Knowledge-Based Systems 229 (2021) 107342

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

What is this Cluster about? Explaining textual clusters by extracting relevant keywords

Antonio Penta *, Anandita Pal

Analytics & AI Group, R&D Global Innovation Center -The Dock Accenture, 7 Hanover Quay, Grand Canal Dock, Dublin, Ireland

ARTICLE INFO

Article history:
Received 3 July 2020
Received in revised form 9 October 2020
Accepted 23 July 2021
Available online 4 August 2021

Keywords:
Document clustering
Text analytics
Explainability
Cluster summarization
Cluster labelling
Clusters explanations

ABSTRACT

Document clustering is a powerful method with numerous applications, where the core idea is to group text into smaller and more manageable pieces of semantically related information. While there has been progress in the research community on improving the quality of clusters, the extraction of information that explains the semantic content of the clusters is still mostly a manual activity, as it requires the inspection of sample documents. In this work, we propose three main cluster explanation approaches that extend the current state of the art, which is mostly based on predicting a label for each cluster (i.e. cluster labelling). The first approach is based on scores extracted from the word distributions; the second is based on augmenting the score-based explanations using external knowledge-bases, and the third one combines the first two methods to exploit the knowledge coming from a labelled dataset. We also discuss the limitations of the current metric presented in the literature and extend it to evaluate our approaches using ground truth. We also provided an extensive set of evaluations to verify the effectiveness of the proposed algorithms, by running multiple experiments to compare our approaches with different baselines. Results indicate that improved explanations are possible and that linking external knowledge can provide more general cluster descriptions. Semi-supervised approaches also demonstrated meaningful insight. User studies were also conducted highlighting that users prefer specialized explanations.

1. Introduction

Text clustering algorithms are designed to group documents based on their similarity, and they have been applied in different use-cases where the main problem is to extract information in an unsupervised manner. The resulting text clusters are used for different tasks such as content-based indexing, intent discovery, text summarization, and so on. We expect that the information within a cluster represents a coherent piece of knowledge, but in reality, we need a human inspection stage to navigate its elements, understand its content, and eventually answer the question: "What is this Cluster about?". This approach limits the use of clusters in the automatic decision-making processes. Instead, an automatic explanation of a cluster content could remove this human step, such that the users and/or application can just focus on consuming the clustering results that fit their needs. A first solution toward the cluster explanation could be a sampling strategy to select some documents within the cluster and propose them as prototypes of an explanation, but this approach has many limitations: (i) text documents could be so long that it will limit both the user's interaction/experience and/or the use or clusters within an application; (ii) it is not clear what is the relevant part of a document that is representative of the cluster; (iii) the sampling strategy could be driven by a biased understanding of the domain knowledge from a user, due to the fact that he/she is not aware of the full content within the collection. Therefore, in this paper, we address the following research question: How can we automatically choose word-based explanations that differentiate the content of a cluster with respect to the others? The purpose is to explain the meaning of the clusters to provide the minimum and relevant information that can be easily consumed in the decision making process. We focus our attention on topic-based explanation, which means that we are addressing the problem of extracting a set of relevant keywords (i.e. words) from the cluster that can help the users to understand its content. An alternative to the topic-based explanation is a visual description of the clusters, but this choice is quite limited because the information that is conveyed is related to the shape of the clusters. Visual description can show how the clusters are well separated but the user still cannot infer the content of the cluster without inspecting the documents. Visual description also requires the use of a dimensionality reduction technique that can also distort the results of the clustering algorithm. Besides some interesting advances in textual clustering [1], little work has been done to properly address the above question. In the literature, a similar

* Corresponding author.
E-mail address: antonio.penta@accenture.com (A. Penta).

https://doi.org/10.1016/j.knosys.2021.107342
0950-7051/

problem has been presented with the name of *cluster labelling* in [2,3]. We point out that summarizing, labelling, and explaining a cluster could be conceptually different tasks. Cluster summarization is related to how much we can compress the content of a set of documents in a smaller document, and the output should be a proper summary. In this case, the evaluation should be based on the standard metrics used in summarization algorithms such as the ROUGE measure [4]. Labelling has the goal of finding a unique concise word for the cluster, which is named *label*, to provide compact knowledge to the user to separate the documents. In other words, labelling a cluster is a well-defined task assuming it is possible to annotate the document with one word. Labelling can be automatically evaluated by matching the proposed label with the target label. However, in the real scenarios multiple words are used to describe the content of the documents, and sometimes they also introduce an order among them to distinguish more important words. We consider the explanation as a middle ground between the summary and the labelling approach. The purpose of the explanations is to have an ordered set of words that accurately explains what are the topics expressed by the documents within a cluster. The explanations provide more information to the user when the clustered documents have multiple rich contents, which is the most common cause in the real scenarios. In this paper, we describe new approaches to extract explanations from clusters, and we summarize our contributions as follows:

- We introduce three different approaches to compute explanations: (i) explanations based on scores; (ii) explanations based on external knowledge, (iii) explanations based on supervision.
- We introduce a flexible framework based on Integer Linear Programming (ILP) to compute the explanations based on external knowledge, which lets us incorporate the domain knowledge directly in the constraints of the mathematical model.
- We extend the metric presented in the literature to overcome its limitation and we run multiple experiments to compare our approaches with scores proposed in literatures and well known topic-modelling technique, discussing the conditions needed to obtain valuable explanations from our approaches.

The paper is divided into the following sections: Section 2 contains the related works, Section 3 describes the methods based on scores; Section 4 describes how we use the external knowledge in the ILP framework for extracting the explanations; Section 5 describes how we combine the supervision (data labelling) and the ILP framework for explaining the clusters; Section 6 is related to the experiments; in Section 8 we present the conclusion of our work and its results.

2. Related works

Although there has been much research in designing clustering algorithms for textual data [1], less work has been developed in understanding how to extract representative information from clusters of documents. The most common approach presented in the current literature is the cluster labelling one, where the main idea is to select a label from the most relevant words or n-grams extracted from clusters. The resulting word relevance scores are mostly based on combining the words frequency distributions with a cluster. For example, [2] presents an algorithm to assign labels to the clusters by first computing a set of scores for each word in the documents belonging to a cluster, and then selecting the best word based on weights that are learned using a linear regression technique. This selection process can be limited in the presence of multicollinearity phenomena that are quite common in textual data and can also lead to overfitting problems. In [3], the authors propose to enhance the cluster labelling task using external knowledge (i.e. Wikipedia) to improve the accuracy of the relevant terms extracted from the clusters. They show that external sources of knowledge are useful in determining labels that are more significant to human judgement, but they exploit only the statistical correlations between the metadata in Wikipedia and the words extracted from the clusters. However, it is also essential to consider the structure of the knowledge such as the links between concepts in order to derive more relevant labels. In this regard, pairing the extracted words with the Wikipedia metadata as done in [3] does not fully exploit the possible impact of the external knowledge in improving the quality of the cluster explanations. Labelling clusters using an ensemble of different methods is another approach used in literature, but the definition of the logic using to combine the labellers could be challenging, for example, [3] shows that different cluster labellers do not have a consistent performance if we consider the same number of words as their input. Roitman et al. [5] presents a fusion approach that combines the same scores on different versions of the clusters obtained by subsampling the original cluster. Their main hypothesis is that a cluster labeller's labelling choice should remain stable even in the presence of a slightly incomplete cluster data. The proposed solution shows good results but requires different parameters that do not have a clear interpretation, therefore they are difficult to be tuned, making the overall approach difficult to scale with other datasets. In the machine learning literature, there are many solutions for creating an ensemble of labellers [6], but most solutions require weights to be learned with some training data. Moreover, it will be much more valuable having just one main relevant score that is robust enough to extract the main content of the clusters, making the whole computation lighter in the context of large corpora. Li et al. [7] present an approach to label clusters using phrases extracted from a set of sentences using linguistic patterns and rules. The candidate phrases are scored using probabilities conditioned on a context window. The proposed approach is limited to textual data where we can extract meaningful linguistic patterns using rules defined a priori, and the score requires fine tuning procedures. Lopes et al. [8] present an approach to labels clusters obtained from numerical features, the procedure requires training a number of neural networks proportional to the number of attributes, which could be really inefficient with high-dimensional space like the one described by BOW textual features. Orangeville et al. [9] introduce an approach to label clusters obtained from the support vector clustering approach, therefore is focused only on one particular clustering approach. Role and Nadif [10] use a graph representation techniques to visualize cluster content, therefore their main objective is not to extract any explanations or labels from clusters but to visualize their content. The work presented in [11] is also using scores and WordNet [12] to extract labels from clusters, their approach is tailored to the WordNet knowledge base, therefore cannot be extended to different scenarios like the ones described in this paper. We use as baseline in our evaluation the work from Carmel et al. [3] and Tseng [11] among others.

