Optimization-based (e.g., [13], [14], [15]), if they apply Singular Vector Decomposition, Integer Linear Programming or Submodular Function Optimization techniques.
(iv) Itemset-based (e.g., [16], [17], [18], [19], if they rely on frequent itemset and association rule mining techniques to capture most significant correlations among multiple article terms.

Although the process of sentence selection is (semi)-automatic, the summaries generated from real news articles often need to be validated by domain experts through manual inspection. Experts may either prune not relevant/redundant sentences from the summary or extend it with new content, e.g., by adding missing sentences taken from the original news articles. The latter validation step is potentially challenging and time consuming, because it requires reverting to the original articles to pick the missing sentences.

This paper investigates the use of itemset-based summarization techniques to generate summaries of news articles. Itemset-based summarizers analyze the co-occurrences between multiple document terms (two or more). Specifically, they first extract frequent itemsets, which represent recurrent combinations of terms. Then, the frequent itemsets are used to drive sentence selection. Intuitively, the more itemsets are contained in a sentence, the more likely the sentence is worth considering in the summary, because it covers the most

significant concepts hidden in the analyzed data. Itemset-based approaches are potentially more accurate than the other general-purpose ones, because they consider also correlations between multiple document terms. Thanks to this property, they have achieved performance superior to most general-purpose summarizers, not relying on advanced linguistic analyses, on benchmark news article collections [17], [19].

We propose a new news article summarizer, namely Feedback-driven News Summarizer (FeedNewsSum), which performs itemset-based news article summarization driven by relevance feedback. A compact subset of frequent itemsets, representing most significant correlations among document terms, is generated first using an entropy-based strategy [20]. Then, a preliminary summary version is generated by selecting the sentences that best represent the combinations of terms included in the itemset-based model. Finally, based on relevance feedback given on sentences, a new, more targeted version of the summary is generated. The sentence-level feedback given by domain experts has two complementary effects:
(i) it directly rates the pertinence of each sentence in the summary, and
(ii) it indirectly rewards missing sentences that have been excluded from the summary at the first round but that include the same words or itemsets (i.e., word combinations) as highly rated sentences.

In the new summarization process driven by relevance feedback redundant/not relevant sentences occurring in the former summary are excluded thanks to effect (i), while new, potentially more significant sentences are selected thanks to effect (ii).

We tested our approach on a set of online published news collections reporting recent emergency situations. To simulate relevance feedback enrichment we run the summarization algorithm multiple times by using a cross-validation strategy [21]. Then, we compared the quality and the characteristics of the generated summaries prior to and after feedback injection. On the tested collections, thanks to feedback injection the summarizer achieved, on average, a performance improvement above 10% in terms of F1-measure score by considering the main measures collected by a standard summary evaluation toolkit (ROUGE) [22].

The paper is organized as follows. Section II describes the main steps of the FeedNewsSum summarizer, while Section III experimentally evaluates its performance on emergency news collections. Finally, Section IV draws conclusions and discusses future works.

II. The Summarization Approach

Feedback-driven News Summarizer (Feedback-driven News Summarizer) is a new itemset-based news article summarizer based on relevance feedback. Figure 1 depicts the main summarizer steps, which are briefly outlined below.

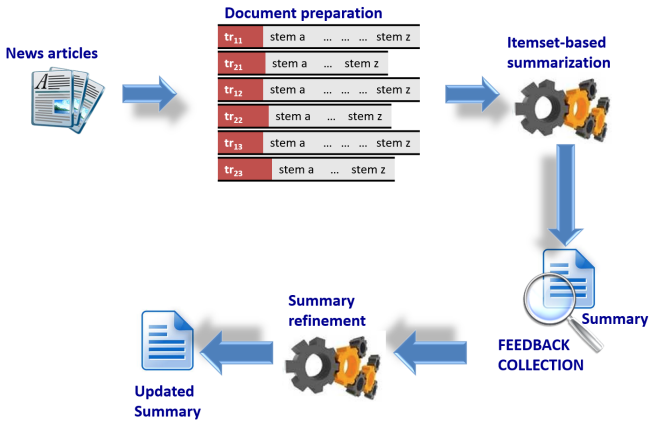

Figure 1. The FeedNewsSum summarizer

- **Document preparation.** The news articles are adapted to the next itemset mining step by applying two established text preprocessing steps, i.e., stemming and stopword elimination. Furthermore, sentences are filtered by considering their relative position in the news articles. Finally, the preprocessed news are transformed into a transactional data format (see Section II-A).
- **Entropy-based itemset mining.** A selection of the most significant itemsets is mined by using an entropy-based heuristics (see Section II-B).
- **Summary generation.** The subset of sentences containing the largest number of interesting itemsets is selected as preliminary output summary by applying an optimization strategy (see Section II-C).
- **Feedback collection and summary refinement.** Summary sentences are evaluated by domain experts or by ad hoc evaluation tools. To each sentence in the summary a feedback score is assigned by rating its relative pertinence/significance. Next, a new summary generation process, driven by the feedback scores on summary sentences, is executed and the output summary is updated (see Section II-D).

