Topic detection in multichannel Italian newspapers*

Laura Po[1], Federica Rollo[1], and Raquel Trillo Lado[2]

[1] Dipartimento di Ingegneria "Enzo Ferrari" - Università di Modena e Reggio Emilia - Italy
[2] Departamento de Informática e Ingeniería de Sistemas - Universidad de Zaragoza - Spain
laura.po@unimore.it, federica.rollo@unimore.it, and raqueltl@unizar.es

Abstract. Nowadays, any person, company or public institution uses and exploits different channels to share private or public information with other people (friends, customers, relatives, etc.) or institutions. This context has changed the journalism, thus, the major newspapers report news not just on its own web site, but also on several social media such as Twitter or YouTube. The use of multiple communication media stimulates the need for integration and analysis of the content published globally and not just at the level of a single medium. An analysis to achieve a comprehensive overview of the information that reaches the end users and how they consume the information is needed. This analysis should identify the main topics in the news flow and reveal the mechanisms of publication of news on different media (e.g. news timeline). Currently, most of the work on this area is still focused on a single medium. So, an analysis across different media (channels) should improve the result of topic detection. This paper shows the application of a graph analytical approach, called Keygraph, to a set of very heterogeneous documents such as the news published on various media. A preliminary evaluation on the news published in a 5 days period was able to identify the main topics within the publications of a single newspaper, and also within the publications of 20 newspapers on several on-line channels.

Keywords: clustering, topic detection, news cycle, news tracking, cross-channel publication, social media

1 Introduction

In the last decade, more and more newspapers have begun using the Internet as a tool for spreading news. Printed newspapers continue to be used but editors are increasingly distributing their newspapers' content over several delivery channels on the Internet due to improved timeliness. In most cases, they are re-purposing content from the printed editions in various electronic editions, on their web sites and on social media [10]. Thus, according to ISTAT research[3], the percentage of Italians reading newspapers on-line was 11% of the total of Italian people using the Internet, while in 2014 this percentage rose to 32.2%. France and Poland have similar rates, while Finland and Sweden have more digital readers (80%).

Recently, social networks have gained a very important role in the dissemination of news, because they allow a greater share of news than websites and are more timely

* The research presented in this paper was partially funded by Keystone Action COST IC1302.

[3] ISTAT `http://tinyurl.com/jc5sfc8`

to provide updates. In fact, it is very common that a newspaper publishes several updated versions of the same piece of news on the same day. Therefore, as well as printed newspapers and their websites, most journals also use social networks, specially Facebook and Twitter. So, it is interesting to delineate how and how much Web and social networks are used to disseminate news content.

The goal of this paper is to analyze published news to determine whether there exist correlations among news published by different newspapers on different channels. In order to do that, we have adapted the Keygraph algorithm [8] for topic detection on multiple communications media. The idea was to devise a new approach that can examine all the contents coming from different media (currently: web sites, Facebook, Twitter) instead of considering an analysis focused on a single media. Besides, we have performed a preliminary evaluation on a 5 days period of the news collected on December 2015, that were published on-line by the 20 most popular Italian newspapers.

Clustering news published on different communication media is difficult since different styles are used in different media or channels. Thus, the news reported on the web sites usually contain several phrases, a title (usually short) and a long description, while the posts about the same pieces of news on social media contain few words and other kinds of information such as hashtags and links. On the one hand, we can exploit the implicit information encoded in the hashtags and in the links toward other pieces of news. On the other hand, the preprocessing and the clustering techniques need to be configured and modified based on the input text.

This paper is organized as follows. In the next section, we review related work, Section 3 presents the Keygraph topic detection algorithm and the modifications introduced to harmonize the topic detection when applied on heterogeneous sources. Then, in Section 4, we test the impact and accuracy of Keygraph on Italian newspaper publications. Finally, we summarize our research and highlight some future directions in Section 5.

