Knowl Inf Syst (2017) 50:945–968
DOI 10.1007/s10115-016-0966-4

REGULAR PAPER

Event-based summarization using a centrality-as-relevance model

Luís Marujo[1] · Ricardo Ribeiro[2,3] · Anatole Gershman[4] · David Martins de Matos[2,5] · João P. Neto[2,5] · Jaime Carbonell[4]

Received: 30 December 2014 / Revised: 29 November 2015 / Accepted: 4 June 2016 /
Published online: 21 June 2016

Abstract Event detection is a fundamental information extraction task, which has been explored largely in the context of question answering, topic detection and tracking, knowledge base population, news recommendation, and automatic summarization. In this article, we explore an event detection framework to improve a key phrase-guided centrality-based summarization model. Event detection is based on the fuzzy fingerprint method, which is able to detect all types of events in the ACE 2005 Multilingual Corpus. Our base summarization approach is a two-stage method that starts by extracting a collection of key phrases that will be used to help the centrality-as-relevance retrieval model. We explored three different ways to integrate event information, achieving state-of-the-art results in text and speech corpora: (1) filtering of nonevents, (2) event fingerprints as features, and (3) combination of filtering of nonevents and event fingerprints as features.

Keywords Event detection · Extractive summarization · Passage retrieval · Automatic key phrase extraction · Centrality

1 Introduction

Automatic summarization aims at creating new documents that, while shorter in length, capture the most important content of an input document or a set of documents. This new document, the summary, is characterized by several aspects, such as the origin of the content, the number of input units, or the coverage of the summary. Concerning its

✉ Luís Marujo
luis.marujo@gmail.com

1 Feedzai Research, Lisbon, Portugal

2 INESC-ID Lisboa, Lisbon, Portugal

3 Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal

4 School of Computer Science, CMU, Pittsburgh, USA

5 Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal

content, a summary might be composed by *extracts*, directly taken from the input, or *paraphrases*, which convey the content of a passage of the input using different words. In relation to the number of input units, if the input consists of only one document, the task is designated *single-document summarization*. However, when dealing with several input documents, we define the problem as *multi-document summarization*. Finally, the coverage of the input source(s) can be comprehensive, when creating *generic* summaries, or selective, if driven by an input *query*. A thorough analysis of the bibliography of this research area clearly shows that the main focus of automatic summarization is news stories: organizations need to have quick access to the information that affects them, and people want to be informed about the environment where they act. The bulk of this information is disseminated, either by written text, such as newspaper articles, or speech as broadcast news. Interestingly, although this type of documents is characterized by conveying information about events, most of the work concentrates on approaches that do not take into account this aspect.

We focus on extractive summarization, which means that the resulting summaries consist of a sequence of extracts (sentences, paragraphs, or, in some cases, sentence-like units if summarizing automatic transcriptions of spoken documents) that are selected according to a relevance rank of the selectable extracts of the input. In this article, we explore event detection to improve a key phrase-guided centrality-as-relevance summarization model. By event detection, we mean the identification and classification of events, such as the ones described in the ACE 2005 Multilingual Corpus [47]. Our approach is based on the fuzzy fingerprint method [13,22,36]. We combine event information with a two-stage summarization approach (KP-Centrality) [24,34]. The first stage consists of the identification of a set of phrases that capture the most important content of an input source (key phrase extraction). Then, this set of key phrases is used in a centrality-as-relevance summarization model to improve the detection of the most important passages. In centrality-as-relevance models, the detection of the most important passages is based on the identification of the central passages of the input source(s). Key phrases are well-known devices for reinforcing precision, thus improving centrality. We will use this framework to incorporate event information, exceeding state-of-the-art results in summarization. The two-stage method starts by extracting a collection of key phrases that are then used in a centrality-as-relevance summarization model. This summarization approach achieves state-of-the-art results and provides an adequate framework for the integration of additional information. Within this framework, we explore different ways of incorporating event information, attaining state-of-the-art results in both written and spoken language documents.

This document is structured as follows: Sect. 2 addresses relevant related work; Sect. 3 describes the event detection method; Sect. 4 presents our summarization model; Sect. 5 details the integration of event information; Sect. 6 describes the experimental validation; and conclusions and future work close the document.

2 Related work

We identify two lines of research related to our work: event detection and summarization. While most summarization work does not identify or classify events according to their type, as we propose, there are, however, approaches that explore pattern-based methods to extract event information. Regarding event detection, current work is mainly concerned with exploring supervised classification methods.

```
<!DOCTYPE source_file SYSTEM "apf.v5.1.1.dtd">
<source_file URI="CNN_ENG_20030630_085848.18.sgm"
             SOURCE="broadcast news"
             TYPE="text"
             AUTHOR="LDC"
             ENCODING="UTF-8">
<document DOCID="CNN_ENG_20030630_085848.18">

[...]

<event ID="CNN_ENG_20030630_085848.18-EV1"
       TYPE="Life"
       SUBTYPE="Injure"
       MODALITY="Asserted"
       POLARITY="Negative"
       GENERICITY="Specific"
       TENSE="Unspecified">
  <event_mention ID="CNN_ENG_20030630_085848.18-EV1-1">
    <extent><charseq START="337" END="344">injuries</charseq></extent>
    <ldc_scope><charseq START="334" END="388">no injuries have been
      reported thankfully hat this time</charseq></ldc_scope>
    <anchor><charseq START="337" END="344">injuries</charseq></anchor>
  </event_mention>
</event>
</document>
</source_file>
```

Fig. 1 ACE 2005 Multilingual Corpus event example

2.1 Event detection

In the late 1990s, the event detection problem was investigated under the topic detection and tracking (TDT) effort [2,5,49,51]. The TDT project was organized into two primary tasks: first story detection or new event detection (NED), and event tracking. The goal of the NED task was to discover documents discussing breaking new events from a news stream. The other task, event tracking, was focused on the tracking of articles describing the same event or topic over time. More recent work using the TDT datasets, focused on event threading which consists of tracking and linking several related events. Recent work [10,14,29] consists of organizing news articles about armed clashes into a sequence of events, but assumes that each article described a single event. Another related type of task, passage threading [10], extends event threading by relaxing the one event per news article assumption and uses a binary classifier to identify "violent" paragraphs.

Even though the TDT project ended in 2004, new event detection research followed: Automatic Content Extraction (ACE), and the associated ACE 2005 Multilingual Corpus, is the most pertinent example. The goal of the event annotations of the ACE corpus (Fig. 1) is the detection of events in text. In addition to the identification of events, the ACE 2005 [47] task identifies participants, relations, and attributes of each event. This extraction is an important step toward the overarching goal of building a knowledge base of events [16], which lead to new projects including the TREC temporal summarization task in 2013.

Event datasets are usually composed of several news articles. Experts defined a list of event types and annotated each sentence of the news articles. In practice, only a few sentences contain these types of events. Frequently, these sentences describe only one event and are complemented with other sentences describing other unrelated events (a type of event not included in the list) or “no events” such as a dateline, leading to an imbalanced dataset. As a result, it is hard to obtain good classification results in these imbalanced datasets with few examples of events. There are several ways to address this problem: namely to increase the number of examples of events through bootstrapping techniques, or augmenting the event-labeled dataset by including documents from other collections (cross-document techniques) such as MUC-6 (Message Understanding Conference, edition 6) [18]. Other works explore a wide range of features using supervised classifiers [30]. Generally, a drawback of these approaches is that performance rapidly decreases as the total number of event types or labels increases. In fact, for multi-label document classification in large datasets, probabilistic generative methods can outperform discriminative methods such as support vector machines [37]. For small datasets, such as the ACE 2005, there is not enough data to successfully learn either a generative or a discriminative model, as it is hard to model probabilistic distributions with few points. The alternative for these cases is to use a frequentist over a bayesian method, such as the Writeprint method [1] or the fuzzy fingerprints [13,22,36] method detailed in Sect. 3. This method has low computational runtime and memory requirements for both training and classification, especially when compared to probabilistic methods. In addition, it is able to detect examples of all event types. This is particularly important for this work, as it will have an impact in the filtering out of sentences without events.