A more thorough description of each step is reported in the following sections.

A. Document preparation

This blocks aims at preparing the news articles for the next mining steps. Specifically, three established preprocessing steps are applied: (i) stopword elimination, (ii) stemming, and (iii) position-based sentence filtering. A separate description of each step is given below. Hereafter, we will denote as A=$\{a_1, \ldots, a_N\}$ an arbitrary news article collection consisting of N articles.

Stopword elimination is applied to each news article in A to filter out the words that usually have little lexical content

(e.g., article, prepositions, conjunctions). These words are likely to occur very frequently in the news collection, but their content is weakly informative for summarization purposes.

To apply stopword elimination on English-written articles, we exploited the Natural Language Toolkit (NLTK) stopword corpus [23]. However, stopwords' lists are available for most spoken languages.

Stemming is applied to each news article in D to reduce its words to their base or root form (i.e., the stem). This step is particularly useful in our context, because after stemming word frequency counts are no more biased by plural forms, past tenses, gerunds, or other word suffixes.

To analyze English-written news articles, we exploited the Snowball stemmer [24]. However, many other stemming algorithms (applicable to article written in non-English languages as well) are available in literature (e.g., the Lucene stemmer [23]).

Sentence filtering is applied to early less meaningful less relevant parts of the news articles thus improving the effectiveness and efficiency of the summarization process. Similar to existing news summarization approaches (e.g., [14], [25], [26]) the FeedNewsSum summarizer considers the sentence position in the article to perform sentence filtering. Specifically, for each news articles it considers, for subsequent analyses, only the top-q sentences of each article (where q is an input parameter provided by the analyst). As discussed in [25], in news articles top placed sentences are most likely to summarize all key concepts. Therefore, the aforesaid pruning is typically beneficial. In the contexts in which the above assumption is deemed as inappropriate, the users can disable the sentence filtering by setting very high q values.

B. Entropy-based itemset mining

Frequent itemset mining [27] is an exploratory data mining technique that focuses on discovering correlations among data items co-occurring in large datasets. An itemset I of length k, i.e., a k-itemset, is a set of k distinct items. In our context, items represent word stems. Hence, an itemset is a set of word stems (of arbitrary size) co-occurring in the sentences of the news article collection.

To perform itemset mining from news articles we adopted a transactional data representation for the preprocessed news articles. Specifically, the input news article collection is transformed into a transactional dataset (hereafter denoted as D_t) consisting of a set of transactions, where each transaction corresponds to a different sentence in the news articles and contains all the word stems occurring in the sentence. Hereafter, we will denote as s_{jk} the j-th sentence of the k-th article in the collection and as tr_{jk} the transaction corresponding to s_{jk}. tr_{jk} is the set of all non-repeated word stems (items) in s_{jk}.

An itemset I is characterized by two notable properties, i.e., tidset and support. The tidset of itemset I in the news transactional dataset D_t, denoted as tidset(I,D_t), is the set of transactions $tr_{jk} \in D_t$ for which the corresponding sentences $s_{jk} \in D$ contain all the word stems in I.

The support of itemset I in D_t is the observed frequency of occurrence of I in D_t, i.e., $sup(I)$=$\frac{|tidset(I,D_t)|}{|D_t|}$.

Since the problem of discovering all itemsets from a transactional dataset is computationally intractable [27], itemset mining is commonly driven by a minimum support threshold. The frequent itemset mining problem entails discovering all the frequent itemsets in a transactional dataset, i.e., all the itemsets whose support is above a given (analyst-provided) threshold *minsup*. However, the number of mined frequent itemsets is typically very large, because it contains a lot of redundant or potentially irrelevant patterns. To generate a more compact set of frequent itemsets representing most significant yet non-redundant knowledge hidden in the analyzed data many research efforts have been made (e.g., [28], [29], [20], [30]). Given a minimum support threshold $minsup$ and a maximum itemset model size K, we extract the top-K most interesting and non-redundant itemsets according to the entropy-based heuristics proposed in [20]. In our context, the top-K itemsets represent the most significant correlations among multiple words hidden in the news article collection. Since they are likely to cover most significant aspects of the news article collection, these itemsets will be considered as a reference model to drive sentence selection during the next summary generation step.