2 Related Work

Detecting the main topics or events of a set of news is related to the creation of summaries of documents or text summarization, as a set of keywords contains the main ideas of the news. In this area, classical approaches use statistical criteria to detect sentences that contain high-frequency terms or the position of the words in the sentences and in the document (title, contained, etc.) [6]. In contrast, the adaptation of Keygraph trusts on the creation of a graph by considering the correlation of the words in the documents to identify the main events. With the explosion of social networks such as Facebook and Twitter, techniques for event detection were adapted to consider streams of shorter documents (entries) produced with a higher frequency (hours or minutes vs days) [2]. Moreover, new challenges arise: dealing with a higher level of grammatical errors, incorrect spelling, etc. [2].

The topics or events extracted from the different collections are usually used to characterize the items of the collection and make recommendations to users. Thus, TMR [5] is an semantic recommender system that takes as input a Topic Map generated by TM-Gen [5] and a profile of a user and outputs a list of items that the user could be interested in. The adapted version of Keygraph described in this paper has the same purpose as

TM-Gen, i.e., extracting information from a set of pieces of news and representing them as a Topic Map. Nevertheless, our proposal not only considers the articles in the website of the newspaper as TM-Gen but it also takes into account the entries published in Twitter and Facebook related to those articles. Moreover, TM-Gen only considers an information source (the news published by the Spanish newspaper "El Heraldo de Aragón") for performing experiments, while the proposal described in this paper has integrated the information from different media companies (20 Italian newspaper companies).

3 Keygraph adapted for the multichannel analysis

The aim of this paper is to carry out an analysis on the news published by the main Italian newspapers: clustering the news around the main topics to understand correlations and make comparisons between news published on different channels and different newspapers. The analysis of news has been made using the Keygraph algorithm for automatic indexing of documents. Keygraph explicitly incorporates word co-occurrence in topic modeling and it has been demonstrated to have scalable and good performances, similar to that of topic modeling solutions (such as GAC and LDA-GS), on a large noisy social media dataset [8].

An event is "a specific thing that happens at a specific time and place" [1]. It may be composed of many sub-events, each of them at a finer level of granularity. For example, the event of the Spanish election occurred in December 2015 covers a broad range of topics: the voter turnout, the announcement of the winner, the reaction of the winner and the opposition, the risk of ungovernability, etc. All of these are sub-events related to the Spanish election event. News or events can be described by a set of terms, representing the asserted main point in the document. Documents describing the same event usually contain similar sets of keywords. Therefore, in order to detect the topic of the news and to cluster similar news, it is crucial to extract meaningful keywords and to discharge unessential words from news text.

Keygraph [8] is an algorithm based on the segmentation of a graph, whose goal is to identify events and clusters around events. Keygraph applies a community detection algorithm to group co-occurring keywords into communities. Each community is formed by a constellation of keywords that represents a topic. The similarity between a community and a document is computed to rank similar documents. The original Keygraph algorithm[4]) was modified to improve the results of indexing, giving consideration to the hashtags and URLs in news text. The algorithm uses a configuration file that is provided as input and contains numerical parameters useful for clustering (the upcoming words written in italics refer to configuration parameters).

In the following, the original Keygraph algorithm phases and the modifications that have been introduced in order to deal with heterogeneous documents (news published on different channels) are described.

[4] The code of the version 2.2 of March 2014 is available on-line at `Keygraph.codeplex.com`.

3.1 Building the Keygraph

The first phase focuses on extracting keywords from documents, which represent pieces of news, and building a graph considering the co-occurrence of keywords.

The body of a document (content, text describing the piece of news) is the principal component and it is analyzed to extract keywords: each word of the body is stemmed and is considered if and only if it does not appear in a stop-word list (a list of very commonly used words irrelevant for searching purposes). Each keyword k_i is characterized by a base form, that is the root of the word (the result of the stemmer), its term frequency TF (how many times the keyword appears in the document), its document frequency DF (how many times the keyword appears in all documents) and the inverse of its document frequency IDF. The TF is initialized to *text-weight* value, given as input in the configuration. At this stage, each document is represented by a set of keywords. Documents that have less than *doc_keywords_size_min* keywords are removed.