2.2 Summarization

Most of the work in automatic summarization focuses on extractive summarization. In fact, extracting the important content is the first step of a generic summarization system. The extracted information can subsequently be further processed if the goal is to generate *abstracts*. The important content in the abstracts is generally devised as a set of concepts that are synthesized to form a smaller set and then used to generate a new, concise, and informative text. On the other hand, in *extracts*, the identified content consists of passages, sentences, or sentence-like unit (SUs), depending on the input, that are concatenated to form the summary.

Compared to written text summarization, summarization of speech documents presents additional challenges. When using text, it is feasible to use syntactic [46], semantic [15,43], and discourse information [45], in addition to features based on structure and significance metrics [28]. In fact, speech processing-related problems, such as recognition errors or disfluencies, constrain the use of text summarization approaches and greatly influence the subsequent processing. Conversely, speech-specific features, including acoustic/prosodic information [27] or recognition confidence scores [52], can provide useful information to determine salient content.

The architecture of our summarization model is similar to the ones using an unsupervised key phrase extraction step [20,35,41,48,53]. In contrast to the work by Litval and Last [20] that explores structural features and uses a graph-based representation in both supervised and unsupervised methods, we use a feature-rich supervised method for key phrase extraction. Moreover, their method does not include a second stage, while our approach uses a centrality-based summarization model. In this sense, closer to our work is the method proposed by Riedhammer et al. [35]: they also present a two-stage method, but the key phrase extraction step is based on part-of-speech information and use Maximal Marginal Relevance [4] as summarization model.

The idea of using event information in summarization was addressed for the first time by Daniel et al. [7]. In this work, summaries are manually created and evaluated against a generic automatic multi-document summarization system. Being essentially human-based, this was an exploratory work to assess the impact of using event information on summarization. Another important aspect is the definition of event: while we use a fine grained, sentence-level definition (ACE-based definition), they used a document-level definition of event (TDT-based definition).

Closely related to our idea is previous work [11,12,17,21] that defines **event**, at sentence-level, as a triplet composed by a **named entity**, a **verb** or **action noun**, and another **named entity**, where the verb/action noun defines a relation between the two named entities. This information is usually included in a generic unit selection model, often trying to minimize redundancy while maximizing the score of the important content. In our work, we use not only event information, but also its classification according to ACE [47]; additionally, we explore the possibility of using events to filter out unimportant content, and to the best of our knowledge, we present the first analysis of the impact of using event information on the summarization of spoken documents.

More recent work based on the idea of combining event information and summarization has been studied in microblog summarization, but it is still in early stages. Many of the Twitter summarization works are restricted to specific filtered streams (topics) [8,39,40,42,44], or are combined with event detection [32], such as sports matches [6,31,32].

3 Event detection

Our event detection algorithm is based on the fuzzy fingerprints classification method [13, 22,36]. Homem and Carvalho [13] approached the problem of authorship identification by using the crime scene fingerprint analogy, to claim that a given text has its author's writing style embedded in it. The algorithm works as follows:

1. Gather the top-k most frequent words (and their frequencies) in all known texts of each known author.
2. Build the fingerprint by applying a fuzzifying function to the top-k list. The fuzzified fingerprint is based on the word order and not on the frequency value.
3. Perform the same calculations for the text being identified and then compare the obtained text fuzzy fingerprint with all available author fuzzy fingerprints. The most similar fingerprint is chosen, and the text is assigned to the author of the fingerprint.

The method, when used for event detection [22], is similar in intention and form, but differs in a few crucial steps. First, it is important to establish the parallel between authorship identification and event detection. Instead of author fingerprints, in this context, we are looking for fingerprints of events in each passage (sentences in text or SUs in speech). This results in passages classified according to the matched fingerprint (event type). The process starts with the creation of an event fingerprint library. Then, each unclassified passage can be processed and compared to the fingerprints existing in the event library. Second, a different criterion was used in ordering the top-k words for the fingerprint. While Homem and Carvalho [13] use word frequency as the main feature to create and order the top-k list, we use an adaptation of the inverse document frequency (IDF) technique, aiming at reducing the influence of frequent terms that are common across several events.

3.1 Building the event fingerprint library

The first step of the event fingerprint library creation stage is computing word frequencies for each event type. We use as training corpus the ACE 2005 corpus [47]. Only the top-k most frequent words are considered. The main difference between the original method and the one used here is due to the small size of each sentence: in order to make the different event fingerprints as unique as possible, its words should also be as unique as possible. Therefore, in addition to counting each word's occurrence, we also account for its inverse topic frequency (ITF), an adaptation of IDF: $\text{itf}_v = \frac{N}{n_v}$, where N is the cardinality of the event fingerprint library (i.e., the total number of events), and n_v becomes the number of fingerprint events where the specific word v is present. After obtaining the top-k list for a given event, we follow the original method and apply a fuzzy membership function to build the fingerprint. The selected membership function (Eq. 1) is a Pareto-like linear function, where 20 % of the top-k elements assume 80 % of the membership degree.

$$\mu(i) = \begin{cases} 1 - (1-b)\frac{i}{k} & \text{if} \quad i \leq ak \\ a\left(1 - \frac{i-a}{k-a}\right) & \text{if} \quad i > ak \end{cases} \quad \text{with } a, b = 0.2 \tag{1}$$

The fingerprint is a $k \times 2$ matrix, where the first column contains the list of the top-k words ordered by their TF–ITF score, and the second column contains the membership value of word i, $\mu(i)$, obtained by the application of Eq. (1). Table 1 shows two examples of event fingerprints ordered by $\mu(i)$ values for the event types *start-organization* and *meet*. The table does not include the complete fingerprints due to space constrains. In the table, we show the top ten entries, the bottom 3 and some intermediate entries. Each entry contains the rank based on $\mu(i)$, calculated setting $K = 600$, word i (where i value is the $TF - ITF$ rank), and $\mu(i)$ the membership value.

3.2 Classifying sentences/sentence-like units

The method for authorship identification has three steps: build the document fingerprint (using the previously described algorithm); compare the document fingerprint with every fingerprint present in the library; and choose the match with highest score. However, for event detection, such approach would not be feasible due to the small number of words comprised in one sentence/SU. This problem is addressed by the Sentence-to-Event score (S2E) that tests the fitness of a sentence/SU to a given event fingerprint. The S2E function (Eq. 2) provides a normalized value ranging between 0 and 1 that takes into account the size of the (preprocessed) sentence/SU (i.e., its number of features). In the present work, we use as features the words of the sentences/SUs. We do not remove stop-words (empirical results show that the best results are obtained without removing stop-words nor by imposing a minimum word size).

$$\text{S2E}(\Phi, S) = \frac{\sum_{v \in \Phi \cap S} \mu_\Phi(v)}{\sum_{i=0}^{j} \mu_\Phi(w_i)} \tag{2}$$

In Eq. (2), Φ is the event fingerprint, S is the set of words of the sentence/SU, $\mu_\Phi(v)$ is the membership degree of word v in the event fingerprint, and j is the number of features of the sentence/SU. Essentially, S2E divides the sum of the membership values $\mu_\Phi(v)$ of every word v, that is common between the sentence/SU and the event fingerprint, by the sum of the top-j membership values in $\mu_\Phi(w_i)$ where $w_i \in \Phi$. Equation (2) will tend to 1 when most to

Table 1 Event fingerprints for start-organization (left) and meet (right) event types order by $\mu(i)$

Rank	Word	$\mu(i)$	Rank	Word	$\mu(i)$
1	Founded	34.3897	1	Meet	33.8333
2	Committee	32.7875	2	Summit	31.9200
3	Collectors	32.4060	3	Ended	31.5000
4	Sheik	32.0245	4	Discussed	30.9400
5	Films	31.4142	5	Meetings	30.8933
6	Budget	31.3379	6	Eu	30.2400
7	Reformist	30.8038	7	Meeting	28.8200
8	Hamshahri	30.6512	8	Discuss	29.4933
9	Cinema	30.4223	9	Saint	29.4467
10	Forging	30.1935	10	Talks	17.1967
24	Launched	10.7511	28	Meets	5.2634
34	Opening	5.5725	38	Talk	5.0322
67	Business	4.4114	119	Contacting	2.6025
100	Empire	3.1298	199	Talked	1.8089
101	Contract	3.1107	240	Resolution	1.5405
178	Acquired	1.4312	360	Reunited	0.8986
195	Launching	1.2786	394	Organization	0.7352
265	Make	0.5324	414	Met	0.6652
365	Then	0.0310	598	Off	0.0117
366	Year	0.0057	599	Eased	0.0117
367	Been	7.3399E−4	600	Knows	0.0078

all words of the sentence/SU belong to the top words of the fingerprint, and tends to 0 when none or very few words of the sentence/SU belong to the bottom words of the fingerprint.