C. Summary generation

This step selects the sentences that are worth including in the summary based on both the itemset-based model and an established term quality index, i.e., the term frequency-inverse document frequency statistics [31], which is briefly introduced below.

The tf-idf statistics. The term frequency-inverse document frequency (tf-idf) evaluator [31] is an established term statistics that is largely used to measure how important a word stem is important in a textual document collection [32]. Tf-idf is defined as follows:

$$ti_{ik} = \frac{n_{ik}}{|a_k|} \cdot \log \frac{|A|}{|\{a_k \in A \ : \ w_i \in a_k\}|} \qquad (1)$$

where n_{ik} is the number of occurrences of the i-th stem w_i in the k-th news article a_k, D is the news collection, $|a_k|$ is the number of stems that are contained in the k-th article a_k, and $\frac{|A|}{|\{a_k \in A \ : \ w_i \in a_k\}|}$ represents the inverse document frequency of the stem w_i in the whole collection. The logarithm of the inverse document frequency is minimal when the inverse document frequency is equal to 1 (i.e., a term occurs in every article of the collection) and thus the corresponding tf-idf value reduces to zero.

The key idea behind the tf-idf statistics is that word stems appearing frequently in a few news articles (i.e., high local term frequency), but rarely in the whole collection (i.e.,

low document frequency), are the most effective ones in discriminating among sentences in a news article collection.

Since the summary should cover the largest number of news facets with the minimal number of sentences, we formalize the sentence selection task as a set covering problem.

Problem statement. Let A=$\{a_1, \ldots, a_N\}$ be the news article collection and let s_{jk} be the j-th sentence of article a_k. Let FI be the set of top-K frequent itemsets mined in the previous step (see Section II-B). Let tidset(I,D_t) be the tidset of itemset $I \in FI$ in the transactional news dataset D_t corresponding to A and let tf-idf(s_{jk},A) be the average tf-idf value of all the word stems in sentence s_{jk}. The set covering problem addressed by the summary generation step entails the selection of the subset $\mathcal{S}$ of sentences $s_{jk} \in A$ that optimizes the following multi-objective optimization problem:

$$\begin{aligned} \underset{\mathcal{F}}{\text{minimize}} \quad & F(\mathcal{S}) = [F_1(\mathcal{S}), F_2(\mathcal{S}), F_3(\mathcal{S})]^T \\ \text{subject to} \quad & \mathcal{S} \subseteq \text{A}, \end{aligned} \tag{2}$$

where

$$\begin{aligned} & F_1(\mathcal{S}) = \text{size}(\mathcal{S}) = |\mathcal{S}|, \\ & F_2(\mathcal{S}) = \text{tidset-size}(\mathcal{FI}) = -\sum_{\mathcal{I} \in \mathcal{FI}} |\text{tidset}(\text{I}, \text{D}_\text{t})|, \\ & F_3(\mathcal{S}) = \text{tf-idf}(\mathcal{S}) = -\sum_{\text{s}_\text{i} \in \mathcal{S}} \text{tf-idf}(\text{s}_\text{i}, \text{A}) \end{aligned} \tag{3}$$

and, given a precedence operator $\prec$ in order of importance, $F_1 \prec F_2 \prec F_3$ holds.

Arranging the considered objective functions in order of relative importance is an established optimization method [33], [34]. Specifically, lexicographical ordering is a technique that requires the decision-maker to establish the priority of each objective function. Then, solutions are first compared with respect to the most important one. In case of ties, the algorithm proceeds to compare the solutions but now with respect to the next most important objective. Hence, the search space for the least important objective functions is reduced.

In our context, the goal is to:

1) first, minimize the number of sentences included in the summary thus maximizing the **compactness** of the result (objective function F_1),
2) secondly, maximize the coverage of the itemset-based model thus maximizing the **significance** of the summary (objective function F_2), and
3) lastly, maximize the interestingness of the individual terms occurring in the summary thus maximizing the **attractiveness** of the summary for readers (objective function F_3).

The aforesaid optimization task aims at selecting the minimal number of sentences of the news article collection covering the maximal number of itemsets. At equal terms, the relevance of individual terms, measured in terms of average tf-idf value of the corresponding word stems, is considered.

Since set covering problems are NP-hard, we exploited a branch-and-bound Integer Linear Programming algorithm [35] to accomplish the task.

D. Feedback collection and summary refinement

This step collects a relevance feedback on the generated summary and exploits it to refine the summarization process. A relevance feedback score is assigned to each summary sentence according to its pertinence/relevance. The feedback can be either humanly generated (by domain experts) through manual inspection of the output summaries or generated by ad hoc evaluation tools (e.g., ROUGE [31]) which evaluate the similarity between the automatically generated summary and a reference model. Hereafter, we will denote as $f(s_{jk})$ the feedback score given to sentence s_{jk} of the news article collection A.