3. Extracting clusters explanations from scores

In this section, we describe three approaches to compute scores for each keyword, which will then be used to decide how to explain the clusters. The first two scores are presented for the first time in the context of the cluster explanation problem, while the third one has already been proposed in the literature, and we report here for completeness. All the scores are designed to model the relevance of the words based on their statistical properties inside and outside the clusters.

3.1. Change in documents and terms frequency method

The first score we consider is designed to measure the difference in the distribution of the words before and after the clustering procedure. In particular, we model the idea that a good cluster is a coherent piece of knowledge, and by aggregating all its samples in one document we can obtain the best description of itself. Therefore, we measure the ratio in the frequencies of the words before the clustering (considering each document as a cluster) and after the clustering (an aggregated document for each cluster). Let us consider a set of documents $\mathcal{C}$, an ith partition (cluster) $\mathcal{C}_i \subseteq \mathcal{C}$, and a word w_j. We define the Change in Documents and Terms Frequency Score (S_{CDTF}) for w_j as follows:

$$S_{CDTF}(w_j, \mathcal{C}_i, \mathcal{C}) = \frac{\phi_1(w_j, \mathcal{C}_i) - \phi_2(w_j, \mathcal{C})}{\max\{\phi_1(w_j, \mathcal{C}_i), \phi_2(w_j, \mathcal{C})\}} \phi_3(w_j, \mathcal{C}_i) \tag{1}$$

with :

$$\phi_1(w_j, \mathcal{C}_i) = \frac{D(w_j, \mathcal{C}_i)}{|C_i|},$$

$$\phi_2(w_j, \mathcal{C}) = \frac{D(w_j, \mathcal{C})}{|\mathcal{C}|},$$

$$\phi_3(w_j, \mathcal{C}_i) = 1 + log_2(1 + \frac{T(w_j, \mathcal{C}_i)}{\max_{w^* \in W(\mathcal{C}_i)} T(w^*, \mathcal{C}_i)})$$

$D(w_j, \mathcal{C}_i)$ is a function that returns the number of documents in the set $\mathcal{C}_i$ containing the word w_j, the same for $D(w_j, \mathcal{C})$; $T(w_j, \mathcal{C}_i)$ a function that returns the number of times the word w_j appears in the documents belonging to $\mathcal{C}_i$; $W(\mathcal{C}_i)$ is a function that returns the set of words extracted from the documents in $\mathcal{C}_i$. The score is a product of two quantities, the first one measures the differences in terms of word counts at document level before and after running the clustering algorithm, while the second quantity is used to logarithmically amplify this score for the words that are more frequent within the cluster. We note that the score is upper bounded within the interval $-\alpha \leq S_{CDTF} \leq 2$, with $\alpha \in \mathbb{R}_{>0}$. If we consider a very frequent word w_j within a cluster (i.e. $\phi_3 \approx 2$), we will have that $S_{CDTF}(w_j) \to 2$ as the w_j is distributed more uniformly across the clustered documents (i.e. $\phi_1 \gg \phi_2$). The score measures the relevance of the words based on the differences in their frequencies across the clusters. We also note that the score is computed for each word in the collection, therefore the score is also lower bounded because in most of the cases there will be at least one word not appearing in the clustered documents so that $\phi_3 = 1$ since it is only considering $\mathcal{C}_i$.

3.2. Relevance feedback method

Let us suppose that we have a hypothetical information retrieval system, and we are interested in finding the best query that retrieves all the documents assigned to a cluster $\mathcal{C}_i$. If the query maximizes precision and recall related to the information in $\mathcal{C}_i$, we can use its terms to explain the cluster. A simple solution could be to use a long query that contains all the keywords in $\mathcal{C}_i$, but a better solution would be to consider a shorter query that contains only the most relevant words without compromising the retrieval results. The relevance of the words belonging to $\mathcal{C}_i$ can be computed using the documents inside and outside $\mathcal{C}_i$ as positive and negative feedback respectively. In this context, the cluster explanation problem is reduced to a query inference problem in the presence of an information retrieval system based on external feedback. In particular, we can score the words using the relevance feedback theory proposed by Ruthven and Lalmas [13] and Robertson and Zaragoza [14] for the probabilistic information retrieval system. In particular, according to the probabilistic ranking principle of Robertson [15], an optimal retrieval system ranks documents using scores based on the probability that the user query is relevant to the documents in the system. This probability of relevance is computed using two approximations: (i) the first is based on the Binary Independence Model [16], which assumes that the distribution of words in a relevant and irrelevant document is independent; (ii) the best strategy to weight the words in the query is the one that considers not only presence of the search term in documents but also their absence [17]. Given a set of documents $\mathcal{S}$, we define $P(w_j \in_d \mathcal{S})$ as the probability that w_j is contained in the documents in $\mathcal{S}$, we have that $P(w_j \notin_d \mathcal{S}) = 1 - P(w_j \in_d \mathcal{S})$, due to the Binary Independence Model assumption. We also introduce $P(w_j \in_w \mathcal{S})$ to refer to the probability that the word w_j belongs to vocabulary extracted from the documents in $\mathcal{S}$. Based on the work in [17], the relevance feedback score S_{RF} for a word w_i can be formulated as follows:

$$S_{RF}(w_j, \mathcal{C}_i, \mathcal{C}) = \log(\frac{\phi_1(w_j, \mathcal{C}_i)}{\phi_2(w_j, \mathcal{C}_i, \mathcal{C})}) \phi_3(w_j, \mathcal{C}_i, \mathcal{C}) \tag{2}$$

with:

$$\phi_1(w_j, \mathcal{C}_i) = \frac{P(w_j \in_d \mathcal{C}_i)}{P(w_j \notin_d \mathcal{C}_i)} = \frac{P(w_j \in_d \mathcal{C}_i)}{1 - P(w_j \in_d \mathcal{C}_i)}$$

$$\phi_2(w_j, \mathcal{C}_i, \mathcal{C}) = \frac{P(w_j \in_d \mathcal{C} - \mathcal{C}_i)}{P(w_j \notin_d \mathcal{C} - \mathcal{C}_i)} = \frac{P(w_j \in_d \mathcal{C} - \mathcal{C}_i)}{1 - P(w_j \in_d \mathcal{C} - \mathcal{C}_i)}$$

$$\phi_3(w_j, \mathcal{C}_i, \mathcal{C}) = P(w_j \in_w \mathcal{C}_i) - P(w_j \in_w \mathcal{C} - \mathcal{C}_i)$$