The description of our event detection based on fingerprints is not complete without a brief discussion about the method performance of our method against a strong baseline (SVM). Our results [22] show that it is possible to detect all 26 different event types defined in the ACE 2005 dataset when using the fuzzy fingerprints approach, while the best competitor, an SVM classifier with enhanced features, only detects roughly 85 % of the different types of events. This leads to a large increase in the G-mean scores (imbalanced classification evaluation metric) when using the fuzzy fingerprints method.

The fuzzy fingerprints method also has the advantage of being much more efficient. In our test conditions, it is more than 20× faster than SVM when classifying the 26 event types.

4 Summarization

Our summarization approach [34] is a two-stage method that starts by extracting a collection of key phrases that are used to guide a centrality-as-relevance summarization model [33]. This approach achieves state-of-the-art results, both in noisy and in clean data, that were confirmed by using as input both automatically transcribed spoken documents and written documents. It is language and genre independent. For instance, the summarization approach was applied to Portuguese and English documents, including broadcast news and event reports. We also demonstrate that our summarization approach is an adequate framework for the integration of additional information, namely the use of key phrases to improve the core centrality-as-relevance summarization model.

Centrality-as-relevance methods consider the most important content to be the most central one. Generally, centrality can be computed in two ways. One way is by comparing each passage to a representative passage of the input—the *centroid*. Passages closer to the centroid are the most central and, consequently, the most important ones. The other way to determine the central passages is to compute the average distance of each passage to every other passage and select the ones closer to every other passage (lower average distance).

Our underlying centrality-as-relevance model is based on the notion of support set. For each passage, we create a list of the most semantically related passages. This list is called a support set. The most important passages are the ones that appear in the largest number of support set. Ribeiro and de Matos [33] explore several metrics to compute semantic relatedness and propose different ways to estimate the cardinality of the support sets. In this work, we use the heuristics based on passage order. This type of heuristics explores the structure of the input source to partition the candidate passages to be in the support set in two subsets: the ones closer to the passage associated with the support set under construction, and the ones further apart (see Algorithm 4.1).

Algorithm 4.1 Generic passage order-based heuristic.

Input: Two values r_1 and r_2, each a representative of a subset, and the set of the passages p_k and corresponding distances d_k^i to the passage p_i associated with the support set under construction.
Output: The support set of the extract under analysis.

$R_1 \leftarrow \emptyset$, $R_2 \leftarrow \emptyset$
for $k \leftarrow 1$ to $N-1$ **do**
  if $|r_1 - d_k^i| < |r_2 - d_k^i|$ **then**
    $r_1 \leftarrow (r_1 + d_k^i)/2$
    $R_1 \leftarrow R_1 \cup \{e_k\}$
  else
    $r_2 \leftarrow (r_2 + d_k^i)/2$
    $R_2 \leftarrow R_2 \cup \{e_k\}$
  end
end
$l \leftarrow \arg\min_{1 \le k \le N-1}(d_k^i)$
if $p_l \in R_1$ **then**
  return R_1
else
  return R_2
end

Considering a segmented input document $I \triangleq p_1, p_2, \ldots, p_N$, the set of the most semantically related extracts of an extract p_i is defined by Eq. (3), where $sim()$ is a similarity function (e.g., cosine) and ε_i is a threshold.

$$S_i \triangleq \{s \in I : \mathrm{sim}(s, p_i) > \varepsilon_i \wedge s \neq p_i\} \tag{3}$$

The model is defined by Eq. (4), which specifies that extracts are ranked in accordance with the number of support sets containing the extract.

$$\underset{s \in \cup_{i=1}^{n} S_i}{\arg\max} \left| \{S_i : s \in S_i\} \right| \quad (4)$$

Our framework [24,34] adapts this model, extending it to a two-stage method. The first stage is an automatic key phrase extraction step, and the second stage is a modified version of the centrality model that uses key phrases to improve the estimation of the most important content. Previous work explores several ways of integrating the key phrases, but here we use only the most successful one: Key Phrase-based Centrality (KP-Centrality). In this approach, key phrases, $K \triangleq k_1, k_2, \ldots, k_M$, are considered separate passages, even if they are contained in the original passages, $I \cup K \triangleq q_1, q_2, \ldots, q_{N+M}$, augmenting the number of support sets, and, therefore, changing centrality (Eq. 5 shows how support sets are defined).

$$S_i \triangleq \{s \in I \cup K : \mathrm{sim}(s, q_i) > \varepsilon_i \wedge s \neq q_i\}, \quad i = 1, \ldots, N + M \quad (5)$$

As defined in Eq. (6), extracts are ranked excluding key phrases (i.e., the final ranking contains only extracts from the input document).

$$\underset{s \in (\cup_{i=1}^{n} S_i) - K}{\arg\max} \left| \{S_i : s \in S_i\} \right| \quad (6)$$

For automatic key phrase extraction, we use the approach described by Marujo et al. [23,25], which is a fairly traditional supervised method enhanced with additional semantic features and preprocessing steps, namely Light Filtering and Co-reference normalization. These new features included the detection of rhetorical devices, Freebase [3] sub-categories, and news articles top categories. Including such new features and preprocessing steps improved the key phrase extraction results beyond the state-of-the-art. Therefore, we used the methodology described by Marujo et al. [23] in our summarization experiments. However, since the preprocessing steps can have impact on the outcome of our experiments, in particular affecting our event-based filter, we decided to explore the preprocessing steps with the event-based method in future work. This fact led to the exclusion of the Freebase subcategories which were only beneficial in combination with the preprocessing steps. The news articles' top categories were also not available in the datasets used.

5 Event-based summarization

Event-based summarization consists of extracting the most important events described in sentences or SUs from the input document(s).

Event detection produces a list of passages containing events. A simple approach to create event-based summaries is to rank this list of passages using a single-document summarizer, such as KP-Centrality. This approach has the disadvantage of not providing any event information about the passages to the ranking algorithm. Consequently, this may lead the algorithm to fail to detect passages about the same event, but written using different lexical realizations.

The simplest, but effective, way to avoid this limitation is to include event information as additional features for the ranking algorithm. Since these two alternatives are not mutually exclusive, it also possible to combine them.

In the rest of this section, we describe in detail our event-based summarization methods.

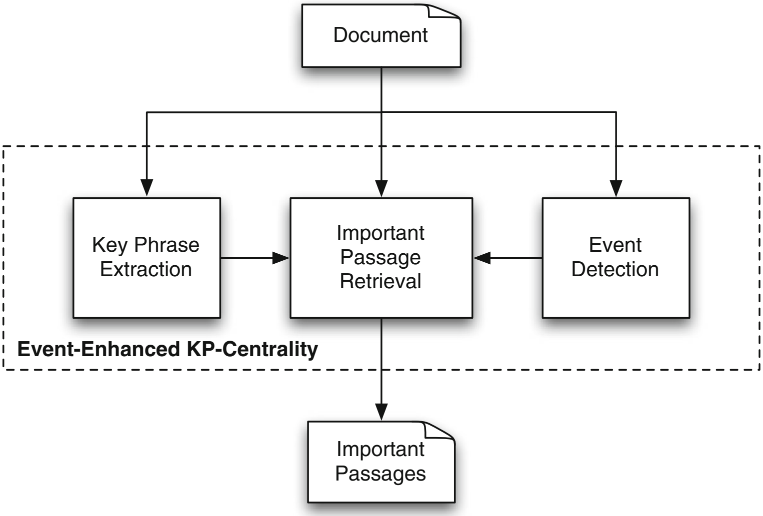


Fig. 2 EE-KPC architecture

5.1 Event-enhanced KP-Centrality (EE-KPC)

As previously mentioned, the KP-Centrality method consists of two steps: first, it extracts key phrases using a supervised approach, and then, it combines them with a bag-of-words model, represented by a terms-by-passages matrix, to compute the most important content.