Collecting relevance feedbacks on summaries has a twofold aim. On the one hand, feedbacks on single sentences can be exploited to discard low-quality sentences accidentally included in the summary. Specifically, sentences with low feedback score are penalized during the next summarization round. On the other hand, feedback scores can be exploited to reward sentences that were not selected at the first summarization round. Specifically, feedback scores associated with summary sentences can be propagated to the occurring word stems. Sentences not in the summary can be re-considered based on the occurrences of the rewarded word stems. In this way, a sentence not in the original summary but covering similar words or word combinations (itemsets) as those covered by an highly rated summary sentence is more likely to be included in the refined summary version.

A new process of summary generation, driven by the sentence-level feedback, is executed on the original news article collection. More specifically, to consider also the feedback scores during the evaluation process objective function F_3 of the set covering problem, formulated in Section II-D, is modified as follows.

$$\begin{aligned} & F_3(\mathcal{S}) = \text{weighted tf-idf}(\mathcal{S}) = \\ & = -[(1-\alpha) \cdot \sum_{s_i \in \mathcal{S}} \text{tf-idf}(\text{s}_\text{i}, \text{D}) + \alpha \cdot \text{f}(\text{s}_\text{i})] \end{aligned} \tag{4}$$

where α is an analyst-provided parameter between 0 and 1.

Unlike the original set covering problem, objective function F_3 maximizes the attractiveness of the selected sentences according to not only the tf-idf of the corresponding terms and also to the sentence feedback scores.

In function F_3 parameter α allows experts to weigh the importance of sentence feedback score with respect to the tf-idf statistics. An analysis of the effect of this parameter on the summarizer performance is given in Section III-C.

III. Case study

We experimentally evaluated the applicability of the proposed approach in a real application scenario, i.e., the analysis of news articles related to emergency situations. Specifically, we summarized five collections of news articles related to emergency situations and we performed the following analyses:

- a comparison between the performance of the summarizer prior to and after relevance feedback injection (see Section III-A).
- a qualitative analysis of the output summaries (see Section III-B).
- a study of the impact of the main algorithm parameters (see Section III-C).

All the experiments were performed on a 3.0 GHz 64 bit Intel Xeon PC with 4 GB main memory running Ubuntu 10.04 LTS (kernel 2.6.32-31).

A short description of the analyzed news article collections is given below.

Emergency news article collections. In September 2017 we retrieved from the Web five different English-written news article collections. Each collection is associated with a different news and it is related to a specific emergency situation. Collections consist of 10 news articles each. They were crawled by providing a query, focused on a given topic, to the Google News search engine and then by selecting the 10 top ranked news articles. The queries addressed the following topics:

- *Climate change*: U.S. president Trump underestimates the effects of climate changes.
- *Hurricane Irma*: Hurricane Irma crashed through the Caribbean.
- *Hurricane Harvey*: Hurricane Harvey hits Texas (U.S.).
- *Earthquakes in southeast Spain*: An earthquake shook Murcia region (southeastern Spain).
- *Diabetes in the UK*: Diffusion, risks, and treatments of type-2 Diabetes in England.

The topics were selected as representatives of different case studies: (i) very focused news of topical interest (e.g., *Hurricane Irma, Hurricane Harvey, Earthquakes in Spain*), (ii) multi-faceted news (e.g, *Diabetes in the UK*), and (iii) broad-spectrum news and long-term matter of contention (e.g., *Climate change*).

The retrieved news articles are available at *http://dbdmg.polito.it/wordpress/research/document-summarization/*.

A. Performance comparison

This section presents the evaluation of the FeedNewsSum summarizer performance on the emergency news collections. To analyze the effect of pushing the relevance feedback into the summarization process, we compared the performance of our approach prior to and after feedback injection (see Section II). Furthermore, we compared FeedNewsSum performance with that of

(i) a recently proposed summarizer relying on word association discovery, i.e., Association Mixture Text Summarization (AMTS) [36], and

(ii) three widely used open source text summarizers, i.e., the ILP-based ICSI multi-document summarization system (ICSIsumm) [14], [37], the Open Text Summarizer (OTS) [38], and TexLexAn [39].

For ISCI, OTS, and TexLexAn we exploited the implementations provided by the authors. Since the source code of the AMTS summarizer was not publicly available on the Web we re-implemented the summarizer to the best of our understanding based on the indications given in the reference article [36].