After that, a node n_i is created for each unique keyword in the dataset. Nodes with low DF or high DF are filtered. An edge $e_{i,j}$ between nodes n_i and n_j is added if k_i and k_j co-occur in the same document. Edges are weighted by how many times the keywords co-occur (DF document frequency of the edge). Edges linking keywords that co-occur below some minimum threshold (*edge_df_min*) are removed. Edges linking keywords that almost always appear together are also removed.

For each remaining edge, conditional probabilities (CP) $p(k_i|k_j)$ and $p(k_j|k_i)$ are computed. For $e_{i,j}$, the conditional probability of the occurrence $p(k_i|k_j)$ is the probability of seeing k_i in a document if k_j exists in the document. The conditional probability is computed in the following way:

$$p(k_i|k_j) = \frac{DF_{i \cap j}}{DF_j} = \frac{DF_{e_{i,j}}}{DF_j}$$

Finally, nodes without edges are removed.

3.2 Extracting topic features

The second stage involves the extraction of communities within the Keygraph created.

The graph appears as a network of interconnected keywords; here some nodes have a stronger connection with others. These groups of interconnected nodes, called connected components, are identified. Each connected component must contain a number of nodes between *cluster_node_size_min* and *cluster_node_size_max*. If the number of nodes is greater than the threshold, the edges with low CP are deleted.

Within these groups, the communities need to be identified. A useful measure for this purpose is the betweenness centrality, an indicator of a node's centrality in a network. The betweenness centrality of an edge is defined as the number of shortest paths for all pairs of nodes of the network that pass through that edge. Of course the edges that connect different communities have a very high betweenness centrality score, since the shortest routes connecting pairs of nodes of different communities will have to pass necessarily by those arcs (edges).

In each connected component it is necessary to identify the edge with the highest betweenness centrality score, through a breadth-first search. This edge is removed from the graph. If two edges have the same score of betweenness centrality, the one with lower DF is removed. Before removing the edge, its value of conditional probabilities is considered. If the CP of the edge is above the threshold *edge_cp_min_to_duplicate*, then the edge and its corresponding nodes are duplicated. In this way, a node might occur in more than one community. This process is repeated until there is no edge with a high betweenness centrality score. After removing all the edges that interconnect different communities, we identify for each community a topic. The topic is characterized by the keywords of the community (the feature vector f_t). So, each community can be seen as a particular document. The documents similar to this "community document" can be clustered together, creating a document cluster.

3.3 Assigning topics to documents

The probability that a topic t is associated with a document d is calculated by considering the cosine similarity of d with respect to the feature vector f_t, as follows:

$$p(t|d) = \frac{cosine(d, f_t)}{\sum_{t' \in T} cosine(d, f_{t'})}$$

The weight of each keyword of the feature vector f_t is calculated using the TF-IDF function. This function increases proportionally to the number of times that the word is used in the document, but grows in inverse proportion with the frequency of the term in the collection. So, it gives more importance to the terms that frequently appear in the document, but are quite rare in the collection [7]. The TF-IDF function can be decomposed in two factors: $tf_{i,j}$ and idf_i. $tf_{i,j}$ is the number of occurrences of the term t_i in the document d_j; while idf_i represents the overall importance of the word in the collection and is calculated as follows:

$$idf_i = \frac{1}{ln2} ln\left(\frac{|D|}{DF_i}\right)$$

where $|D|$ is the number of documents in the collection and DF_i is the total number of occurrences of the term t in all documents. The sum of the TF-IDF functions calculated for each node in a community is called *vector size*. The vector size is calculated for the community, for the document and for the set of keywords that are shared between the community and the document. The cosine similarity is the ratio between the latter and the product of the first two. For each document the cosine similarity is computed with respect to each community and its value is compared with the *doc_sim2Keygraph_min* threshold: if the similarity between a document and a topic is greater than this parameter, the topic is assigned to the document. So, similar documents form clusters.

A document may be assigned to multiple topics, unless "hard clustering" (forcing the assignment of a document to the topic with the greatest cosine similarity) is specified. If no "hard clustering" is performed, there may be a significant overlap between the sets of documents in different document clusters. So, a merging operation is performed if the following equation is verified:

$$\frac{intersect}{min(|DC1|, |DC2|)} \geq cluster\_intersect\_min$$

where intersect is the number of common documents and $|DC1|$ and $|DC2|$ are the number of documents that are part of the first and the second document cluster. After this final step, the algorithm created a set of document clusters. Documents within the same cluster are about the same topic.