In the above formula, we have two main quantities, the first one is the logarithmic odds ratio which takes into account the positive (ϕ_1) and the negative (ϕ_2) feedback in terms of document frequencies, the second one (ϕ_3) is an absolute difference between two word frequencies, one computed using the positive feedback and one based on the negative feedback. This helps us to balance the relevance between very rare and more frequent terms. We can further details the above probabilities with the following notations. Let us consider N as the number of documents in the collection, R_i as the number of documents in the cluster i, r_j^i as the number of documents in the cluster i that contain the word w_j, n_j as the number of documents in the collection that contains the word w_j, c_j^i as the count of the term w_j in the vocabulary extracted from cluster i, c_j^* as the count of the term w_j in the vocabulary extracted from the entire collection of documents, $\mathcal{W}_i$ as the vocabulary extracted from the documents belonging to the cluster i, $\mathcal{W}$ as the vocabulary extracted from the documents belonging to the entire collection. Then, we can reformulate the above terms as following:

$$\phi_1(w_j, \mathcal{C}_i) = \frac{r_j^i + 0.5}{R_i - r_j^i + 0.5},$$

$$\phi_2(w_j, \mathcal{C}_i, \mathcal{C}) = \frac{n_j - r_j^i + 0.5}{N - R_i - n_j + r_j^i + 0.5},$$

$$\phi_3(w_j, \mathcal{C}_i, \mathcal{C}) = \frac{c_j^i + 0.5}{|\mathcal{W}_i| + 0.5} - \frac{c_j^* - c_j^i + 0.5}{|\mathcal{W}| - |\mathcal{W}_i| + 0.5}$$

In the computation of the above probabilities, we take into account a small pseudo-count of frequency 0.5. This score does not have a numeric upper bound. For example, if we consider the case where w_j is only contained in the cluster i, and all the documents in the cluster contains w_j, we have $n_j = r_j^i$, and $R_i = r_j^i$, then $S_{RF} \to \infty$ as $N \to \infty$.

3.3. Jensen–Shannon divergence method

In this section, we discuss a score proposed in [18] for modelling queries and used in [3] for cluster labelling. The idea is to find the set of words that maximizes the Jensen–Shannon

Divergence (JSD) between their distribution inside the cluster and in the whole collection. Each word is scored according to its contribution to the JSD. The top words are selected as cluster labels. The Jensen–Shannon Divergence score (S_{JSD}) is defined as follows:

$$S_{JSD}(w_j, \mathcal{C}_i, \mathcal{C}) = \phi_1(w_j, \mathcal{C}_i, \mathcal{C}) + \phi_2(w_j, \mathcal{C}_i, \mathcal{C}) \tag{3}$$

with:

$$\phi_1(w_j, \mathcal{C}_i, \mathcal{C}) = P(w_j \in_w \mathcal{C}_i) \log(\frac{P(w_j \in_w \mathcal{C}_i)}{\phi_3(w_j, \mathcal{C}_i, \mathcal{C})})$$

$$\phi_2(w_j, \mathcal{C}_i, \mathcal{C}) = P(w_j \in_w \mathcal{C}) \log(\frac{P(w_j \in_w \mathcal{C})}{\phi_3(w_j, \mathcal{C}_i, \mathcal{C})})$$

$$\phi_3(w_j, \mathcal{C}_i, \mathcal{C}) = \frac{1}{2}(P(w_j \in_w \mathcal{C}_i) + P(w_j \in_w \mathcal{C}))$$

Based on the notations introduced in the previous subsection, we can estimate the above probabilities (based on maximum likelihood estimation) as follows:

$$\phi_1(w_j, \mathcal{C}_i, \mathcal{C}) = \frac{c_j^i}{|\mathcal{W}_i|} log(\frac{2c_j^i * |\mathcal{W}|}{c_j^i * |\mathcal{W}| + c_j^* * |\mathcal{W}_i|})$$

$$\phi_2(w_j, \mathcal{C}_i, \mathcal{C}) = \frac{c_j^*}{|\mathcal{W}|} log(\frac{2c_j^* * |\mathcal{W}_i|}{c_j^i|\mathcal{W}| + c_j^*|\mathcal{W}_i|})$$

$$\phi_3(w_j, \mathcal{C}_i, \mathcal{C}) = \frac{1}{2}(\frac{c_j^i}{|\mathcal{W}_i|} + \frac{c_j^*}{|\mathcal{W}|})$$

We note that if we consider an extreme case where a word appears only in a selected cluster, we have that $c_j^* = c_j^i = c_j$, then the above score can approximated as follows:

$$\frac{c}{|\mathcal{W}_i|} \underbrace{\log(\frac{2|\mathcal{W}|}{|\mathcal{W}| + |\mathcal{W}_i|})}_{>0} + \frac{c}{|\mathcal{W}|} \underbrace{\log(\frac{2 * |\mathcal{W}_i|}{|\mathcal{W}| + |\mathcal{W}_i|})}_{<0}$$

We observe that in this extreme case the S_{JSD} is a difference between two factors: one proportional to the document frequency (DF) within the selected cluster and one related to the proportion of the cluster size respect to the dimension of the whole collection. On the opposite side, if a word does not appear at all in a cluster, we have that c_j^i=0, and the S_{JSD} score is proportional to the DF of the word within the collections. These two extreme cases show that in the case of maximum likelihood estimation, the S_{JSD} score could be affected by the size of the cluster compared to the size of the whole collection, therefore the cluster can be biased by the size of the cluster rather than by the statistical property of the words within the cluster. We note that the method in Section 3.1 is less sensitive to the cluster size because its upper bound is not related to the cluster of size.

3.4. Correlation coefficient method

In this section, we report for completeness the method proposed in [11], which is used as one of the baseline in our analysis. Let us consider a word w_i, a cluster $\mathcal{C}_i \subseteq \mathcal{C}$, where $\mathcal{C}$ is the set of all the documents, then we can define the following quantities: *TP* the cardinality of the set all the documents in $\mathcal{C}_i$ which contain w_i - (True Positive), *TN* the cardinality of the set all the documents in $\mathcal{C} - \mathcal{C}_i$ which not contain w_i - (True Negative), *FP* the cardinality of the set all the documents in $\mathcal{C}_i$ which not contain w_i - (False Positive), and *FN* the cardinality of the set all the documents in $\mathcal{C} - \mathcal{C}_i$ which contain w_i - (False Negative) . Tseng [11] propose the following correlation coefficient (S_{CC}) to score the importance of the word w_i in the cluster $\mathcal{C}_i$:

$$S_{CC}(w_j) = \frac{TP \times TN - FN \times FP}{\sqrt{(TP + FN)(FP + TN)(TP + FP)(FN + TN)}} \tag{4}$$

We note that in an ideal scenario where FP and FN are equals to 0, which means that w_i is present in all the documents in $\mathcal{C}_i$, we have that $S_{CC}(w_j)$ is equal to 1. The S_{CC} score, as S_{CDTF}, is also based on evaluating the differences between the selected cluster and the remained documents, but the S_{CDTF} considers only the difference between the cluster and the full set of documents. This makes the S_{CDTF} score more robust because is less dependent on the size of the other clusters, while the S_{CC} is influenced by their sizes due the fact that it explicitly includes the true negative. The S_{CDTF} ha also a component related to the term frequency, which is also mentioned as an option in S_{CC} in the [11] work.