To compare the performance of the FeedNewsSum summarizer with that of the other approaches on the emergency news article collections we used the ROUGE toolkit [31], which has been adopted as official evaluation tool for various summarization contests (e.g., the Document Understanding Conferences [22]).[1]. The ROUGE toolkit measures the quality of a summary by counting, by means of different metrics, the unit overlaps between the candidate summary and a set of reference summaries. The summarizer that achieves the highest scores can be considered the most effective. To perform a fair comparison, before using the evaluation tool, the generated summaries have been normalized by truncating each of them at 665 bytes (rounding the number down in case of straddled words), following the same approach used in the DUC competitions [22]. Several automatic evaluation scores are implemented in ROUGE. For the sake of brevity, we only report the results for ROUGE-2 and ROUGE-SU4, which are considered as the most representative scores [31].

Since reference summaries are not available for the crawled news, to evaluate the summarization performance we adopted, as previously done in [40], a leave-one-out cross validation [21]. More specifically, for each collection we summarized nine out of ten news articles and we compared the achieved summary with the remaining (not yet considered) one, which was selected as reference summary. Next, we tested all the other possible combinations by varying the reference summary and for each summarizer we computed the average performance results, in terms of precision (P), Recall (R), and F1-measure (F1), for both the ROUGE-2 and ROUGE-SU4 evaluation scores. Since within each collection we cope with articles ranging over the same news, at each iteration a news article is considered as a representative summary of all the other ones.

To generate feedback scores on summary sentences we first computed the tf-idf values of single word stems in the reference summary. Then, for each sentence in the auto-

[1]We used the command: ROUGE-1.5.5.pl -e data -x -m -2 4 -u -c 95 -r 1000 -n 4 -f A -p 0.5 -t 0 -d -a

matically generated summary we averaged the previously computed scores over all the word stems co-occurring in both summaries. To simulate the presence of bias in the process of feedback score assignment, we randomized the distribution of word stem scores by injecting a 30% random noise. To this aim, we exploited the AddNoise function provided by the RapidMiner tool [41].

Tables I reports the average results achieved by FeedNewsSum (prior to and after relevance feedback injection) as well as by all the other summarizers (i.e., ICSIsumm, AMTS, OTS, and TexLexAn) on the emergency news article collections. For all the considered datasets and measures the statistical significance of the performance difference between FeedNewsSum (with relevance feedback) and the other approaches was evaluated by the paired t-test [42] at 95% significance level. Statistically relevant differences are starred in Tables I. For each considered collection and measure, the results that were achieved by the most effective summarizers are written in boldface.

The FeedNewsSum summarizer performed better than all the other summarization algorithms for all the considered evaluation measures. The performance improvements are statistically significant in terms of ROUGE-2 and ROUGE-4 F1-measure against all of the tested competitors.

Pushing of relevance feedback into the summarization process relevantly improved the quality of the produced summaries. The improvement was above 10% on both ROUGE-2 and ROUGE-4 F1-measures. The sentence selection process appeared to be significantly more precise, because non-pertinent sentences were discarded thanks to relevance feedback injection. Furthermore, the recall measure has slightly improved thanks to the selection of new sentences which have not been selected in the first summarization round.

B. Summary examples

Table II reports some examples of summaries generated by FeedNewsSum from one representative news article collection, i.e., *Hurricane Harvey*. Specifically, it reports both the top 3 sentences of the summaries generated prior to feedback score enrichment and after (i.e., by considering relevance feedbacks).

The top sentence is the same for both summaries. It reports the number of deaths and the geographical location where the intense flooding has made serious damages. Conversely, the other sentences change. More specifically, the second sentence of the summary generated without feedback mentions the use of shelters in rescue, whereas the corresponding sentence in the refined version explained why shelter are still in use. Furthermore, the third sentence of the original summary is, to a certain extent, a repetition of the first one, as it just recalls the number of deaths by comparing the event with similar emergency situations.

C. Parameter analysis

The setting of the algorithm parameters may relevantly affect the quality and the characteristics of the generated summaries. Hereafter, we will analyze the impact of the main FeedNewsSum parameters on summarization performance and we will indicate the most appropriate configuration settings based on our experiments. In general, the most appropriate parameter settings to use depend on the characteristics of the analyzed data.

Influence weight α of the feedback score. It weights the importance of the feedback score in the computation of sentence attractiveness (see Section [42]). The higher α value, the more important the sentence relevance feedback with respect to the tf-idf score computed on the original articles.

Figure 2 shows the FeedNewsSum summarizer performance (in terms of ROUGE-2 F1-measure) by varying the value of α between zero and one. By setting α between 0.5 and 0.8 the feedback scores positively affect summarizer performance. Setting low α values yields not significant performance variations with respect to the original summary (F1-measure = 0.0299), while too high α values tend to penalize too much word stems not occurring in the highly rated sentences.

