A comparative study on community detection and clustering algorithms for text categorisation

Michelangelo Misuraca¹, Germana Scepi², Maria Spano²

¹ DiScAG, Università della Calabria – michelangelo.misuraca@unical.it
² DiSES, Università di Napoli Federico II – germana.scepi@unina.it, maria.spano@unina.it

Abstract

One of the main tasks of Text Mining is organising a large number of unlabelled documents into a smaller set of meaningful and coherent clusters, similar with respect to their content. Clustering algorithms are usually carried on documents × terms matrices, algebraically representing each document as a vector. Nevertheless, a collection of documents can also be encoded differently, e.g. by considering a documents × documents representation. This peculiar data structure can be seen as an adjacency matrix and graphically displayed as a graph. In the framework of Network Analysis, community detection is performed on such graphs to find groups of nodes sharing common characteristics, and play similar roles. This paper aims at evaluating the use of different data structures and different grouping criteria, showing the effectiveness of the different alternatives in a text categorisation strategy. We performed a comparative study involving both classical text clustering approaches and community detection approaches, testing and discussing their performances.

Keywords: text clustering, community detection, data representation, weighting schemes, similarity measures

1. Introduction

One of the primary tasks of Text Mining is to organise a collection of documents written in natural language. Texts can express a broad and varied range of information, but this information is often difficult to process automatically due to its particular encoding. The family of techniques that allows grouping the different documents without any other knowledge than the textual content itself, having a data-driven standpoint, is known as text clustering. Clustering algorithms are usually carried on documents × terms matrices, algebraically representing the different documents belonging to the analysed collection as vectors. This representation – known as vector space model (Salton et al., 1975) – is the most common way to transform documents in structured data, disregarding the grammatical and syntactical roles of the different terms within the texts. The strategies adopted to group the documents are almost similar to those used for classical data and usually based on the optimisation of a criterion function. The two main approaches include hierarchical algorithms and partitive algorithms. Nevertheless, data can be organised differently, e.g. by considering a documents × documents matrix. This data structure can be seen as an adjacency matrix and diagrammatically depicted as a graph. Each document is then seen as a node and linked to the others when there is a similar textual content, i.e. documents share common terms. Different similarity measures can be used to express the level of proximity between linked documents. In the framework of Network Analysis, the so-called community detection allows highlighting if there are groups of nodes sharing common characteristics, and/or playing similar roles within the graph. The rationale of these techniques is then the same as the one at the base of clustering analysis.
In this paper, we want to evaluate the use of different data representations and grouping strategies, aiming at showing the effectiveness of the alternative strategies in a text categorisation context. We performed both classical clustering approaches and community detection approaches, considering the effect of different weighting schemes to express the importance of the terms used by each document and of different similarity measures to quantify the proximities between the different documents of the collection.

2. Theoretical background

There are several ways to model documents for quantitative analyses, but most of the clustering algorithms are commonly carried on a collection encoded via a bag of words (BoW) scheme. In this scheme, a document is seen as an unordered set of terms, disregarding grammatical and even syntactic roles. Let consider a corpus of \( n \) documents \( d_i \) (\( i = 1, \ldots, n \)). According to the vector space model, each \( d_i \) can be represented as a vector in the space spanned by the \( p \) terms belonging to the vocabulary of the corpus:

\[
d_i = (t_{i1}, \ldots, t_{in}, \ldots, t_{ip})
\]

where \( t_{im} \) is the importance of the \( m\)-th term in \( d_i \). This importance is usually measured by the term frequency – i.e. the number of occurrences of the terms into the document – but other different weighting schemes can also be considered (for a wider discussion on the choice of \( t_{im} \), see Balbi and Misuraca, 2005).

By juxtaposing the different document-vectors, it is possible to build a documents \( \times \) terms matrix \( T \). It is possible to derive from this data structure both a terms \( \times \) terms matrix and a documents \( \times \) documents matrix. Generally, in a terms \( \times \) terms matrix, each element is the number of times two terms co-occur in the document collection. In a documents \( \times \) documents matrix, each element is the number of terms which occur together in two different documents. In both cases, it is possible to consider more complex weights to express the relationship between couples of terms (Cheng et al., 2013) or couples of documents (Huang, 2008).