4. Extracting clusters explanations from an external knowledge base

The approaches described in the previous sections select keywords for cluster explanations only based on their statistical properties. Now, let us assume that we have an external knowledge base, which is a set of documents annotated by humans and an ontology that expresses the relations among the annotations. We would like to leverage this information to improve the explanations by linking the concepts within the knowledge base with the words in the clusters. In this case, the extracted explanations will be closer to how the humans will explain the content of a cluster which is mostly based on the knowledge that they have of the clustered domain, instead of filtering the words based on some statistical properties as done in the previous sections. Therefore, the question is: how can we use that external information to improve the explanations by exploiting the correspondence between the clustered documents and an external knowledge base? In this regard [3] already showed that a knowledge base method based on score propagation outperforms feature selection methods for the cluster labelling task. The algorithm is based on a voting mechanism that takes as input the words extracted using the S_{JSD} method and finds the most voted label among categories in Wikipedia and synsets in WordNet. In this context, the knowledge base has a particular structure, the set of documents are organized by concepts, topics, entities and the annotations are the categories or concepts belonging to a predefined list. The suggested approach is also based on the strong assumptions that a user will be able to use the Wikipedia/WordNet knowledge for all the applications, but this is not the case for many industrial problems where the domain is quite specific, and the users do not have well-structured domain knowledge. At the same time, the results still rely on the process used to build Wikipedia, where the editors can choose from a pool of categories to index the pages to enforce links among articles belonging to the same context, or in the case of polysemy they need to refer to a disambiguation page with predefined categories. In light of these limitations, we present a more general approach, where we assume: (i) our external knowledge base is a set of annotated documents; (ii) we have an ontology that models the semantic relations among the annotations, (iii) we can retrieve the annotated documents by specifying a query made by the explanations extracted from one of the score-based methods described in the previous sections. We note that these assumptions are quite general, and they can include how Wikipedia has been used from Carmel et al. [3] for the clusters labelling task, but they can also hold in different contexts, for example when we have a dictionary where the subject matter of experts have annotated documents with relevant tags, or we have a search engine indexing medical regulations annotated with tags, or we have indexed repositories of dialogues where customers' utterances have been annotated with tags expressing users' intentions. At the core, our approach is based on an algorithm that searches for the best explanations among the information provided by

the knowledge-base within some constraints. The possibility to formalize in the program which properties an annotation select as an explanation makes the model flexible enough to be applicable in broader scenarios. Henceforth, we will mostly use the term "*tags*" to refer to the annotations of the documents in the knowledge base, because they can have multiple semantic roles (e.g. from general concepts, factual entities to artificial codes used to reference domain-based knowledge) as their main scope is to describe multiple aspects of the documents. The full algorithm for extracting clusters explanations using the external knowledge is described in Algorithm 1. The objective of the algorithm is to find an optimal mapping between tags and clusters in a weighted graph that connects them. The graph is used to model all the possible explanations between a cluster and the tags and the semantic properties of the tags according to the ontology stored in the knowledge base. The goal is to find the relevant tags that explain the content of the clusters. The tags are extracted from the annotated documents retrieved from the knowledge-base using the explanations obtained from a score-based method as a query. We name these explanations as *initial explanations* and they are used as inputs of our knowledge-based approach. The weights of the nodes and edges in the graph reflect their importance according to the information in the cluster and in the knowledge-base. Let us now introduce some initial notations using in the Algorithm 1. Let us consider a set of tuples ($\mathcal{D}$) made by a text document (d_i) and a set of tags ($\mathcal{T}_j$), $\mathcal{D} = \{(\mathcal{T}_1, d_1), \ldots, (\mathcal{T}_n, d_n)\}$, where $\mathcal{T}_i \cap \mathcal{T}_j \neq \emptyset, \forall i, j = 1 \ldots n$. Now, let us consider a text clustering algorithm, that produces a set of clusters $\mathcal{C}$. Let us then extract the initial explanations of these clusters using one of the methods described in Section 3. The initial explanations for the cluster $c_i \in \mathcal{C}$ are a set of keywords $\mathcal{K}_{c_i}$. The algorithm is looking for a set of tags to explain c_i (cluster) using $\mathcal{K}_{c_i}$ (the initial explanations), $\mathcal{D}$ (documents annotated with tags) and the information in the ontology $\mathcal{O}$. In most applications, we have that $|\mathcal{W}| \gg |\mathcal{T}|$, where $\mathcal{W}$ is the set of all possible unique words (i.e. the source of the explanations), and $\mathcal{T}$ is the set of all the unique tags used in $\mathcal{D}$ that can be used to extract the explanations. The algorithm is divided into two parts. The first part (lines 1–20 described in Section 4.1) is used to extract, process, normalize the tags from $\mathcal{D}$, and to store all the useful information in a set of data structures. The second part (line 21 described in Section 4.2) creates the weighted graph between the tags and the clusters, and it then solves an optimization problem to find the optimal mappings between the clusters and the tags based on the objective function that maximizes the strength of the associations under constraints related to the number of possible mappings and their coherence properties. The optimal tags will be selected as labels for the clusters. We are now in a position to describe the details of both parts of the algorithm in the following sub-sections.

4.1. Tag extractor section — Algorithm 1 (lines 1–20)

The purpose of this section is to explain how we process the input words (i.e. initial explanations) to retrieve the information from the external knowledge-base, which are then used in the next section of the algorithm. First given a cluster, we use its initial explanations (i.e. a set of words with scores) to query the annotated documents in the knowledge-base, to extract the related annotations, which we have called tags in our approach and to associate them to the cluster. We also extract the semantic relations among the tags using the ontology in the knowledge-base. We then compute three different weights that will be used in the next section: (i) one related to the global importance of each tag; (ii) one to weigh the relationships between the tags and the clusters; iii) one related to the semantic coherence of the tags related to the same cluster. In particular, the association between a tag and a cluster is weighted based on two factors: (i) the first one is related to how important the word is to retrieve the tag and the importance is measured by its scores; (ii) the second one is related to how often the same tag is used as an annotation in the retrieved documents. The scope of the semantic coherence is to select tags that are semantically related. By considering this weight, we force the algorithm to choose a cluster explanation where tags are related to the same concepts, therefore the explanations will be semantically coherent. This information extracted in this section of the algorithm is hosted in the following data structures, where we use c to refer to the cluster, t to refer to the tag:

- $Dict_{ct}$ is the double-indexed dictionary used to map the clusters to the tags and the tags to the set of keywords and scores;
- $Dict^{*}_{ct}$ is the double-indexed dictionary used to map the clusters to the tags, and then the tags to the sum of the scores;
- $Dict_{t}$ is the dictionary used to map the tags to the average scores across all the clusters.
- $Dict_{c_{tt}}$ is the dictionary used to store the pairwise semantic similarity among the tags related to the same cluster.

As described in the section above, we use $\mathcal{D}$ to refer to the set of annotated documents within the external knowledge-base. In particular, the algorithm processes only the *TopD* retrieved documents for the next steps. In the algorithm, we use the following notations to represent clusters, tags, and the initial explanations with words and scores:

- n is the total number of clusters, m is the total number of tags, c_i refers to a cluster belonging to the set of all the clusters ($\mathcal{C}$), t_p, t_{p_1}, t_{p_2} refer to the tags, $\mathcal{T}$ refers to a set of tags,
- $\mathcal{IE}_{c_i} = \{(wo_1, s_1), \ldots, (wo_{TopE}, s_{TopE})\}$ is the set of the initial explanations of the cluster c_i made by a set of words and scores. We use the symbol wo to refer to a word and s to refer to the score. The score belongs to $\mathbb{R}$ and the set is ordered $s_i > s_j \forall i < j$. We assume that $|\mathcal{IE}_{c_i}| = TopE$ $\forall i = 1 \ldots, n$, where $TopE$ is maximum number of initial explanations to consider,
- with r_j we refer to the position of the word wo_j in $\mathcal{IE}_{c_i}$, we name r_j as ranking weight.