The EE-KPC method includes event information in the KP-Centrality-based summarization process at the important passage retrieval module level. This is accomplished by expanding the bag-of-words matrix representation of passages with event descriptors—vectors of S2E describing each event type obtained using the event fingerprint method for each sentence/SU and key phrase. Figure 2 shows the complete architecture.

Equation 7 defines the new matrix representation, where w is a function of the number of occurrences of term t_i in extract e_j or key phrase k_l, T is the number of terms, M is the number of key phrases, c is a function of the S2E score of each extract e_j or key phrase k_l for each event type ev_m.

$$\begin{bmatrix} w(t_1, e_1) & \ldots & w(t_1, e_N) & w(t_1, k_1) & \ldots & w(t_1, k_M) \\ \vdots & & & & & \vdots \\ w(t_T, e_1) & \ldots & w(t_T, e_N) & w(t_T, k_1) & \ldots & w(t_T, k_M) \\ \mathbf{c}(\mathbf{ev_1}, \mathbf{e_1}) & \ldots & \mathbf{c}(\mathbf{ev_1}, \mathbf{e_N}) & \mathbf{c}(\mathbf{ev_1}, \mathbf{k_1}) & \ldots & \mathbf{c}(\mathbf{ev_1}, \mathbf{k_M}) \\ \vdots & & & & & \vdots \\ \mathbf{c}(\mathbf{ev_E}, \mathbf{e_1}) & \ldots & \mathbf{c}(\mathbf{ev_E}, \mathbf{e_N}) & \mathbf{c}(\mathbf{ev_E}, \mathbf{k_1}) & \ldots & \mathbf{c}(\mathbf{ev_E}, \mathbf{k_M}) \end{bmatrix} \tag{7}$$

Each column represents an extract p_i. The extracts are ranked to produce a summary according to Eqs. 5 and 6.

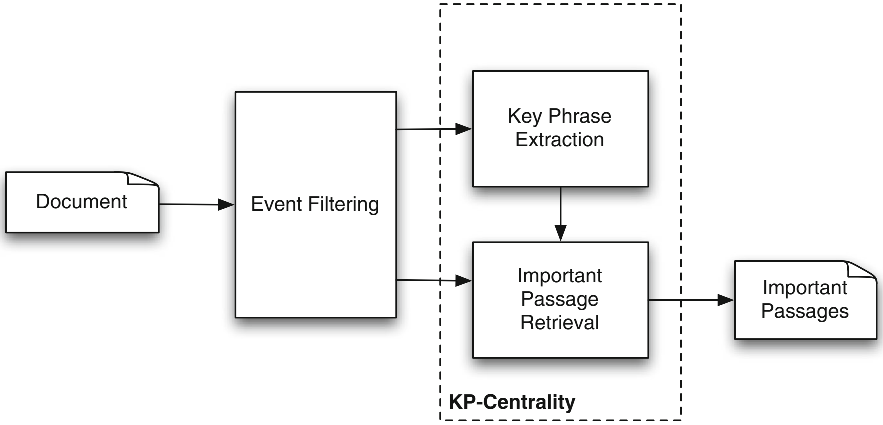


Fig. 3 EF-KPC architecture

5.2 Event filtering-based KP-Centrality (EF-KPC)

The EF-KPC method includes the same stages of the previous method, EE-KPC, but it uses event information in a different manner. Instead of expanding the bag-of-words matrix representation, it discards sentences/SUs that do not contain events. This corresponds to the event filtering stage shown in Fig. 3, which includes an event detection step. All passages that the event fingerprint method classifies as not containing any event are removed. The exception to this simple rule occurs when the method is not confident about the classification result ($\max$ S2E < 0.0001). Then, KP-Centrality is used to produce a summary. Note that, although pattern matching-based methods could adopt this strategy, our event detection method is more robust and allows us to correctly identify a larger number of event types. In fact, we are able to detect events in passages with different syntactic structures, something which is more difficult to accomplish for pattern matching-based methods.

For example, the passage "Four marines died in the crash of an Osprey in North Carolina." that appear in the Concisus dataset is not detected by pattern matching-based methods because it does not have a named entity followed by a verb or action noun, and another named entity. Within the passage, there is only one named entity, the location North Carolina. Our event detector is able to detect that the passage describes a *Die* event type (S2E score = 0.1553).

5.3 Combination of event filtering-based and event-enhanced KP-Centrality (CE-KPC)

The CE-KPC method combines the two previous methods, as shown in Fig. 4. It starts by filtering the passages without events, as in the EF-KPC method, and includes the S2E-based event descriptors in the bag-of-words matrix representation, as in the EE-KPC method.

6 Experiments

To perform the evaluation of the detection of the most important sentences/SUs, we used ROUGE [19], namely ROUGE-1, which is the most widely used evaluation measure for this

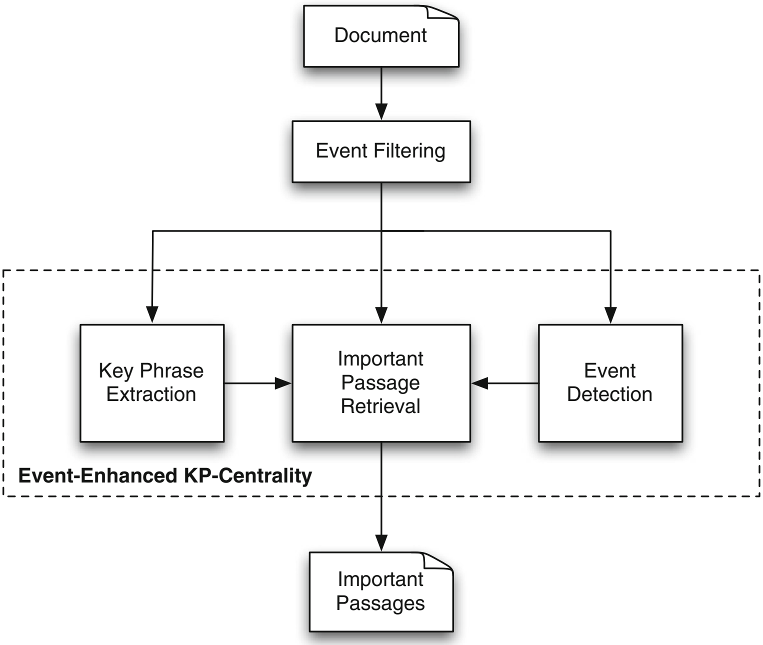


Fig. 4 CE-KPC architecture

scenario. In the following experiments, we generate three sentence summaries, commonly found in online news Web sites and news aggregators, such as Google News.[1]

6.1 Datasets

To assess the influence of using event information in our extractive summarization method, we use three datasets: the Concisus Corpus of Event Summaries [38]; a subcorpus of the Columbia Broadcast News Speech Summarization Corpus [26]; and the DUC 2004 task 1 corpus.[2] The first dataset is composed by event reports (written text) and can be seen as an ideal dataset for this kind of approach because the list of events types of our event detector includes the main events of the dataset. The second dataset is composed of common broadcast news stories. The third dataset is composed by written news stories. The use of these three datasets provides different experimental conditions, helping to better understand the real impact of the method.

6.1.1 Concisus corpus of event summaries

The corpus is composed by 78 event reports and corresponding summaries, distributed across three different types of events: aviation accidents, earthquakes, and train accidents. This corpus also contains comparable data in Spanish. However, since our event detection was trained for English, we opted for not using that part of the dataset. Table 2 shows statistics

1 https://news.google.com/.