Supervised and unsupervised categorisation methods aim at grouping similar documents in distinct subsets, taking into account their content. In the first case, some prior knowledge is available, usually related to the topics embodied in the collection. The information concerning the number of groups, their peculiarities and their composition, is used in the categorisation process. An unsupervised approach, instead, aims at grouping a collection of unlabelled documents into a smaller number of meaningful categories similar in content, without any additional knowledge. In this latter case, the process is based solely on the available data. The strategies usually adopted for unsupervised text categorisation are similar to those used for classical data, both considering hierarchical and partitive algorithms. Hierarchical algorithms allow visualising the association structure among documents at different levels of granularity. One of the main consequences is that the number of clusters is not an input parameter of the algorithm, and the different solutions are sequentially nested and displayed in a tree structure. Although, hierarchical algorithms are not very scalable in the case of large collections of documents (Steinbach et al., 2000). Partitive algorithms – also known as centre-based – create a one-level solution instead, given the number \( k \) of desired clusters as input parameter. The documents initially selected as centroids of the \( k \) clusters are usually chosen randomly, but other options can be considered (see Larsen and Aone, 1999). After that, the proximity (by a suitable measure) of each other document to the \( k \) centroids is computed, and the assignment
to a cluster of a document is done looking at the highest proximity. The clustering process is repeatedly refined in order to optimise the chosen criterion function.

The terms \( \times \) terms matrix and the documents \( \times \) documents matrix can also be seen as a graph \( G=(V,E) \), where \( V \) is a finite set of nodes (or vertices) and \( E \) is a finite set of edges (or lines). Edges indicate the relationships between the nodes. If the nodes are the terms, the edges express the co-occurrence between linked terms. If the nodes are the documents, the edges represent the strength of the similarity between linked documents. In a Network Analysis framework, the task of grouping nodes sharing common characteristics, and/or playing similar roles within the graph, is usually performed by referring to community detection methods. Communities are usually thought as subgraphs densely inter-connected and sparsely connected to other parts of the network (Wasserman and Faust, 1994). From a theoretical viewpoint, communities are then not very different from clusters (Fortunato and Hric, 2016). As a consequence, there is an overlap among the scientific contributions developed in these two research areas. Clustering algorithms have been successfully used for graph data (e.g. Flake et al., 2003; Rattigan et al., 2007). Many recent works (e.g. Lim et al., 2017; Misuraca et al., 2018; Jia et al., 2018) proposed community detection methods on terms \( \times \) terms networks for identifying combinations of terms – i.e. concepts or topics – occurring in the collection of documents. Nevertheless, the possibility of using a community detection approach on the documents \( \times \) documents network for a text clustering has not been sufficiently explored. To the best of our knowledge, the papers of Mikhina and Trifalenkov (2018) and Cadot et al. (2018) are the only ones evaluating a clustering strategy based on the idea that the collection of documents can be represented as a weighted graph.

3. A comparative study

As claimed above, clustering and community detection share the same logic so that they can achieve equivalent results in a text categorisation framework. In the following, in order to test the effectiveness of the two approaches, we performed a comparative study.

3.1. Data description and pre-processing

We downloaded from Kaggle a dataset\(^1\) containing 2225 complete news articles published during 2004-2005 on the BBC website, and reporting stories categorised in five different topical areas (Greene and Cunningham, 2005). The type-token ratio of the collection is 0.031, substantiating the use of clustering techniques (Bolasco, 2013). Table 1 shows the distribution of documents over the five topics and some descriptive statistics.