In this section of the algorithm, we also use the following subroutines :

- *initial_explanation*, which extracts the initial explanations using one of the methods described in Section 3. It takes in input a cluster and the maximum number of explanations to consider ($TopE$).
- *retrieve*, which extracts a set of annotations (tags) from the annotated documents retrieved from the knowledge-base using a word in the initial explanations to formulate the input query. It takes as input a word, the set of annotated documents, and the maximum number of documents to retrieve ($TopD$). The retrieval can be based on state-of-the-art text retrieval tools or on any web services (e.g. Wikipedia API).
- *inverse_average*, which computes a global weight for each tag based on the inverse of the average weight computed across all the weighted associations between a tag and all the clusters.
- *normalize*, which is used to normalize the weights in a data structure in a fixed interval. It could take in input one of the data structure specified above ($Dict_{ct}$, $Dict^{*}_{ct}$, $Dict_{c_{tt}}$).

Given a cluster, the extraction of the initial explanations ($\mathcal{IE}_{c_i}$), the retrieval of the set of tags ($\mathcal{T}$), the computation of the weights (r_j), and the association between the cluster and the tags in $Dict_{ct}$ is done from line 3 to line 9 of the Algorithm 1. From line 10 to line 16, we compute further steps in the Algorithm 1. Given a cluster (c_i), we store in $Dict^*_{ct}$ the final weights between a cluster (c_i) and a tag (t_p), computed as the sum of the ranking weights (r_j). This weight is taking into account the two factors discussed at the start of this section, one related to the importance of the words (ranking weights) and one related to the frequency of the tags in the results of the retrieval process (sum). We note that for each cluster, the same tag can be retrieved multiple times using words with different ranking weights, therefore we consider the sum of them as final weight. We also note that the ranking weights are already normalized across the clusters, and [3] showed that using the rank as a method to score the importance of the metadata coming from the external knowledge base produces better results in the quality of the cluster labels. We note that we did not balance the weights of the tags based on the rank of the documents after the query. We also note that there are multiple ways to rank documents [19], therefore we will leave for future work the study of how the rank of the document can be used to further balance the weights. We then compute the weight using to measure the semantic coherence between tags in the same cluster. This is computed by the *semantic_coherence* subroutine and the output is stored in $Dict_{c_{tt}}$. We model the coherence among the tags by forcing them to have a closer semantic relation. There are different ways to model semantic relations in a concept space [20], but here we use the cosine similarity in a vector space. The vector space of tags is obtained by their distributional representations learned from an external shared corpora [21–23]. In line 17, we compute a global weight for each tag using the *inverse_average* subroutine, and this information is stored in $Dict_t$. The inverse average score helps the next section of the algorithm to penalize tags that are spread among all the clusters which we can assume to be noisier, therefore they will have lower scores. We finally normalize all the weights in the same interval (i.e. lines 18–20).

4.2. Graph formulation and optimization section — Algorithm 1 (lines 21).

In this section, we describe how we select the tags to create cluster explanations. We start from an initial set of tags that are extracted in the first part of the algorithm, which is described in the above subsection. The problem then is to filter these tags by selecting the ones that explain better the cluster content. Therefore, we frame this problem as an optimization problem where we find the optimal set of tags that maximizes an objective function within some constraints. The optimal solution is described in terms of an Integer Linear Program (ILP). The input of the program is a weighted graph between tags and clusters. An ILP solver is processing the program and it finds an optimal solution (i.e. assignment to the binary variables) that maximizes the objective function and is admissible respect to the constraints. The solution is a filtered weighted graph where we use the selected edges between tags and clusters to create the cluster explanations (i.e tags associated with a cluster). We use the same notations introduced in the above sections, plus the following ones for the weighted graph :

- with $edge_{t,c}(i,j)$ we refer to the edge in the weighted graph used to connect the cluster i with the tag j,
- with $edge_{t,t}, (i,(r,p))$ we refer to the edge in the weighted graph used to connect the tag r and the tag p, they are both belonging to the cluster i,
- with $w_{t,c}(i,j)$ we refer to the weight for the $edge_{t,c}(i,j)$, and it is used to represent the strength of the relation between a tag and a cluster,
- with $\mathcal{T}_{c_i}$ we refer to the set of tags associated with the cluster c_i,
- with $w_{t,t}(i,(r,p))$ we refer to the weight of the $edge_{t,t}(i,(r,p))$, and it is used to measure the semantic coherence among tags,
- with $w(p)$ we refer to a global weight for the tag p, and it is used to represent how important the tag p is globally.

The nodes, edges and weights are retrieved from the information stored in the data structure computed in the previous section: $Dict_{ct}, Dict^*_{ct}, Dict_{c_{tt}}$. An example of a weighted graph that can be instantiated during this section is reported in Fig. 1. The Integer Linear Program is defined as follows:

$$\begin{aligned}\text{maximize}\quad & h\sum_{i=1}^{m}\sum_{j=1}^{n} w_{t,c}(i,j)edge_{t,c}(i,j)\\ & +m\sum_{i=1}^{n}\sum_{r,p\in\mathcal{T}_{c_i}} w_{t,t}(i,(r,p))edge_{t,t}(i,(r,p))\\ & -nh\sum_{p=1}^{m} w(p)t_p\end{aligned}$$

subject to

$$\begin{aligned} & edge_{t,c}(p,j)\le c_j, \quad \forall p:1,\dots,m,\forall j:1,\dots,n\\ & edge_{t,c}(p,j)\le t_p, \quad \forall p:1,\dots,m,\forall j:1,\dots,n\end{aligned} \tag{5}$$

$$\begin{aligned} & edge_{t,t}(i,(r,p))\le t_r, \quad \forall i:1,\dots,n,\forall r,p\in\mathcal{T}_{c_i}\\ & edge_{t,t}(p,(r,p))\le t_p, \quad \forall i:1,\dots,n,\forall r,p\in\mathcal{T}_{c_i}\end{aligned} \tag{6}$$

$$\begin{aligned} & \sum_{p=1}^{m} edge_{t,c}(p,j)\ge\alpha \quad \forall j:1,\dots,n\\ & \sum_{p=1}^{m} edge_{t,c}(p,j)\le\beta \quad \forall j:1,\dots,n\end{aligned} \tag{7}$$

$$\forall p:1,\dots,m \sum_{j=1}^{n} edge_{t,c}(p,j)\le\gamma \tag{8}$$

$$\begin{aligned} & t_p\in\{0,1\} \quad \forall p:1,\dots,m,\\ & c_j\in\{1\} \quad \forall j:1,\dots,n\\ & \forall p:1,\dots,m,\forall j:1,\dots,n \quad edge_{t,c}(p,j)\in\{0,1\}\\ & \forall i:1,\dots,n,\forall r,p\in\mathcal{T}_{c_1},\dots,\mathcal{T}_{c_n} \quad edge_{t,t}(i,(r,p))\in\{0,1\}\\ & \forall p:1,\dots,m \quad w(p)\in\mathbb{R}\\ & \forall p:1,\dots,m,\forall j:1,\dots,n \quad w_{t,c}(p,j)\in\mathbb{R}\\ & \forall i:1,\dots,n,\forall r,p\in\mathcal{T}_{c_1},\dots,\mathcal{T}_{c_n} \quad w_{t,t}(i,(r,p))\in\mathbb{R}\\ & \alpha,\beta \quad constant \quad in \quad \{1,\dots,|\mathcal{T}|\} \quad \alpha\le\beta\\ & \gamma \quad constant \quad in \quad \{1,\dots,|\mathcal{C}|\}\end{aligned}$$