2 See http://duc.nist.gov/duc2004/tasks.html.

Table 2 Statistics about Concisus corpus of event summaries (english part)

	#Docs	Avg. #sentences	Avg. #words
Input documents	77	10.377	224.922
Reference summaries	77	1.831	65.820

Table 3 Statistics about Columbia Broadcast News Speech Summarization Corpus test set

	#Docs	Avg. #sentences	Avg. #words
Input documents	16	11.313	203.313
Reference summaries	16	3.563	70.688

Table 4 Statistics about DUC 2004

	#Docs	Avg. #sentences	Avg. #words
Input documents	489	26.278	590.701
Reference summaries	489	1	10.219

about the size of the corpus, namely number of documents, average number of sentences, and average number of words.

6.1.2 Columbia Broadcast News Speech Summarization Corpus test set

The corpus consists of a random sample of 16 broadcast news stories from the test subcorpus of the Columbia Broadcast News Speech Summarization Corpus II (CBNSCII). The CBNSCII is composed by 20 CNN Headlines News shows from TDT-4 corpus. For each news story, there is a human summary that is used as reference. Table 3 provides some statistics about the corpus.

6.1.3 DUC 2004

We used the DUC 2004 dataset created for task 1. The goal of task 1 was to generate a headline, which is assumed to be 75-byte single-document summary. The documents come from the 50 TDT clusters. Each cluster contains about ten documents. The source of documents is the AP newswire and New York Times newswire. The total number of news documents is 489. Table 4 includes additional statistics about the corpus.

6.2 Results

Table 5 shows the ROUGE-1 results for the Concisus dataset; Table 7 shows the results for the Columbia Broadcast News dataset and Table 9 for the DUC 2004 dataset. As previously mentioned, it is possible to use different metrics to compute semantic similarity in the centrality-as-relevance summarization model. In these experiments, we explored the best performing metrics (for clean and noisy data) as presented by Ribeiro and de Matos [33]: cosine similarity and frac133, that is, the generic Minkowski distance, Eq. (8), in which N is

Table 5 ROUGE-1 scores for the Concisus dataset

#Key phrases	30	40	50	60
LexRank	0.428			
Event-based	0.443			
Centrality	0.443			
KP-Centrality	0.572	0.575$^\diamond$	0.581	0.570
EE-KPC	0.574	**0.586**$^\diamond$	0.585	0.581
EF-KPC	0.574	0.584	0.583	0.581
CE-KPC	0.574	0.584	0.583	0.579

Bold value indicates the best results
$\diamond$ indicates statistical significance difference under macro t test after rank transformation (p value <0.05)

Table 6 Percentage of number of sentences that are different between KP-Centrality and the event-based generated 3-sentence summaries in the Concisus corpus using 40 key phrases

	KP-Centrality	EE-KPC	EF-KPC	CE-KPC
KP-Centrality	0.00	22.41	18.97	19.40
EE-KPC		0.00	16.38	11.64
EF-KPC			0.00	6.47
CE-KPC				0.00

set to 1.(3). Since results using frac133 for the Concisus dataset do not show improvement over the baseline, we opted for not presenting them. In the same way, it is possible to configure Algorithm 1, used by the centrality-as-relevance summarization model, with different representatives (r_1 and r_2) of the subsets corresponding to the passages closer and passages further apart to the passage under analysis. Again, in these experiments, we used the best performing configuration reported by Ribeiro and de Matos, with r_1 and r_2 corresponding to the distances to the first and second passages of the document, respectively. In this way, the support set-specific threshold (ε_i in Eq. 5) is dynamically set to a non-explicit value (Tables 6, 8).

$$\mathrm{dist}_{\mathrm{minkowski}}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^N\right)^{\frac{1}{N}} \tag{8}$$

To better assess the performance of the proposed methods, we also provide results for common baselines: LexRank [9], support sets-based centrality-as-relevance (centrality) [33], and an event-based summarizer (event-based) [11].

For all the performed experiments, the use of event information clearly improves baselines: for the Concisus dataset, we observe differences between EE-KPC and KP-Centrality, using 40 key phrases, with statistical significance (p value <0.05) under the macro t test after rank transformation [50]; in the broadcast news dataset, both EF-KPC (frac133, 30 and 50 key phrases) and CE-KPC (frac133, 50 key phrases) are significantly better than KP-Centrality using the same statistical test (p value <0.04); finally, on the DUC 2004 dataset, it was also possible to observe statistically significant differences between both CE-KPC and EE-KPC and KP-Centrality, using 30 key phrases (p value <0.05).

Another interesting aspect is that, even though not all variations achieve statistically significant differences, the resulting summaries are still different. As we can see in Table 6, for the Concisus dataset, differences between the variants and the KP-Centrality baseline range from

Table 7 ROUGE-1 scores for the Columbia Broadcast News dataset

#Key phrases	30	40	50	60
Similarity metric: cosine				
Centrality	0.564			
KP-Centrality	0.673	0.697	0.656	0.684
EE-KPC	0.673	0.697	0.656	0.684
EF-KPC	0.678	**0.710**	0.668	0.692
CE-KPC	0.678	**0.710**	0.668	0.692
Similarity metric: frac133				
Centrality	0.700			
KP-Centrality	0.695‡	0.702	0.738†,◇	0.698
EE-KPC	0.714	0.693	0.743	0.700
EF-KPC	0.729‡	0.699	**0.752**†	0.702
CE-KPC	0.712	0.691	**0.752**◇	0.702
LexRank	0.653			
Event-based	0.132			

Bold values indicate the best results
Compared pairs of systems are marked with the same symbol (‡, †, ◇); differences are statistically significant under the macro t test after rank transformation (p value <0.04)

Table 8 Differences (in percentage) in terms of number of sentences between KP-Centrality and the event-based generated 3-sentence summaries (50 key phrases) in the broadcast news corpus

	KP-Centrality	EE-KPC	EF-KPC	CE-KPC
KP-Centrality	0.00	6.38	10.64	23.40
EE-KPC		0.00	19.15	19.15
EF-KPC			0.00	12.77
CE-KPC				0.00

Table 9 ROUGE-1 scores for the DUC 2004 dataset

#Key phrases	30	40	50	60
LexRank	0.352			
Event-based	0.371			
Centrality	0.407			
KP-Centrality	0.441◇,†	0.443	0.441	0.439
EE-KPC	**0.445** ◇	0.442	0.441	0.440
EF-KPC	0.442	0.444	0.441	0.439
CE-KPC	**0.445** †	0.440	0.441	0.439

Bold values indicate the best results
◇ and † indicates statistically significant difference under macro t test after rank transformation (p value <0.05)

18.97 to 22.41 %. Table 8 shows the same information for the broadcast news dataset, with differences ranging from 6.38 to 23.40 %. Table 10 shows the differences between summaries in the DUC 2004 dataset, with values ranging from 10.4 to 24.7 %. In fact, having different summarization approaches generating different summaries with similar performances is in line with the possibility of having different good summaries for the same document.

Table 10 Differences (in percentage) in terms of number of sentences between KP-Centrality and the event-based generated 3-sentence summaries (30 key phrases) in the DUC 2004

	KP-Centrality	EE-KPC	EF-KPC	CE-KPC
KP-Centrality	0.00	24.7	10.4	24.0
EE-KPC		0.00	25.6	11.9
EF-KPC			0.00	24.1
CE-KPC				0.00

Table 11 Statistics about event classification and filtering on the Concisus, the CBNSCII, and DUC 2004 datasets

	Concisus	CBNSCII	DUC 2004
#Sentences/SUs	815	181	12,850
Avg. #sentences/SUs	10	11	26
#Sentences/SUs after filtering	786 (96 %)	172 (95 %)	12,485 (97 %)
Avg. #sentences/SUs after filtering	10	10	26
#Event-classified sentences/SUs	734 (90 %)	159 (88 %)	10,534 (82 %)
Avg. #event-classified sentences/SUs	9	9	21

The best results on the Concisus dataset are obtained when integrating event information as a feature in the summarization model (EE-KPC), followed by the use of event information to filter out unimportant content (EF-KPC) and the combination of both strategies (CE-KPC). Contrary to what was observed in the Concisus dataset, the EF-KPC and CE-KPC methods outperform the EE-KPC method in the broadcast news corpus. The justification for this difference in performance is threefold. The first reason is the noisy nature of speech data, where removing passages without events helps discarding unimportant content. The second reason is that the percentage of sentences/SUs filtered out is higher in the broadcast news corpus than in the Concisus corpus, as shown in Table 11. The third reason is the nature of the corpus: since the Concisus corpus is composed of event reports, the filtering approach does not have the same impact as in the broadcast news corpus, and the use of event information as a feature helps distinguishing the most important facts or events within each report. In addition, the length of the documents to be summarized can influence the performance of EE-KPC, since the number of event features is constant and the number of term features increases according to Zipf's law. Concerning the better performance of the semantic relatedness metric frac133 over cosine similarity, which, in general, is the best performing metric, it might be related to the influence of the S2E values. These values vary inversely proportional to sentence/SU length. The high average sentence length of the Concisus corpus makes S2E lower, and the use frac133 makes it more difficult to distinguish close passages. On the other hand, on the broadcast news corpus, the average SU length is lower and, inversely, S2E values are higher, which makes frac133 more effective. In what concerns the DUC 2004 dataset, its similarities with the Concisus dataset justify the results achieved using the EE-KPC method. On the other hand, the performance of the CE-KPC method is mainly due to average length of the input documents of this dataset, considerably higher than of the documents of the other two datasets.