3.4 Modification to the algorithm

Initially, minor changes were made to cluster news coming from different channels. These minor changes regard increasing the importance of hashtags, deleting mentions and names of authors from the news text, and defining proper configuration parameters. Hashtags, textual tokens prefixed by hash marks (#), are very useful for our purpose, since they are used as proxies for topics. Each news post can contain several hashtags (especially Twitter posts). In Keygraph, hashtags are extracted, splitted in a list of words, and added to the set of keywords describing the document (if they are not considered stopwords). The hashtag segmentation regards finding the best way of splitting an input string into words. In literature, empirical methods and supervised or unsupervised techniques based on multiple corpora are available for word segmentation. Posts about the same topic may use different, but still similar hashtags. The hashtag segmentation is useful to identify the correlation between these posts. For example, two posts the first containing "#expo2015" and the second "#expoMilano" share the keyword "expo".

In addition to these changes, a major improvement was the implementation of a mechanism that would allow the consideration of link between different documents. Each post can contain one or more URLs, i.e. links to external resources. Generally, posts on Twitter and Facebook contain two links: one connects to the news on the social network and the other leads to the web site page of the newspaper in which the same news is published. A URL can also be used to link to a previous version of the news regarding the same topic. Moreover, some URLs are links to multimedia contents (such as Youtube videos). In a few cases, URLs are links to "general" pages. This kind of URLs do not connect to a specific news thus we will not consider them. The important point for our purposes is that posts that share a link are strongly correlated, thus we can suppose they are about the same topic. So, we know in advance that these posts should be part of the same document cluster.

All the modifications introduced have led to the implementation of three extended versions of Keygraph, that are variants of the original algorithm [8]:

- Keygraph 1 is a variant of Keygraph that eliminates the authors' names and mentions from the news posts, extracts and analyses the hashtags, uses three different configuration parameters according to each publication channel; and increases the weight of hashtags by doubling their term frequencies.
- Keygraph 2 is a variant of Keygraph 1 that creates arcs and, if necessary, nodes for each keyword of each pair of news with at least one link in common; and doubles the weight of these arcs.
- Keygraph 3 is a variant of Keygraph 2 that, in the final phase of the document clusters creation, forces news that shared a link to appear in the same cluster, i.e., it adds in each cluster news that share at least one link with other news in the cluster.

Newspaper	Paper editions	Digital editions	Total circulation
Corriere della Sera	368 981	95 447	464 428
Repubblica (La)	323 525	58 709	382 234
Sole 24 Ore (Il)	200 155	115 366	315 521
Gazzetta dello Sport (La)	203 516	21 042	224 558
Stampa (La)	214 461	7 198	221 659
Messaggero (Il)	137 678	4 510	142 188
QN – Il Resto del Carlino	122 513	1 234	123 747
Corriere dello Sport - Stadio	121 128	1 272	122 400
Giornale (Il)	103 658	2 115	105 773
Avvenire (L')	103 985	1 578	105 563
QN – La Nazione	98 812	1 094	99 906
Tuttosport	94 970	818	95 788
Libero	75 301	886	76 187
Italia Oggi	54 166	18 157	72 323
Gazzettino (Il)	66 163	4 276	70 439
Fatto Quotidiano (Il)	50 763	13 621	64 384
Secolo XIX (Il)	57 068	1 208	58 276
Tirreno (Il)	56 639	1 539	58 178
Mattino (Il)	50 946	2 429	53 375
QN – Il Giorno	50 597	232	50 829

Fig. 1. The list of the 20 most popular Italian newspapers.

4 Evaluation

The algorithm has been tested on the posts published in different time slots on three channels (website, Facebook and Twitter) by 21 italian newspapers: the 20 most popular Italian newspapers and the Italian multimedia information agency (Agenzia Nazionale Stampa Associata - ANSA[5]-). The list of the newspapers in daily periodicity considered in the experiments and their circulation are shown in Figure 1[6]. The circulation of a newspaper is the number of copies it distributes on average per day. Circulation could be greater than the number of copies sold, since some newspapers are distributed without cost to the readers. The number of readers is usually higher than the circulation because of the assumption that a copy of a newspaper is read by more than one person.