In the objective function of the above program, we multiply each member by some constants related to the cardinality of the variables to ensure that all the members of the functions have the same contribution. In particular, we have used the constant h to refer to the number of variables summed in the second factor of the function. The objective function has the goal to maximize the sum of the weights related to the edges between tags and clusters (first part), the sum of the weights related to the semantic coherence among the tags belonging to the same cluster (second part) and to minimize the sum of the tags that are noisy due to their global weights (third part). In the above program we have also used the following parameters: α, which represents the minimum number of tags to select for each cluster, β, which represents the maximum number of tags to select for each cluster, γ, which represents the maximum number of tags that can be

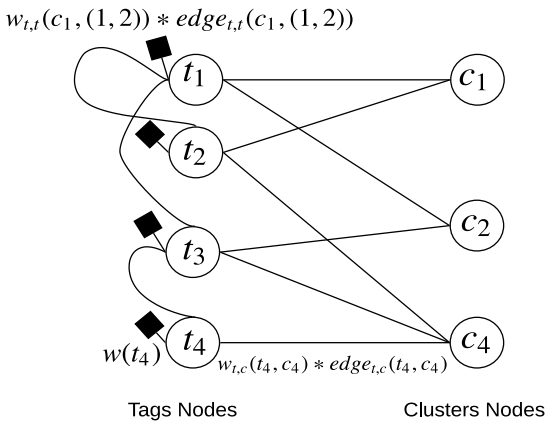

Fig. 1. An example of a weighted graph in input to the optimization stage of the Algorithm 1. The notations and the optimization stage are described in Section 4.2. For a better visualization, we insert only the notations for some nodes and edges.

shared among clusters. η, which is a threshold used to filter the numbers of edges among the tags ($edge_{t,t}$) by considering only the ones that are more significant for representing the semantic coherence. The Constraint (5) is ensuring that if a solver is selecting an edge, it needs to select the related c_i and t_p also. The Constraint (6) is also ensuring that if a solver is selecting an edge, it needs to select the related t_r and t_p. The Constraint (7) regulates the number of tags that should be selected for each cluster. We note that $\alpha > 0$ ensures that each cluster has at least one tag. The Constraint (8) controls how many tags can be shared among the clusters.

We note that all the computations described in this section (the formulation of the weighted graph, the creation of the ILP program, and the search of the optimal solution) are represented by the subroutine *optimization* in the line 21 of Algorithm 1. In the case of an infeasible model, we have adopt a technique called "elastic programming" or "elastic filter" [24]. The core idea of this technique is to first find the constraint that makes the model infeasible and then relax it by introducing extra variables ("slack" variables) and by changing the objective function considering very high penalty cost for these new variables. For example, if we find that the second set of constraints specified in (7) makes the ILP model infeasible we change it as follows:

$$\sum_{p=1}^{m} edge_{t,c}(p,j) - \varepsilon(p,j) \leq \beta \quad \forall j : 1,\ldots,n \tag{9}$$

where $\varepsilon(p,j) \in [0,M]$ are the "slack" variables, which can get an integer value between 0 and upper bound $M \in \mathbb{N}$. We then change the objective function adding an high penalty cost for the $\varepsilon_{p,j}$ as follows:

$$\begin{aligned}\text{maximize} \quad & nmh \sum_{i=1}^{m}\sum_{j=1}^{n} w_{t,c}(i,j) edge_{t,c}(i,j) \\ & + nm^2 \sum_{i=1}^{n}\sum_{r,p\in\mathcal{T}_{c_i}} w_{t,t}(i,(r,p)) edge_{t,t}(i,(r,p)) \\ & - n^2 mh \sum_{p=1}^{m} w(p) t_p \\ & - nmh \sum_{p=1}^{n}\sum_{j=1}^{m} C\varepsilon(p,j)\end{aligned}$$

where $C \in \mathbb{N}$ is a constant used to penalize the $\varepsilon(p,)$, therefore C needs to be much greater than the other weights used in model. In the above objective function, we have also to rescale its members introducing multiplicative factors to ensure that all them are equally considered by the solver.

Algorithm 1: Explanation Discovery from External Sources (EDES) Algorithm . The algorithm and the notations are detailed in Section 4.1 (lines 1–20) and in Section 4.2 (line 21)

inputs : $\mathcal{C}$, text clusters with $|\mathcal{C}|= n$; $\mathcal{D}$ documents with tags; TopE, maximum number of initial explanations to select for each cluster; TopD, maximum number of documents to retrieve; α minimum number of tags to select; β maximum number of tags to select; γ maximum number of tags to share across the cluster; $\mathcal{O}$ ontology; η coherence threshold

outputs : $\mathcal{L}$=$\{(c_1,\mathcal{E}_1),\ldots,(c_n,\mathcal{E}_n)\}$, with $c_i \in \mathcal{C}$, and $\mathcal{E}_i$ is one of the selected explanations for c_i made by a tuples of tags and scores.

1 **begin**
2 $Dict_{ct}$={{}}; $Dict^*_{ct}$={{}}:$Dict_t$={}
3 **for** $c_i \in \mathcal{C}$ **do**
4 $\mathcal{IE}_{c_i} = score\_explanation(c_i, TopE)$
5 **for** $(wo_j, s_j) \in \mathcal{IE}_{c_i}$ **do**
6 $\mathcal{T} = retrive(wo_j, \mathcal{D}, TopD)$;
7 $r_j = \frac{1}{j}$;
8 **for** $t_p \in \mathcal{T}$ **do**
9 $Dict_{ct}[c_i][t_p]$=$Dict_{ct}[c_i][t_p] \cup \{(wo_j, r_j)\}$
10 **for** $c_i \in \mathcal{C}$ **do**
11 $\mathcal{T}_{temp} = \{\}$
12 **for** $t_p \in Dict_{ct}[c_i]$ **do**
13 $Dict^*_{ct}[c_i][t_p] = \sum_{l\in Dict_{ct}[c_i][t_p]} Dict_{ct}[c_i][t_p][l][1]$
$\mathcal{T}_{temp} = \mathcal{T}_{temp} \cup t_p$
14 **for** $t_{p1} \in \mathcal{T}_{temp}$ **do**
15 **for** $t_{p2} \in \mathcal{T}_{temp}$ *with* $t_{p2} != t_{p1}$ **do**
16 $Dict_{c_{tt}}[c_i][t_{p1}] = Dict_{c_{tt}}[c_i][t_{p1}] \cup$
$\{(t_{p2}, semantic\_coherence(t_{p1}, t_{p2}, \mathcal{O}))\}$
17 $Dict_t = inverse\_average(Dict^*_{ct})$
18 $Dict_t = normalize(Dict_t)$
19 $Dict^*_{ct} = normalize(Dict^*_{ct})$
20 $Dict_{c_{tt}} = normalize(Dict_{c_{tt}})$
21 $\mathcal{L} = optimization(Dict^*_{ct}, Dict_t, Dict_{c_{tt}}, \alpha, \beta, \gamma, \eta)$
22 return $\mathcal{L}$