<table>
<thead>
<tr>
<th>Topic</th>
<th># of doc</th>
<th>% of doc</th>
<th>Avg terms per doc</th>
<th>Avg sentences per doc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business</td>
<td>510</td>
<td>22.92</td>
<td>374.25</td>
<td>15.67</td>
</tr>
<tr>
<td>Entertainment</td>
<td>386</td>
<td>17.35</td>
<td>375.88</td>
<td>16.30</td>
</tr>
<tr>
<td>Politics</td>
<td>417</td>
<td>18.74</td>
<td>511.80</td>
<td>20.88</td>
</tr>
<tr>
<td>Sports</td>
<td>511</td>
<td>22.97</td>
<td>374.30</td>
<td>16.90</td>
</tr>
<tr>
<td>Tech</td>
<td>401</td>
<td>18.02</td>
<td>563.37</td>
<td>24.04</td>
</tr>
</tbody>
</table>

*Table 1 – Characteristics of BBC news collection.*

\(^1\) https://www.kaggle.com/pariza/bbc-news-summary
Before carrying on the analyses, we stripped numbers, dates, punctuation, and URLs from the original documents. We also normalised the documents by removing special characters and any separators than blanks. On the cleaned collection, we then performed lemmatisation and removed English stopwords. Moreover, we deleted terms occurring less than two times and kept only the terms which occurred at least in two documents.

3.2. Experimental setup

At the end of the pre-processing process, we obtained a documents × terms matrix \( T \) with 2225 rows and 14511 columns. This matrix, whose generic element \( t_{ij} \) is the frequency of the \( j \)-th term in the \( i \)-th document, was transformed according to two other weighting schemes, obtaining a matrix \( T_b \) (Boolean scheme) and a matrix \( T_{tf-idf} \) (tf-idf scheme: Salton and Buckley, 1988). In a Boolean scheme, only the presence/absence of each term in each document is considered. In a tf-idf scheme, the occurrences of each term in a document are multiplied by the reciprocal of the fraction of documents containing the term on the total number of documents in the collection, jointly using as term importance a local weight and a global weight.

<table>
<thead>
<tr>
<th>weighting schemes</th>
<th>Boolean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Term Frequency</td>
</tr>
<tr>
<td></td>
<td>Term Frequency – Inverse Document Frequency</td>
</tr>
<tr>
<td>clustering algorithms</td>
<td>UPGMA (Sokal and Michener, 1958)</td>
</tr>
<tr>
<td></td>
<td>K-means (McQueen, 1967)</td>
</tr>
<tr>
<td></td>
<td>Spherical K-means (Dhillon and Modha, 2001)</td>
</tr>
<tr>
<td>community detection</td>
<td>Louvain algorithm (Blondel et al., 2008)</td>
</tr>
<tr>
<td></td>
<td>proximity measures:</td>
</tr>
<tr>
<td></td>
<td>jaccard similarity, cosine similarity, matching coefficient</td>
</tr>
</tbody>
</table>

Table 2 – Comparative study setup.

On each of these three matrices, we ran three clustering algorithms: UPGMA (Sokal and Michener, 1958), K-means (McQueen, 1967) and Spherical K-means (Dhillon and Modha, 2001). These algorithms are widely used in Text Mining for unsupervised text categorisation (Misuraca et al., 2018). For UPGMA, which returns a set of nested partitions, we chose a 5-clusters solution according to the true numbers of categories in the BBC dataset. For centre-based algorithms (K-means and Spherical K-means), we instead set a priori the true numbers of categories in the BBC news collection as the number of clusters.

To execute the community detection, we considered the similarities among document-vectors. Starting from each lexical matrix, we computed a proximity matrix with three different measures: jaccard similarity, cosine similarity and matching coefficient. The jaccard similarity was used only for the Boolean weighting scheme, whereas the cosine similarity was calculated for the three different weighting schemes. The matching coefficient for each couple of document-vectors was calculated as their dot product. If the weighting scheme is Boolean, the similarity between two different documents is measured as the numbers of shared terms. Considering more complex schemes, the matching coefficient embodies the different occurrences of the shared terms in the documents or the occurrences of the terms in each document dampened by their discriminative power in the collection (i.e. the tf-idf scheme). On these matrices,
graphically viewed as networks of documents, we applied the well-known Louvain algorithm (Blondel et al., 2008). According to Yang et al. (2016), this algorithm outperforms the other modularity-based algorithms with shorter computing times both in the case of small and large networks. Since the Louvain algorithm does not have input parameters, the number of communities is automatically determined and may differ from the true number of categories. Table 2 synthetically reports the alternatives used in this study.