Minimum support threshold. It indicates the least observed frequency for all the mined itemsets. Setting very high support values (e.g., above 2%) yields very general and weakly informative itemset-based models. Thus, the quality of the generated summaries significantly degrades. On the other hand, setting very low support thresholds (e.g., below 0.5%) may result in a very detailed itemset-based model, which may over-fit the analyzed news article collection. On the analyzed collections,the best trade-off between model generality and quality was achieved by setting $minsup$ values between 0.5% and 1%.

Number K of most relevant itemsets. It indicates how many frequent itemsets are kept by the entropy-based heuristics. To achieve fairly good summarization performance, at least 5 itemsets must be kept for all the analyzed collections. These number may grow while coping with more complex data distributions, because a larger number of interesting word combinations may be extracted. By setting a too high K value, some redundant itemsets may be extracted. Therefore, the quality of the output summaries may get worse. We recommend to set K between 5 and 10 on similar news collections.

Number q of selected sentences per article. It indicates the number of sentences per article considered during the summarization process. In the analyzed news articles, the top 10 sentences contain, in most cases, the most salient information about the news without discarding potentially interesting knowledge. While coping with other textual documents other than news articles, this option can be disabled

Table I
EMERGENCY NEWS ARTICLE COLLECTIONS. EVALUATION BASED ON ROUGE [31]. STATISTICALLY RELEVANT DIFFERENCES IN THE COMPARISON BETWEEN FEEDNEWSSUM WITH RELEVANCE FEEDBACK (*minsup*=0.7%, K=7, Q=10) AND THE OTHER APPROACHES ARE STARRED.

Summarizer	ROUGE-2			ROUGE-SU4		
	R	Pr	F1	R	Pr	F1
FeedNewsSum with relevance feedback	**0.0219**	**0.1137**	**0.0366**	**0.0368**	**0.1816**	**0.0610**
FeedNewsSum without relevance feedback	0.0192	0.1001*	0.0299*	0.0331	0.1710*	0.0521*
ICSIsumm	0.0179*	0.0955*	0.0289*	0.0298*	0.1586*	0.0478*
AMTS	0.0146	0.0744*	0.0232	0.0309	0.1536	0.0488
OTS	0.0143*	0.0725*	0.0224*	0.0272*	0.1455*	0.0434*
TexLexAn	0.0134*	0.0709*	0.0212*	0.0289*	0.1505*	0.0459*

Method	Summary (top-3 sentences)
FeedNewsSum summarizer without relevance feedback	**(1st)** Texas officials said Thursday that they believe at least 82 people died as a result of Hurricane Harvey and the intense flooding it brought to Houston and coastal areas, although it could take weeks to determine the exact death toll. **(2nd)** The Red Cross is preparing to move those evacuees to a different shelter 11 miles away at the Northwest Mall, a largely abandoned shopping center. **(3rd)** At least 75 deaths have been reported from Harvey more than the combined death toll in the Caribbean and United States from Hurricane Irma, as of Thursday evening
FeedNewsSum summarizer with relevance feedback	**(1st)** Texas officials said Thursday that they believe at least 82 people died as a result of Hurricane Harvey and the intense flooding it brought to Houston and coastal areas, although it could take weeks to determine the exact death toll. **(2nd)** Abbott said there were about 5,250 people still living in shelters and the state was working with federal authorities to rebuild homes and businesses. **(3rd)** Hurricane Harvey could have made a dent in industrial production due to the shutdown of refiners in its path and the reduced utility use in storm-ravaged areas.

Table II
SUMMARY EXAMPLES. HURRICANE HARVEY COLLECTION. FEEDNEWSSUM CONFIGURATION SETTINGS: *minsup*=0.8%, K=7, Q=10, α=0.7

in case position-based sentence filtering is deemed as not appropriate.

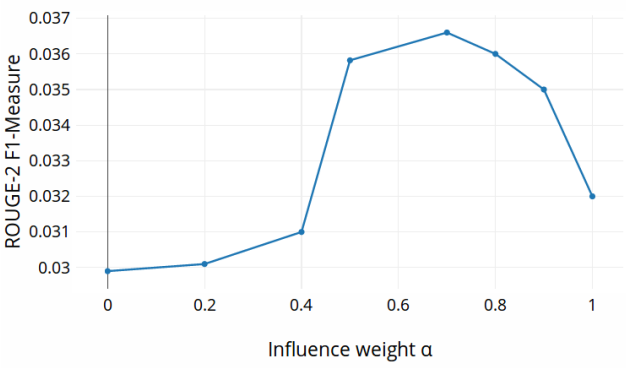

Figure 2. Impact of parameter α on FeedNewsSum performance

IV. CONCLUSIONS AND FUTURE WORKS

This paper presents an itemset-based approach to summarizing news article collections. The proposed approach takes into account relevance feedbacks given on intermediate summary results to refine the summarization process. Feedback scores on summary sentences are exploited to prune non-relevant sentences accidentally included in the summary or to reward significant sentences not selected at the first summarization stage.