5. Supervised-based clusters explanations

In this section, we describe an approach to create clusters explanations driven by target labels from a subset of annotated documents. In this context, the collection of documents is divided into two sets. The first one is annotated by humans with some target labels, the second one is without annotations, and it is partitioned in clusters. This approach has the goal of finding the explanation for each cluster in terms of the target labels used to annotate the first set of documents. We leverage the approach described in Section 4, which assumes the presence of an external knowledge-base where the documents have multiple shared tags, and the purpose is to select a subset of these tags to create an explanation for each cluster. The main difference is that in our case the annotations and explanations have particular properties: each document is annotated with one target label, and the explanations are based on the target labels. This approach

is useful in applications where we have a small subset of human annotations that are not large enough to train an accurate classifier, therefore an unsupervised technique (i.e. clustering) is applied to discover clusters independently from the annotations, and then we would explain the clusters in terms of those human annotations. The inspection of the extracted explanations can also facilitate the grouping of the data for speeding up a further annotation process. The approach consists of a data pipeline made by different components, and the related reference architecture is represented in Fig. 2. The main components are described as follows:

- *Clustering Procedure* implements the clustering algorithm. It takes in input the unlabelled documents and the outputs are the clusters.
- *Extract Dictionary* creates a dictionary where each word (i.e. key) is mapped to a set of couples, where the first element is the target label and the second is the number of times the key is associated with the first element (i.e. target label frequency). It takes in input the labelled documents and the output is the dictionary.
- *Score-based Explanation* implements one of the approaches described in Section 3 to extract clusters explanations. It takes in input the clusters and the outputs are the score-based clusters explanations.
- *ANN Word Embedding Indexer* indexes each word using their semantic representation (i.e. word embeddings). It takes in input the dictionary, and the output is the index.
- *ANN Word Embedding Search* retrieves the target labels with their related frequency using an Approximate Nearest Neighbour (ANN) search [25]. The search is done using the semantic representation (i.e. word embeddings) of the words. It takes in input the index from the above component, an input word, and the output is the target labels with their related frequencies.
- *EDES* implements the algorithm described in Section 4. It takes in input the retrieved target labels with their related frequencies, the initial score-based explanations, and a domain ontology, the outputs are the clusters explanations.

Note that the pipeline leverages the results from the previous score-based and the external knowledge-based approaches to explain the clusters of the unlabelled data using the information provided by the target labels. The ANN component is used as a retrieval mechanism to recover the information for the EDES algorithm. The EDES algorithm is also exploiting the information in the ontology to enforce a semantic coherence among the target label candidates used to explain the clusters.

6. Experiments settings

In this section, we describe the settings used to run our experiments in terms of selected datasets, text processing, and feature extraction approaches, clustering procedures, external knowledge-bases, and metrics used to evaluate the results.

6.1. Datasets

To evaluate the proposed approaches, we have used three different datasets. Each dataset has been used to study various aspects of the methods presented in this paper. Since these datasets are labelled, we have used their labels to measure the accuracy of the extracted explanations.

- *20 News Groups* (20NG)[1] is a data collection of newsgroup documents that were manually classified into one of 20 categories. Each category contains 1000 documents, for a total collection size of 20,000 documents. We use this dataset to explore the impact of the parameters on the accuracy of the presented methods and to study how the quality of the clusters could influence the results. This dataset is also used in [3], but it is not clear from paper what the authors have used as ground truth labels for the evaluation in case of multiple words. For example, the class "talk.politics.mideast" could contain different labels or combinations of them: "talk, politics, mideast", "mideast", "talk", etc. In this case, we choose the most specific ones without considering the abbreviations, for example "talk.politics.mideast" label becomes "politics".
- CLUTO datasets consists of multiple datasets extracted from the CLUTO [26] website.[2] In particular, we download from the website only eight datasets, which are the ones whose labels express concepts that can be used in our evaluation. For example we did not consider datasets where the labels where numeric values. These datasets are used in literature to evaluate clustering algorithms [27,28]. The datasets are saved as sets of BOW files with only stemmed words. We use these datasets to evaluate the differences between the CDTF 3.1 and CC 3.4 scores. The stemmed words cannot be used as search keys, therefore we are not able to use these datasets to evaluate the approach described in Section 4, which uses external knowledge-bases.
- *BBC dataset* (BBC)[3] is a collections of 2225 documents from the BBC news website corresponding to stories in five topical areas from 2004–2005: "business", "entertainment", "politics", "sport", "tech". We use this dataset to compare the results between a scored-based approach and the one based on an external knowledge-base. We used also this dataset to compare our approaches with the Latent Dirichlet Allocation (LDA) [29] due to the domains and the diversity of its topics.
- *Intent Clinc dataset* (IntentC)[4] is a dataset presented in [30], and it consists of 22,500 crowdsourced queries (i.e. small sentences) covering 150 intents, which can be grouped into 10 general domains. We use this dataset to evaluate the quality of the cluster explanations based on the supervised approach. In our analysis, we did not consider the label named as "out-of-scope" and we removed the samples related to the following labels: "meta", "work", "small talk", "utility" and "home" due to their smaller size compared to the others which makes them not well represented in the clusters.

We note that in the clustering labelling literature there are also evaluations based on a subsample of Open Directory Project (ODP). Unfortunately, the ODP dataset is not available any more on the web, and we are not able to replicate the same subsampling procedure used in the previous papers. All the above datasets have documents with *human labels* (i.e. *class annotations*, *target labels*, *class labels*), and the objective of our analysis is to use those labels to describe the semantic content of the clusters. We compare those annotations with the words (i.e. *predicted labels*, *predicted words*) extracted with our algorithms to validate the quality of the cluster explanations.

6.2. Textual preprocessing

The textual data are preprocessed such that we consider only noun and verb occurrences. We remove words whose length is

[1] https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html.

[2] http://glaros.dtc.umn.edu/gkhome/fetch/sw/cluto/datasets.tar.gz.

[3] http://mlg.ucd.ie/datasets/bbc.html.

[4] https://github.com/clinc/oos-eval.

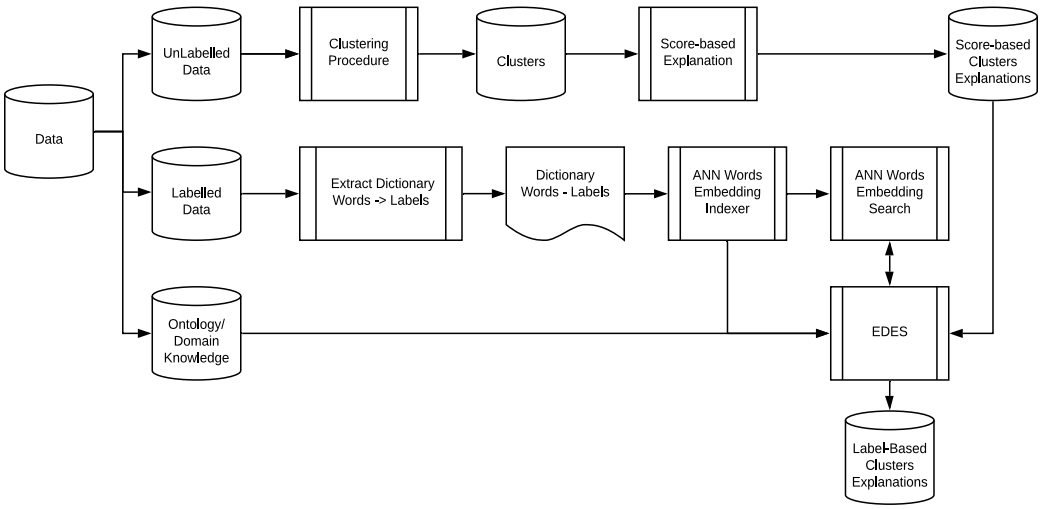

Fig. 2. Reference Architecture used to extract Supervised-based explanations.

less than 2 or greater than 10. This interval covers most of the English words as studied in [31], and we also remove words using English stopwords list.[5] We transform each word into its lemma form. The lemmatization is done with the Spacy library.[6] We have used only noun and verb because the objective of our explanation is to extract labels that are closer to the concepts expressed in the clusters, but the presented approaches are still applicable in presence of adjectives and adverbs, which could be useful for use cases where the explanation is related to opinions or sentiments of concepts contained in clusters.