3.3. Results evaluation

The evaluation of the different approaches was performed, both considering the categorisation process accuracy and the running time. We calculated two external validation indices, purity (Manning et al., 2008) and normalised mutual information (NMI: Strehl and Ghosh, 2002) to compare the obtained solutions with the so-called gold standard, i.e. the true classification of the BBC news collection. The advantage of using these metrics over other well-known external validation measures – such as precision, recall and F-measure (see Sokolova and Lapalme, 2009) – relies on the fact that they do not require a one-to-one correspondence between clusters and classes. Moreover, NMI provides a robust indication of the level of agreement between a given clustering solution and the gold standard. Noteworthily, the running times depend on the technical characteristics of the computer used to perform the analyses. The results presented in the following were obtained on a MacBook Pro 2.8 GHz Intel Core i7, 16 GB 1133 MHz DDR3 with macOS Sierra ver. 10.12.6.

Table 3 – Comparison of clustering results on BBC news collection.

<table>
<thead>
<tr>
<th></th>
<th>BOOLEAN</th>
<th>TERM FREQUENCY</th>
<th>TF-IDF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Purity</td>
<td>NMI</td>
<td>TIME (sec.)</td>
</tr>
<tr>
<td>UPGMA</td>
<td>0.240</td>
<td>0.006</td>
<td>136.026</td>
</tr>
<tr>
<td>Kmeans</td>
<td>0.746</td>
<td>0.595</td>
<td>43.266</td>
</tr>
<tr>
<td>Spherical Kmeans</td>
<td>0.948</td>
<td>0.842</td>
<td>15.486</td>
</tr>
<tr>
<td>Louvain algorithm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jaccard similarity</td>
<td>0.230</td>
<td>0.000</td>
<td>8.891</td>
</tr>
<tr>
<td>cosine similarity</td>
<td>0.082</td>
<td>0.169</td>
<td>3.011</td>
</tr>
<tr>
<td>matching coefficient</td>
<td>0.790</td>
<td>0.662</td>
<td>3.080</td>
</tr>
</tbody>
</table>

Table 3 reports the validation measures together with the running times for each algorithm. In bold font, we highlighted the best-performing algorithms according to the different weighting schemes. Louvain algorithm outperformed all the other clustering algorithms, considering as similarities among document vectors the matching coefficient and as the importance of each term the tf-idf. Louvain algorithm also provided good results with the boolean and term frequency weighting scheme. However, in latter this case, spherical K-means returned better partitions than the ones obtained with a community detection approach. These results were also confirmed by looking at Figure 1, where the confusion matrices of the best clustering solutions (highlighted in bold in Table 3) are shown. The colour scale ranges from red to green, looking at the frequency value in each cell. The closer the colour gets to the green, the better is the matching between the partition and the gold standard. Among the solutions obtained with the community detection approach, only in one case (term frequency - matching coefficient) Louvain algorithm returned a number of clusters (4) different from the gold standard.
Concerning the running time, we saw that a community detection approach with the Louvain algorithm was faster than the other alternative strategies.

Figure 1 – Confusion matrices of the clustering results.

4. Remarks and future development

The study proposed in this paper was devoted to comparing the clustering algorithms widely used in the literature of Text Mining with the algorithms used in network analysis for discovering groups of nodes with similar characteristics, in the framework of the so-called community detection. Our preliminary results suggested that the choice of a different weighting scheme may affect the performances of the algorithms. We noted that a more complex weighting scheme such as tf-idf produced better results than those obtained with a binary or a term frequency scheme. In particular, using tf-idf and computing the proximities between documents with the matching coefficient, Louvain algorithm outperformed all the other alternative strategies both in terms of partition quality and running time.
In an unsupervised framework, it is essential to consider that the researcher does not have any prior information about the number and the composition of the clusters. Therefore, a community detection approach returns good results without any other input than the matrix of similarities among documents.

This paper can be considered as a preliminary study. Our findings have to be confirmed by analysing other collections of documents, including also corpora of short texts. Future developments will be lead to a broader and more detailed comparison, considering other weighting schemes for the documents × terms matrix, other proximity measures for the documents × documents matrix and other community detection methods.

References