We experimentally validated the effectiveness of the proposed approach on real news article collections related to emergency situations. The results show that pushing the collected feedback relevantly improves the quality of the generated summary with respect to its original version.

Future works will investigate
(i) the use of feedback scores at the itemset level, to improve the quality of the itemset-based model, and
(ii) the development of a self-learning summarization approach, which integrates statistics-based itemset evaluation measures to automatically assess summary quality thus refining the summarization process with limited human intervention.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No 700256 ("I-REACT" project).

REFERENCES

[1] D. Yates and S. Paquette, "Emergency knowledge management and social media technologies: A case study of the 2010 haitian earthquake," *International Journal of Information Management*, vol. 31, no. 1, pp. 6 – 13, 2011.

[2] G. Carenini, R. T. Ng, and X. Zhou, "Summarizing email conversations with clue words," in *World Wide Web Conference Series*, 2007, pp. 91–100.

[3] J. G. V. Mittal, J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz, "Multi-document summarization by sentence extraction," in *Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*, 2000, pp. 40–48.

[4] 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*, 2010, pp. 279–288.

[5] D. Wang, S. Zhu, T. Li, Y. Chi, and Y. Gong, "Integrating document clustering and multidocument summarization," *ACM Trans. Knowl. Discov. Data*, vol. 5, pp. 14:1–14:26, August 2011. [Online]. Available: http://doi.acm.org/10.1145/1993077.1993078

[6] M. Dredze, H. M. Wallach, D. Puller, and F. Pereira, "Generating summary keywords for emails using topics," in *Proceedings of the 13th international conference on Intelligent user interfaces*, ser. IUI '08. New York, NY, USA: ACM, 2008, pp. 199–206. [Online]. Available: http://dx.doi.org/10.1145/1378773.1378800

[7] D. R. Radev, H. Jing, M. Stys, and D. Tam, "Centroid-based summarization of multiple documents," *Information Processing and Management*, vol. 40, no. 6, pp. 919 – 938, 2004.

[8] J. Zhu, C. Wang, X. He, J. Bu, C. Chen, S. Shang, M. Qu, and G. Lu, "Tag-oriented document summarization," in *Proceedings of the 18th international conference on World wide web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 1195–1196. [Online]. Available: http://doi.acm.org/10.1145/1526709.1526925

[9] Z. Yang, K. Cai, J. Tang, L. Zhang, Z. Su, and J. Li, "Social context summarization," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, ser. SIGIR '11. New York, NY, USA: ACM, 2011, pp. 255–264. [Online]. Available: http://doi.acm.org/10.1145/2009916.2009954

[10] E. Baralis, L. Cagliero, N. A. Mahoto, and A. Fiori, "Graphsum: Discovering correlations among multiple terms for graph-based summarization," *Inf. Sci.*, vol. 249, pp. 96–109, 2013.

[11] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of the seventh international conference on World Wide Web 7*, 1998, pp. 107–117.

[12] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, Sep. 1999.

[13] J. Steinberger, M. Kabadjov, R. Steinberger, H. Tanev, M. Turchi, and V. Zavarella, "JRC's participation at TAC 2011: Guided and multilingual summarization tasks," in *TAC'11: Proceedings of the The 2011 Text Analysis Conference*, 2011.

[14] D. Gillick, B. Favre, D. Hakkani-Tur, B. Bohnet, Y. Liu, and S. Xie, "The ICSI/TUD summarization system at TAC 2009," in *Proceedings of the Text Analysis Conference*, ser. TAC '09, Gaithersburg, MD (USA), 2009.

[15] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 510–520. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002537

[16] J. Hynek and K. Jezek, "Practical approach to automatic text summarization," in *ELPUB*, 2003.