Considering the set of news published between 20th and 22th December 2015 by all the newspapers on the three channels (11423 news in total), we performed several tests to find the best configuration parameters for Keygraph. For example, decreasing the similarity threshold between a document and the community, more documents are clustered together, but the precision value obtained for the algorithm also decreases. We tested two values of the *doc_sim2Keygraph_min* threshold: 0.18 and 0.30. With a 0.18 threshold, we increase the percentage of documents clustered (from 15% to 27%) but the precision decreases (from 75% to 63%). With a 0.30 value, the percentage of documents decreases (from 14% to 8%) but the precision increases (from 83% to 94%). Changes on other parameters also affect the values of precision, accuracy and recall. Figure 2 shows the configuration parameters that we considered the best. These parameters were used in the two tests that have been conducted for the three versions of Keygraph shown in sections 4.3 and 4.4.

[5] `http://www.ansa.it/`

[6] The average circulations of each newspaper refer to February 2015 as reported by the Italian Federation of Newspaper Publishers (Federazione Italiana Editori Giornali available at `http://www.fieg.it`).

	Site	Facebook	Twitter
TEXT_WEIGHT	0.6	0.8	1
KEYWORDS_WEIGHT	1		
HASHTAG_WEIGHT	2		
HARD_CLUSTERING	false		
NODE_DF_MIN	2		
NODE_DF_MAX*	0.04		
EDGE_CORRELATION_MIN*	0.03		
EDGE_DF_MIN	3		
DOC_KEYWORDS_SIZE_MIN	3	2	2
DOC_SIM2KEYGRAPH_MIN	0.3		
CLUSTER_NODE_SIZE_MAX	1000		
CLUSTER_NODE_SIZE_MIN	2		
CLUSTER_INTERSECT_MIN	0.65		
TOPIC_MIN_SIZE	2		
EDGE_CP_MIN_TO_DUPLICATE	1		
CLUSTERING_ALG	betweenness		

Fig. 2. Configuration parameters.

4.1 Performance measures

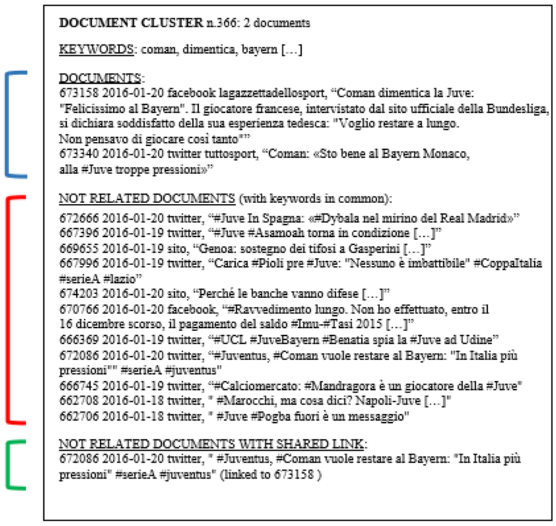
The evaluation of Keygraph has been conducted by manually identifying true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) within every document cluster. After that, the following performance measures were calculated:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}; precision = \frac{TP}{TP+FP}; recall = \frac{TP}{TP+FN};$$

TP and FP are evaluated on the list of documents within the document cluster. While FP and FN has to be judged on documents that have not been selected by the algorithm to be part of the cluster. Since it is not feasible to evaluate all the documents that are not selected within a document cluster, we identified two possibilities to retrieve a reasonable number of documents: 1) to recover documents that shared a reasonable number of keywords with the document cluster[7], and 2) to retrieve documents that shared at least one URL link with one document in the cluster.