Table 12 Example of important passage retrieval using KP-Centrality, EE-KPC, EF-KPC, and CE-KPC

Document terremoto-31011906
The 1806 Ecuador–Colombia earthquake occurred at 15:36 UTC on January 31, off the coast of Ecuador, near Esmeraldas. The earthquake had a magnitude of 8.8 and triggered a destructive tsunami that caused at least 500 casualties on the coast of Colombia
The earthquake occurred along the boundary between the Nazca plate and the South American plate. The earthquake is likely to be a result of thrust faulting, caused by the subduction of the Nazca plate beneath the South American plate
The coastal parts of Ecuador and Colombia have a history of great megathrust earthquakes originating from this plate boundary
The greatest damage from the tsunami occurred on the coast between Ro Verde, Ecuador and Micay, Colombia. Estimates of the number of deaths caused by the tsunami vary between 500 and 1500
Event classification of the sentences in the document
(Event = Charge-Indict, S2E = 0.011)—the 1806 Ecuador–Colombia earthquake occurred at 15:36 UTC on January 31, off the coast of Ecuador, near Esmeraldas
(Event = Injure, S2E = 0.018)—the earthquake had a magnitude of 8.8 and triggered a destructive tsunami that caused at least 500 casualties on the coast of Colombia
(Event = *N*, S2E = 1.4E-45)—the earthquake occurred along the boundary between the Nazca plate and the South American plate
(Event = Injure, S2E = 0.021)—the earthquake is likely to be a result of thrust faulting, caused by the subduction of the Nazca plate beneath the South American plate
(Event = *N*, S2E = 0.016)—the coastal parts of Ecuador and Colombia have a history of great megathrust earthquakes originating from this plate boundary
(Event = Charge-Indict, S2E = 0.012)—the greatest damage from the tsunami occurred on the coast between Ro Verde , Ecuador and Micay, Colombia.
(Event = Die, S2E = 0.041)—estimates of the number of deaths caused by the tsunami vary between 500 and 1500
Three-passage summary using KP-Centrality
The earthquake is likely to be a result of thrust faulting , caused by the subduction of the Nazca plate beneath the South American plate
The earthquake had a magnitude of 8.8 and triggered a destructive tsunami that caused at least 500 casualties on the coast of Colombia
The coastal parts of Ecuador and Colombia have a history of great megathrust earthquakes originating from this plate boundary
Three-passage summary using EE-KPC
The earthquake is likely to be a result of thrust faulting , caused by the subduction of the Nazca plate beneath the South American plate

Table 12 continued

Document terremoto-31011906
The greatest damage from the tsunami occurred on the coast between Ro Verde, Ecuador and Micay, Colombia
The earthquake had a magnitude of 8.8 and triggered a destructive tsunami that caused at least 500 casualties on the coast of Colombia
Three-passage summary using EF-KPC
The earthquake had a magnitude of 8.8 and triggered a destructive tsunami that caused at least 500 casualties on the coast of Colombia
The earthquake is likely to be a result of thrust faulting, caused by the subduction of the Nazca plate beneath the South American plate
The 1906 Ecuador–Colombia earthquake occurred at 15:36 UTC on January 31, off the coast of Ecuador, near Esmeraldas
Three-passage summary using CE-KPC
The earthquake had a magnitude of 8.8 and triggered a destructive tsunami that caused at least 500 casualties on the coast of Colombia
The earthquake is likely to be a result of thrust faulting , caused by the subduction of the Nazca plate beneath the South American plate
The greatest damage from the tsunami occurred on the coast between Ro Verde, Ecuador and Micay, Colombia
Reference
January 31, 1906
The 1906 Ecuador–Colombia earthquake occurred at 15:36 UTC on January 31, off the coast of Ecuador, near Esmeraldas
The earthquake had a magnitude of 8.8 and triggered a destructive tsunami that caused at least 500 casualties on the coast of Colombia

All methods use 40 key phrases and a document from the Concisus corpus of event summaries

Table 11 shows the specific effects of using event information: as we can see, the number of sentences/SUs classified as containing, at least, one event type is high ($\approx$90 % for the Concisus dataset; 88 % for the broadcast news corpus; and 82 % for the DUC 2004 dataset). However, the number of sentences/SUs kept after filtering is even higher ($\approx$96 % for the Concisus dataset; 95 % for the broadcast news corpus; and 97 % for the DUC 2004 dataset). The reason to keep such large number of sentences/SUs after filtering is mainly due to the need to cope with the classifier errors, as described in Sect. 5.

Table 12 shows an example of important passages retrieved using KP-Centrality, EE-KPC, EF-KPC, CE-KPC methods. The methods are configured with 40 key phrases, which is the best configuration found for the Concisus corpus. We also included the event label and respective S2E values for each sentence/SU of the original document. The event detector identified two sentences that do not cover any of the event types (N—"no event" or "null event"). However, only one of the sentences obtained a "high" S2E score (S2E $>$ 0.0001). This kind of sentences usually describes secondary topics and details. The EF-KPC filter explores this information to filter out irrelevant sentences to improve the quality of the summaries.

According to ROUGE, the summary produced by EF-KPC, shown in Table 12, should be the best, but for the human reader, it looks the most incoherent. One aspect that was not given attention in this work was the order in which sentences appear in the summary. One simple solution is to order the sentences in the order they appear in the original document. Yet, existing metrics do not take into account the order in which sentences occur, making it difficult to evaluate the sentence order. As it is not trivial to define and test a metric for this problem, this will be left as future work.

Notice that the reference summary, in Table 12, was composed of the first two sentences or the original article, which is consistent with the standard model of news articles, namely that the first paragraph provides a good summary.

7 Conclusions

In this work, we introduced three event-based summarization techniques that perform above current state-of-the-art methods. Our event detection method is based on the fuzzy fingerprints classification method and trained on the ACE 2005 Multilingual Corpus. The obtained event information is integrated in a two-stage summarization method in three ways: one approach consists of expanding the feature representation of sentences/SUs with event information (EE-KPC); another technique filters out sentences/SUs without events (EF-KPC); and finally, we also explore the combination of both techniques (CE-KPC). The approach that yielded the best results in the written text dataset (the Concisus corpus of event reports and DUC 2004) was EE-KPC. The use of event information to filter out unimportant passages was the best performing approach in the speech dataset, the Columbia Broadcast News Corpus. Still, EE-KPC also achieved better results than the baselines. In general, CE-KPC had a similar or worse performance than EF-KPC, because this method accumulates errors from both stages. Since the filtering stage discards sentences/SUs without events, the available segments to be selected are similar to EF-KPC method. Inherently, the next stage cannot overcome the errors made in the filtering step and, possibly, introduces additional errors. Given the experimental results, we believe that there is a relation between the performance of EE-KPC and the length of the input. This might be mitigated by increasing the weight of the event features. Another aspect that influences the results is the performance of the classifier. In our experiments, we gave preference to recall, which maximized the number of sentences/SUs containing events.