[17] E. Baralis, L. Cagliero, S. Jabeen, and A. Fiori, "Multi-document summarization exploiting frequent itemsets," in *Proceedings of the ACM Symposium on Applied Computing, SAC 2012, Riva, Trento, Italy, March 26-30, 2012*, 2012, pp. 782–786. [Online]. Available: http://doi.acm.org/10.1145/2245276.2245427

[18] E. M. Baralis, L. Cagliero, A. Fiori, and S. Jabeen, "PatTexSum: A pattern-based text summarizer," in *Mining Complex Patterns Workshop*, 2011, pp. 18–29. [Online]. Available: http://porto.polito.it/2460874/

[19] E. Baralis, L. Cagliero, A. Fiori, and P. Garza, "Mwi-sum: A multilingual summarizer based on frequent weighted itemsets," *ACM Trans. Inf. Syst.*, vol. 34, no. 1, p. 5, 2015. [Online]. Available: http://doi.acm.org/10.1145/2809786

[20] M. Mampaey, N. Tatti, and J. Vreeken, "Tell me what I need to know: Succinctly summarizing data with itemsets," in *Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2011.

[21] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[22] Document Understanding Conference, "HTL/NAACL workshop on text summarization," 2004.

[23] E. Loper and S. Bird, "NLTK: the Natural Language Toolkit," in *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics - Volume 1*, ser. ETMTNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 63–70. [Online]. Available: http://dx.doi.org/10.3115/1118108.1118117

[24] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*. O'Reilly Media, 2009.

[25] J. M. Conroy, J. Goldstein, J. D. Schlesinger, and D. P. OLeary, "Left-brain/right-brain multi-document summarization," in *DUC 2004 Conference Proceedings*, 2004.

[26] J. Conroy, J. Schlesinger, J. Kubina, P. Rankel, and D. OLeary, "CLASSY 2011 at TAC: Guided and multi-lingual summaries and evaluation metrics," in *TAC'11: Proceedings of the The 2011 Text Analysis Conference*, 2011.

[27] R. Agrawal, T. Imielinski, and Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD 1993*, 1993, pp. 207–216.

[28] M. J. Zaki, "Mining non-redundant association rules," *Data Min. Knowl. Discov.*, vol. 9, no. 3, pp. 223–248, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:DAMI.0000040429.96086.c7

[29] S. Brin, R. Motwani, and C. Silverstein, "Beyond market baskets: Generalizing association rules to correlations," *SIGMOD Rec.*, vol. 26, no. 2, pp. 265–276, Jun. 1997. [Online]. Available: http://doi.acm.org/10.1145/253262.253327

[30] N. Tatti and M. Mampaey, "Using background knowledge to rank itemsets," *Data Min. Knowl. Discov.*, vol. 21, no. 2, pp. 293–309, 2010. [Online]. Available: http://dx.doi.org/10.1007/s10618-010-0188-4

[31] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, 2003, pp. 71–78.

[32] P.-N. Tan, V. Kumar, and J. Srivastava, "Selecting the right interestingness measure for association patterns," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, July 2002, pp. 32–41.

[33] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, Apr. 2004. [Online]. Available: http://dx.doi.org/10.1007/s00158-003-0368-6

[34] J. Castro-Gutierrez, D. Landa-Silva, and J. M. Pérez, "Improved Dynamic Lexicographic Ordering for Multi-Objective Optimisation," in *Parallel Problem Solving from Nature–PPSN XI, 11th International Conference, Proceedings, Part II*, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds. Kraków, Poland: Springer, Lecture Notes in Computer Science Vol. 6239, September 2010, pp. 31–40.

[35] T. Ralphs and M. Guzelsoy, "The SYMPHONY callable library for mixed integer programming," *The Next Wave in Computing, Optimization, and Decision Technologies*, vol. 29, pp. 61–76, 2006, software available at http://http://www.coin-or.org/SYMPHONY.

[36] O. Gross, A. Doucet, and H. Toivonen, "Document summarization based on word associations," in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, ser. SIGIR '14. New York, NY, USA: ACM, 2014, pp. 1023–1026. [Online]. Available: http://doi.acm.org/10.1145/2600428.2609500

[37] D. Gillick, B. Favre, and D. Hakkani-Tur, "The ICSI summarization system at TAC 2008," in *Proceedings of the Text Analysis Conference*, ser. TAC '08, Gaithersburg, MD (USA), 2008.

[38] N. Rotem, "Open text summarizer (OTS). Retrieved from http://libots.sourceforge.net/ in July 2011," 2011.

[39] TexLexAn, "TexLexAn: An open-source text summarizer. Retrieved from http://texlexan.sourceforge.net/ in July 2011," 2011. [Online]. Available: http://texlexan.sourceforge.net/

[40] W. T. Chuang and J. Yang, "Extracting sentence segments for text summarization: a machine learning approach," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '00. New York, NY, USA: ACM, 2000, pp. 152–159. [Online]. Available: http://doi.acm.org/10.1145/345508.345566

[41] M. Hofmann and R. Klinkenberg, *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC, 2013.

[42] T. G. Dietterich, "Approximate statistical test for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, 1998.