A clarifying example is shown in Figure 3. The blue color indicates the news that are selected by Keygraph to be part of the cluster. The cluster is described by the keywords: “coman”, “bayern”, “dimentica” (meaning “forgot”), etc. The topic is about the new engagement of the player Coman in the Bayern Monaco team. The two clustered documents are true positives, and no false positives are detected. The red part identifies the news that share a reasonable number of keywords with the set of keywords of the cluster. Here, we can detect that the news with ID number “672086” is related with the topic of the cluster, so this news is a classify as a FN, while all the other news are TN. The green color indicates the news that share at least one URL link with the other documents in the cluster. The detected FN refers to the same news “672086”. Note that false negatives are considered two, even if the detected news is the same. In this small example, we got $TP = 2$, $FP = 0$, $FN = 2$, $TN = 10$.

[7] As the content available on web sites and Facebook news is greater than the content on Twitter news, the number of shared keywords is different according to the channel: 5 if news is published on a Website or on Facebook, 3 if news is published on Twitter.


DOCUMENT CLUSTER n.366: 2 documents

KEYWORDS: coman, dimentica, bayern […]

DOCUMENTS:
673158 2016-01-20 facebook lagazzettadellosport, "Coman dimentica la Juve: "Felicissimo al Bayern". Il giocatore francese, intervistato dal sito ufficiale della Bundesliga, si dichiara soddisfatto della sua esperienza tedesca: "Voglio restare a lungo. Non pensavo di giocare così tanto""
673340 2016-01-20 twitter tuttosport, "Coman: «Sto bene al Bayern Monaco, alla #Juve troppe pressioni»"

NOT RELATED DOCUMENTS (with keywords in common):
672666 2016-01-20 twitter, "#Juve In Spagna: «#Dybala nel mirino del Real Madrid»"
667396 2016-01-19 twitter, "#Juve #Asamoah torna in condizione […]"
669655 2016-01-19 sito, "Genoa: sostegno dei tifosi a Gasperini […]"
667996 2016-01-19 twitter, "Carica #Pioli pre #Juve: "Nessuno è imbattibile" #CoppaItalia #serieA #lazio"
674203 2016-01-20 sito, "Perché le banche vanno difese […]"
670766 2016-01-20 facebook, "#Ravvedimento lungo. Non ho effettuato, entro il 16 dicembre scorso, il pagamento del saldo #Imu-#Tasi 2015 […]"
666369 2016-01-19 twitter, "#UCL #JuveBayern #Benatia spia la #Juve ad Udine"
672086 2016-01-20 twitter, "#Juventus, #Coman vuole restare al Bayern: "In Italia più pressioni"" #serieA #juventus"
666745 2016-01-19 twitter, "#Calciomercato: #Mandragora è un giocatore della #Juve"
662708 2016-01-18 twitter, " #Marocchi, ma cosa dici? Napoli-Juve […]"
662706 2016-01-18 twitter, " #Juve #Pogba fuori è un messaggio"

NOT RELATED DOCUMENTS WITH SHARED LINK:
672086 2016-01-20 twitter, " #Juventus, #Coman vuole restare al Bayern: "In Italia più pressioni" #serieA #juventus" (linked to 673158)

Fig. 3. Keygraph evaluation - an example.

4.2 Evaluation of the impact

We decide to evaluate the impact of our approach as the number of new correlations created by the algorithm once the documents clusters are built. The correlations can be divided into: correlations among news published on the same channel, correlations among news published on different channels, correlations among news published on the same newspaper, correlations among news published on different newspapers.

The total number of correlations in a document cluster is computed by the formula $Correlations = \binom{N}{2}$ where N is the number of documents in the cluster. For each cluster the number of news published on the same channel is also calculated for each channel (N_{site}, $N_{Facebook}$ and $N_{Twitter}$). If the number of news on a channel is not equal to 0 or 1, the next formulas are used to find the number of correlations between news published on the same channel:

$$Corr_{site} = \binom{N_{site}}{2}; Corr_{facebook} = \binom{N_{Facebook}}{2}; Corr_{twitter} = \binom{N_{Twitter}}{2}$$

The total number of correlations between the news published on the same channel and on different channels are: $Corr_{sameChannel} = Corr_{site} + Corr_{facebook} + Corr_{twitter}$ and $Corr_{differentChannels} = Correlations - Corr_{sameChannel}$, respectively. Moreover, the same type of calculus is adopted to find correlations among news published by one newspaper or different newspapers.