In the future, we plan to adapt our single-document event-based summarization to explore multi-document summarization, a much more complex task than single-document summarization. The complexity comes from the inevitable large diversity of events within a large set of documents, and it also comes from redundant information.

Acknowledgements We thank anonymous reviewers for their very useful comments and suggestions. This work was supported by national funds through FCT under Project UID/CEC/50021/2013, the Carnegie Mellon Portugal Program, and Grant SFRH/BD/33769/2009.

References

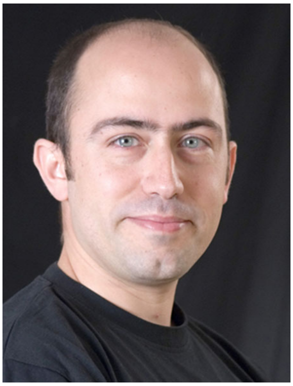
1. Abbasi A, Chen H (2008) Writeprints: a stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Trans. Inf. Syst. 26(2):7:1–7:29
2. Allan J, Carbonell J, Doddington G, Yamron J, Yang Y, Archibald B, Scudder M (1998) Topic detection and tracking pilot study final report. In: Proceedings of the broadcast news transcription and understanding workshop
3. Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J (2008) Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on management of data, SIGMOD ’08. ACM, New York pp 1247–1250

4. Carbonell J, Goldstein J (1998) The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: SIGIR '98: proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval. ACM, New York, pp 335–336
5. Carbonell J, Yang Y, Lafferty J, Brown RD, Pierce T, Liu X (1998) CMU approach to TDT: segmentation, detection, and tracking. In: Proceedings of the DARPA broadcast news conference
6. Chakrabarti D, Punera K (2011) Event summarization using tweets. In: Proceedings of the 5th international conference on weblogs and social media (ICWSM)
7. Daniel N, Radev D, Allison T (2003) Sub-event based multi-document summarization. In: Proceedings of the HLT-NAACL 03 on text summarization workshop-Vol 5, HLT-NAACL-DUC '03. Association for Computational Linguistics, Stroudsburg, pp 9–16
8. Duan Y, Chen Z, Wei F, Zhou M, Shum H (2012) Twitter topic summarization by ranking tweets using social influence and content quality, In: COLING 2012, 24th international conference on computational linguistics, proceedings of the conference: technical papers, 8–15 December 2012, pp 763–780
9. Erkan G, Radev DR (2004) LexRank: graph-based centrality as salience in text summarization. J Artif Intell Res 22:457–479
10. Feng A, Allan J (2007) Finding and linking incidents in news. In: CIKM '07: proceedings of the 16th ACM conference on information and knowledge management. ACM, New York, pp 821–830
11. Filatova E, Hatzivassiloglou V (2004) Event-based extractive summarization. In: Proceedings of ACL workshop on summarization, pp 104–111
12. Glavaš G, Šnajder J (2014) Event graphs for information retrieval and multi-document summarization. Expert Syst Appl 41(15):6904–6916
13. Homem N, Carvalho JP (2011) Authorship identification and author fuzzy "fingerprints". In: Proceedings of 2011 annual meeting of the North American fuzzy information processing society (NAFIPS). IEEE pp 1–6
14. Hong Y, Zhang J, Ma B, Yao J, Zhou G, Zhu Q (2011) Using cross-entity inference to improve event extraction. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies—vol 1, HLT '11. Association for Computational Linguistics, Stroudsburg, pp 1127–1136
15. Huang X, Wan X, Xiao J (2014) Comparative news summarization using concept-based optimization. Knowl Inf Syst 38(3):691–716
16. Ji H, Grishman R (2011) Knowledge base population: successful approaches and challenges. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies—vol 1', HLT '11. Association for Computational Linguistics, Stroudsburg, pp 1148–1158
17. Li W, Wu M, Lu Q, Xu W, Yuan C (2006) Extractive summarization using inter- and intra-event relevance. In: ACL 2006, 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics, proceedings of the conference, Sydney, Australia, 17–21 July 2006. Association for Computational Linguistics, Stroudsburg, pp 369–376
18. Liao S, Grishman R (2010) Using document level cross-event inference to improve event extraction. In: Proceedings of the 48th annual meeting of the association for computational linguistics, ACL '10. Association for Computational Linguistics, Stroudsburg, pp 789–797
19. Lin C-Y (2004) ROUGE: a package for automatic evaluation of summaries. In: Text summarization branches out: proceedings of the ACL-04 workshop. Association for Computational Linguistics, pp 74–81
20. Litvak M, Last M (2008) Graph-based keyword extraction for single-document summarization. In: Proceedings of the workshop on MMIES', MMIES '08. Association for Computational Linguistics, Stroudsburg pp 17–24
21. Liu M, Li W, Wu M, Lu Q (2007) Extractive summarization based on event term clustering. In: Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions', ACL '07. Association for Computational Linguistics, Stroudsburg, pp 185–188
22. Marujo L, Carvalho JP, Gershman A, Carbonell J, Neto JP, de Matos DM (2015) Textual event detection using fuzzy fingerprints. In: Angelov P, Atanassov K, Doukovska L, Hadjiski M, Jotsov V, Kacprzyk J, Kasabov N, Sotirov S, Szmidt E, Zadrożny S (eds) Intelligent systems' 2014, vol 322 of advances in intelligent systems and computing. Springer, Berlin, pp 825–836
23. Marujo L, Gershman A, Carbonell J, Frederking R, Neto JP (2012) Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. In: Proceedings of the 8th language resources and evaluation conference (LREC 2012), ELRA
24. Marujo L, Portelo J, Martins de Matos D, Neto JP, Gershman A, Carbonell J, Trancoso I, Raj B (2014) Privacy-preserving important passage retrieval. In: Proceedings of the 1st international workshop on privacy-preserving IR: when information retrieval meets privacy and security co-located with 37th annual international ACM SIGIR conference (SIGIR 2014). CEUR, pp 7–12