Finally, the evaluation of the impact is compared with respect to a baseline, called *link cluster*. The *link cluster* is built taking into account only the URL links contained in the news: the news that share at least a URL link are joined in the same document cluster. Thefore, the *link cluster* produces a set of document clusters in which each news share at least a URL link with another news in the same cluster.

4.3 Test 1 - multichannel publishing by a single newspaper

This test has been executed on the news published by La Repubblica on all channels. This test set contains 2430 news (125 published on the Website -5%-, 1307 on Facebook -54%- and 998 on Twitter -41%-) and 1404 links connecting the news in the test set. Besides, 112 documents share at least one link with other document.

The same configuration file was used to run the three versions of the algorithm and the results are shown in Table 1. In the first phase of the algorithm, all documents are loaded, however, 31 documents are discharged because they contain less than *doc_keywords_size_min* keywords.

	Keygraph 1	**Keygraph 2**	**Keygraph 3**
Clustered Documents	404 (16,8%)	290 (12%)	294 (12,3%)
Nodes	397	814	814
Edges	362	634	634
Communities	101	125	125
Document Clusters			
- before merging	(99)	(97)	(97)
- after merging	82	77	(77)
- after link analysis and merging			69
Documents with shared links outside Clusters	9	9	0
Performance Measures			
Accuracy	81%	85%	87%
Precision	60%	75%	73%
Recall	65%	69%	73%

Table 1. Test 1 - Clustering results and performance.

The number of clustered documents in Keygraph 2 approximately decreases a 28% with respect to Keygraph 1, while Keygraph 3 obtaines a number of clustered documents similar to Keygraph 2. The first version is able to classify more documents than the other versions, but this does not mean that the first version is the best, because we must verify that the news in the cluster are about the same event.

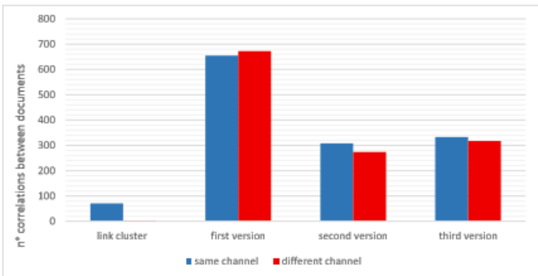


Fig. 4. Test 1 - The correlations among the published news.

In Figure 4, we represent how many correlations were found among news: the first column represents correlations in the link cluster baseline (see section 4.2 for the definition), the other columns show how many correlations are found by Keygraph 1, 2 and 3. The total number of correlations in link cluster is 75. 71 of them are among news on the same channel and only 4 among news on different channels. Nevertheless, the correlations found by Keygraph are many more than those found by using only link cluster. Keygraph 1 is able to find approximately the double of correlations than the other versions. In contrast to the results in link cluster, there is not a marked difference between same-channel correlations and different-channel correlations.

4.4 Test 2 - multichannel publishing by different newspapers

The second test has been executed on the news published by the 21 Italian newspapers on all channels. This test set contains 21457 news (5505 published on the Website -26%- 8867 on Facebook -41%-, and 7085 on Twitter -33%-) and 26063 links connecting the news in the test set is (so, on average, a news has more links). Besides, 3233 documents share at least one link with other document. All documents are loaded, but 281 are discharged because they have few keywords.

	Keygraph 1	**Keygraph 2**	**Keygraph 3**
Clustered Documents	2777 (13%)	1025 (4,8%)	1074 (5%)
Nodes	4972	9916	9916
Edges	13575	63487	63487
Communities	789	761	761
Document Clusters			
- before merging	(572)	(278)	(278)
- after merging	514	265	(265)
- after link analysis and merging			261
Documents with shared links outside Clusters	135	52	0
Performance Measures			
Accuracy	83%	88%	90%
Precision	75%	85%	86%
Recall	75%	79%	82%

Table 2. Test 2 - Clustering results and performance.