25. Marujo L, Viveiros M, Neto JP (2011) Keyphrase cloud generation of broadcast news. In: Proceeding of interspeech 2011: 12th annual conference of the international speech communication association, ISCA
26. Maskey SR (2008) Automatic broadcast news speech summarization. Ph.D. thesis, Columbia University
27. Maskey SR, Hirschberg J (2005) Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. In: Proceedings of the 9th EUROSPEECH—INTERSPEECH 2005
28. Mei J-P, Chen L (2012) Sumcr: a new subtopic-based extractive approach for text summarization. Knowl Inf Syst 31(3):527–545
29. Nallapati R, Feng A, Peng F, Allan J (2004) Event threading within news topics. In: CIKM '04: Proceedings of the 13th ACM international conference on information and knowledge management. ACM, New York, pp 446–453
30. Naughton M, Stokes N, Carthy J (2008) Investigating statistical techniques for sentence-level event classification. In: Proceedings of the 22nd international conference on computational linguistics—vol 1, COLING '08. Association for Computational Linguistics, Stroudsburg, pp 617–624
31. Nichols J, Mahmud J, Drews C (2012) Summarizing sporting events using twitter. In: Proceedings of the 2012 ACM international conference on intelligent user interfaces, IUI '12. ACM, New York, pp 189–198
32. Olariu A (2014) Efficient online summarization of microblogging streams. In: Proceedings of the 14th conference of the European chapter of the association for computational linguistics, vol 2: short papers. Association for Computational Linguistics, Gothenburg, pp 236–240
33. Ribeiro R, de Matos DM (2011) Revisiting centrality-as-relevance: support sets and similarity as geometric proximity. J Artif Intell Res 42:275–308
34. Ribeiro R, Marujo L, Martins de Matos D, Neto JP, Gershman A, Carbonell J (2013) Self reinforcement for important passage retrieval. In: SIGIR '13: proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval. ACM, New York, pp 845–848
35. Riedhammer K, Favre B, Hakkani-Tür D (2010) Long story short—global unsupervised models for keyphrase based meeting summarization. Speech Commun 52:801–815
36. Rosa H, Batista F, Carvalho JP (2014) Twitter topic fuzzy fingerprints. In: 2014 IEEE international conference on fuzzy systems (FUZZ-IEEE). IEEE, pp 776–783
37. Rubin TN, Chambers A, Smyth P, Steyvers M (2012) Statistical topic models for multi-label document classification. Mach Learn 88(1–2):157–208
38. Saggion H, Szasz S (2012) The CONCISUS corpus of event summaries. In: Proceedings of the 8th language resources and evaluation conference (LREC 2012), ELRA
39. Sharifi B, Hutton M-A, Kalita J (2010) Summarizing microblogs automatically. In: Human language technologies: the 2010 annual conference of the North American chapter of the association for computational linguistics, HLT '10. Association for Computational Linguistics, Stroudsburg, pp 685–688
40. Shou L, Wang Z, Chen K, Chen G (2013) Sumblr: continuous summarization of evolving tweet streams. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, SIGIR '13. ACM, New York, pp 533–542
41. Sipos R, Swaminathan A, Shivaswamy P, Joachims T (2012) Temporal corpus summarization using submodular word coverage. In: CIKM '12: proceedings of the 21st ACM international conference on information and knowledge management. ACM, New York, pp 754–763
42. Takamura H, Yokono H, Okumura M (2011) Summarizing a document stream. In: Proceedings of the 33rd European conference on advances in information retrieval, ECIR'11. Springer, Berlin, pp 177–188
43. Tucker RI, Spärck Jones K (2005) Between shallow and deep: an experiment in automatic summarising. Technical report 632, University of Cambridge
44. Uysal I, Croft WB (2011) User oriented tweet ranking: a filtering approach to microblogs. In: Proceedings of the 20th ACM international conference on information and knowledge management, CIKM '11. ACM, New York, pp 2261–2264
45. Uzêda V, Pardo T, Nunes M (2010) A comprehensive comparative evaluation of RST-based summarization methods. ACM Trans Speech Lang Process (TSLP) 6(4):1–20
46. Vanderwende L, Suzuki H, Brockett C, Nenkova A (2007) Beyond SumBasic: task-focused summarization and lexical expansion. Inf Process Manag 43:1606–1618
47. Walker C, Strassel S, Medero J (2006) ACE 2005 multilingual training corpus. Linguistic Data Consortium, Philadelphia
48. Wan X, Yang J, Xiao J (2007) Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In: Proceedings of the 45th annual meeting of the association for computational linguistics (ACL 2007). Association for Computational Linguistics Prague, pp 552–559
49. Yang Y, Carbonell JG, Brown RD, Pierce T, Archibald BT, Liu X (1999) Learning approaches for detecting and tracking news events. IEEE Intell Syst 14(4):32–43

50. Yang Y, Liu X (1999) A re-examination of text categorization methods. In: SIGIR' 99: proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval. ACM, New York, pp 42–49
51. Yang Y, Pierce T, Carbonell J (1998) A study of retrospective and on-line event detection. In: SIGIR '98: proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval. ACM, New York, pp 28–36
52. Zechner K, Waibel A (2000) Minimizing word error rate in textual summaries of spoken language. In: Proceedings of the 1st North American chapter of the association for computational linguistics conference, Morgan Kaufmann, pp 186–193
53. Zha H (2002) Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In: SIGIR '02: proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval. ACM, New York pp 113–120

Luís Marujo finished his dual-degree Ph.D. in Language Technologies in 2015 from Carnegie Mellon University (CMU) and the Instituto Superior Tcnico (IST), Technical University of Lisbon, Portugal. He also obtained a MSc. (2012) in Language Technologies from CMU. He holds MSc. (2009) and BSc. (2007) in Computer Science and Engineering from IST. He was awarded the best poster award entitled Keyphrase Cloud Generation of Broadcast News at the S3MR 2011. His research interests include Natural Language Processing, Information Retrieval, Machine Learning, and Machine Translation. In the past, he participated in several European and National projects and has published 18 research papers on international conferences and journals. He is a member of the Order of Portuguese Engineers.
Currently, he is working as Data Scientist at Feedzai Research in Lisbon, Portugal.



Ricardo Ribeiro has a Ph.D. (2011) in Information Systems and Computer Engineering and an MSc (2003) in Electrical and Computer Engineering, both from Instituto Superior Técnico, and a graduation degree (1996) in Mathematics/Computer Science from Universidade da Beira Interior. From 1999 to 2011, he was a Lecturer at ISCTE-Instituto Universitário de Lisboa, where he is now an Assistant Professor and the Director of the Computer Engineering undergraduate programme. Since 1996, he is a researcher at INESC-ID working in speech and language processing, being a member of the Spoken Language systems Laboratory since its creation (2001). His current research interests focus on the extraction of high-level information from unrestricted text or speech and on improving machine-learning techniques using domain-related information. He has participated in several European and nationally funded projects, organized and was a member of the program committee of several scientific events, and edited a book on the processing of Portuguese.

Anatole Gershman has been a Distinguished Career Professor of Computer Science at Carnegie Mellon University since April 2007 after his retirement from the position of Chief Scientist and Global Director of Research at Accenture Technology Laboratories. His research interests are focused on the applications of Artificial Intelligence techniques to solving information-intensive real-world problems. During his 17 years of leadership, research at the Accenture laboratories was dedicated to innovation in services.
Prior to joining Accenture in 1989, Anatole spent over 15 years conducting research and building commercial systems based on Artificial Intelligence and NLP. He held R&D positions at Coopers & Lybrand, Cognitive Systems, Inc., Schlumberger, and Bell Laboratories.
Anatole studied Mathematics and Computer Science at Moscow State Pedagogical University and received his Ph.D. in Computer Science from Yale University in 1979. He authored over 60 papers and holds 12 US patents.

David Martins de Matos graduated in Electrical and Computer Engineering in 1990 from Instituto Superior Tcnico (IST), Lisbon. He received a master's degree in Electrical and Computer Engineering in 1995 (from IST), on object-oriented programming in distributed systems. In 2005, he received a Doctor of Engineering Degree in Systems and Computer Science (IST), on automatic natural language generation. He has been teaching since 1993. He was a researcher at INESC, from 1988 to 2000, in the Distributed Systems, Telematic Services and Systems, and Software Engineering Groups. In 1998, he became a member of the Telematic and Computational Systems Center (IST), until 2001, when he became a member of INESC-ID's newly created Spoken Language Systems Lab (L2F). In the past, he participated in several European and National projects. He is a member of the ACM, the Order of Portuguese Engineers. He is a senior member of the IEEE.

João P. Neto received his graduation, MSc., and Ph.D. in Electrotechnical and Computers Engineering from the Instituto Superior Tcnico, Technical University of Lisbon, in 1987, 1991, and 1998, respectively. He has been teaching at this University since 1991 where he is currently Assistant Professor, teaching signal processing courses. He is a researcher at INESC ID Lisbon since 1987 and was one of the co-founders of the Spoken Language Systems Laboratory (L2F) in 2000. In 2008, he was a co-founder of VOICEINTERACTION, a start-up company from INESC-ID and IST where is the CEO. His Ph.D. thesis was on speaker adaptation in a context of hybrid artificial neural networks and HMM continuous ASR systems with applications on broadcast news speech and embodied conversational agents. He has participated in several European and National projects and published more than 70 papers on international conferences and magazines. He is a member of IEEE and ISCA.

Jaime Carbonell is University Professor and Allan Newell Professor of Computer Science. He joined the Carnegie Mellon community as an assistant professor in 1979 and is a widely recognized authority in MT, NLP, and machine learning. Carbonell has invented several famous algorithms, including proactive machine learning and MMR. His research resulted in multiple companies, including Carnegie Speech, Carnegie Group, and Dynamix Technologies.
Carbonell created the university's Ph.D. program in language technologies and is co-creator of the Universal Library and its Million Book Project. He founded CMU's Center for MT in 1986 and led its transformation in 1996 into the Language Technologies Institute, which he currently directs. He has advised more than 55 Ph.D. students and authored more than 350 research papers.
Before joining the CMU faculty, Carbonell earned bachelor's degrees in mathematics and physics at the Massachusetts Institute of Technology and his master's degree and Ph.D. in computer science at Yale University.