As shown in Table 2, the clustered documents in Keygraph 2 are less than the half of clustered documents in Keygraph 1, the same happens with the number of document clusters. In Keygraph 3 the clustered documents slightly increases w.r.t. Keygraph 2 and the document clusters decreases, like in the previous test.

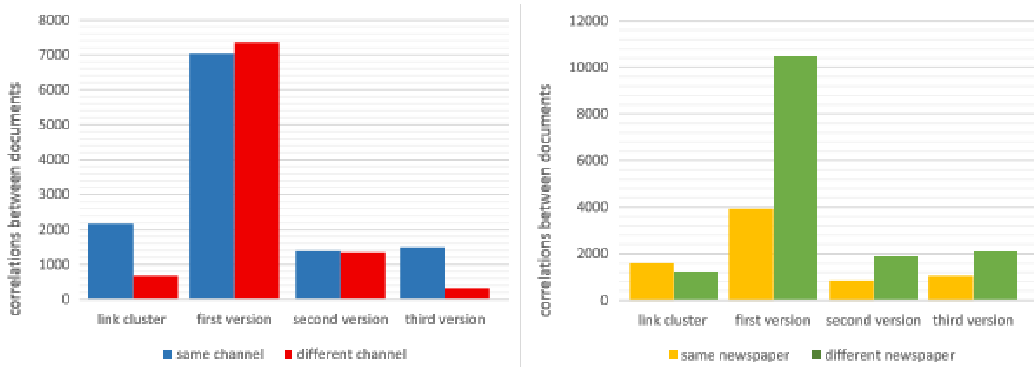


Fig. 5. The correlations among the news published by the same or different newspaper.

In Figure 5, the first columns show that in link cluster most of the correlations are between news published on the same channel or by the same newspaper. The other columns reveal that the Keygraph algorithm finds much more correlations between news: the first version finds many more correlations than the other two versions.

5 Conclusion

This paper represents an introductory analysis on the news published by the main Italian newspapers. By exploiting three extended versions of a graph analytical approach for topic detection and automatic indexing of documents, called Keygraph [8], we demonstrated how to cluster the news around the main topics to understand correlations and compare news published on different channels and different newspapers.

A preliminary evaluation of the three extended Keygraph versions on the news published in a 5 days period has shown promising results. Keygraph 3 was able to identify the main topics within the publications of a single newspaper reaching a 73% of precision and recall and also within the publications of 20 newspapers on several on-line channels reaching a 86% of precision and 82% of recall.

Future work will be focused on comparing the Keygraph algorithm w.r.t. other topic models such as LSA or LDA [4]. Moreover, we would like to investigate how disambiguation techniques might improved the results of Keygraph [3,9].

References

1. J. Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.
2. F. Atefeh and W. Khreich. A survey of techniques for event detection in twitter. *Comput. Intell.*, 31(1):132–164, February 2015.
3. S. Bergamaschi, D. Beneventano, L. Po, and S. Sorrentino. Automatic normalization and annotation for discovering semantic mappings. In *Search computing*, pages 85–100. Springer, 2011.
4. S. Bergamaschi, L. Po, and S. Sorrentino. Comparing topic models for a movie recommendation system. In *WEBIST (2)*, pages 172–183, 2014.
5. A. L. Garrido, M. G. Buey, S. Escudero, S. Ilarri, E. Mena, and S. B. Silveira. TM-gen: A topic map generator from text documents. In *25th IEEE International Conference on Tools with Artificial Intelligence, Washington (USA)*. IEEE Computer Society, November 2013.
6. H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165, April 1958.
7. A. Rajaraman and J. D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, NY, USA, 2011.
8. H. Sayyadi and L. Raschid. A graph analytical approach for topic detection. *ACM Trans. Internet Technol.*, 13(2):4:1–4:23, December 2013.
9. R. Trillo, L. Po, S. Ilarri, S. Bergamaschi, and E. Mena. Using semantic techniques to access web data. *Information Systems*, 36(2):117–133, 2011.
10. A. Veglis. Cross-media publishing by us newspapers. *Journal of Electronic Publishing*, 10(2), 2007.