6.3. Document features extraction

Words features extracted from the layers of deep neural network models are the state-of-the-art approach in the literature, in particular the ones based on transformer architectures [32]. How to combine the words features to get a document representation is still under investigation in the research, and currently, no approach shows better performances on downstream tasks on different datasets. Some approaches create document embedding directly [33] but they require a well-balanced dataset in terms of topics distribution which is always far from true in real cases. Initially, we have experimented two approaches:

- we extract the BERT embeddings [32] for each word and then we use the average pooling strategy proposed in [34] to extract the document feature.
- we extract the document-term matrix using the well-known TF–IDF and the truncated Singular Value Decomposition (tSVD) decomposition [35]. In the tSVD we use the minimum number of components that can explain at least 95% of the data variation in the training data.

In our initial experiments, we did not observe a clear advantage in the clustering quality using document features based on the first approach versus the second one, therefore we choose the simpler approach based on TF–IDF and tSVD. At the same time, we would like to point out that the objective of this work is to evaluate strategies for explaining the content of clusters and not to propose a new clustering or feature extraction algorithms. In this regard, our evaluation can be considered as lower bounds on the systems' real performance. In the case of more accurate clustering solutions and textual features, we expect that results can only improve.

6.4. Clustering procedures

We used three main strategies to cluster the documents. They are described as follows:

- *Perfect Clustering*. We group each document based on the target label. This is named a perfect clustering solution because each cluster contains the documents associated with the same class label.
- *Clustering with Noise*. We adapted the method proposed in [2] to produce noisy clusters from the perfect ones. We define a *noise level* $p \in [0, 1]$, and for each perfect cluster, we swapped each document with a probability lower than p with a document from a randomly sampled cluster. In our experiments, we chose $p = \{0.1, 0.4\}$. Due to the presence of a random sampling component, we evaluate three times the results based on this strategy and we present the average results.
- *K-means: Imperfect Clustering* We use the K-means clustering algorithm. Despite the presence of newer clustering algorithms, K-means is still among the most popular solution for practical applications for its well-studied properties, the maturity of the tools, and scalable versions of the algorithm [36]. We used the Elbow method [37] to estimate the number of clusters (K). We consider the clusters computed with the K-means as *imperfect* because most of the time they contain documents with different target labels. Therefore in the case of K-means, each cluster has multiple target labels.

In our evaluation, we consider that the perfect clustering and the clustering with noise will have one target label for each cluster, while in the case of K-means each cluster will have multiple target labels.

6.5. External knowledge bases

The methods explained in Sections 4 and 5 are based on the use of an external knowledge base. In particular, in Sections 4 and 4.1 we specify what we mean for knowledge-base and we describe how we use it to retrieve the tags used to explain the clusters. In our work, we consider:

[5] https://github.com/stopwords-iso/stopwords-en/blob/master/stopwords-en.txt.

[6] https://spacy.io/.

- Wikipedia[7] (referred also as Wiki) for the 20NG and BBC datasets. In this context, we used as tags (we named them *Wiki tags*) the categories listed at the bottom of the pages (we named them *Wikipedia categories*),
- Roget[8] for the 20NG dataset. For this knowledge-base, we created a lexicon where we mapped the English words to the Roget categories and we used them as tags (we named them *Roget tags*),
- WordNet [12], which is a semantic taxonomy of concepts. This knowledge based is used to compare our approach described in Section 4 with the one reported in [11].
- A domain-based Taxonomy[9] (referred also as IntentC-T) for the IntentC dataset. In this taxonomy, we have two layers, the first one is made by general concepts, while the second one is related to the document annotations. The annotations shared the general concepts in the first layer.

We note that our approach does not require knowledge-bases with a structure similar to Wikipedia as explained in Section 4, thus Roget and IntentC-T have a structure different from Wikipedia. We also apply the following pre-processing rules on the above tags:

1. We consider only Wiki tags made by three words, and each word is considered as a tag. This filtering step helps us to remove specific Wikipedia categories related to entities and organizations
2. Wiki tags that have less than 2 characters or that are related to disambiguation pages like "pages", "disambiguation", "article" are not considered.
3. If a word from the initial explanations retrieves an empty list of Wikipedia categories, we assume that the word as representative of the Wiki tag.
4. Wiki and Roget tags that are associated only once to a cluster are removed. We consider this as noisy tags and by removing them we reduce the number of variables in the ILP problem formulated in Section 4.2.

6.6. Evaluation metrics for cluster explanations

In [3] and [2], the authors evaluate their algorithms for cluster labelling in the context of perfect clustering and clustering with noise by measuring the match between the human label and the words extracted from their methods. They assume that a match exists if the class label and the word have the same lemma or if the class label appears in the word's Wordnet's synset [12]. If the extracted words have all the same importance, the accuracy could bias toward two extreme cases: polysemous words with multiple synsets and rare words with few synsets. Their work was also focused on the cluster labelling task, while we are interested to evaluate the quality of words sets that represent the clustering explanations. In our evaluation, we have adopted a similar approach with two main modifications: (1) we modify their evaluation metric considering the semantic similarity among distributed representations of words instead of the exact match between the extracted keywords and the target label; (2) we focus our evaluation in evaluating ordered sets of words, where the order is related to words scores. These modifications have the goal to reduce both the bias toward extreme cases and to understand the quality of the cluster explanation as ordered words set instead of measuring the impact of one word at a time.

In the case of *perfect clustering and clustering with noise* we use the *Weighted Mean Reciprocal Rank* (*WMRR@K*). Given an ordered list of k predicted labels $L^p = (p_1^p, \ldots, p_k^p)$ as explanation for a cluster, the Weighted Mean Reciprocal Rank is defined as the sum of the inverse of the rank (i.e. ranking weight) of each word in L weighted by its semantic relatedness with the target label (l^g):

$$WMRR@K(L^p, l^g) = \sum_{p_j^p \in L^p} \frac{1}{j} semr(p_j^p, l^g) \tag{10}$$

where the $semr(w_i, w_j)$ measures the semantic relatedness between two words w_i, w_j and it returns a value in [0, 1], where 1 means that the words w_i and w_j are semantically related. The semantic relatedness is computed as cosine similarity between the two distributional representations of the words (i.e. word embeddings). We use pre-trained word embeddings on Wikipedia.[10] The $semr(w_i, w_j)$ helps us to have an evaluation that is less biased toward polysemous and rare words because each word will have just one embedding, and it also takes into account the "semantic meaning" of the word instead of computing an exact match. We also note that the semantic relatedness could also be biased by the type of data used to train the embedding model to mitigate this risk we have used a model trained on a general, diverse and large datasets Wikipedia and Common Crawl.[11] The introduction of the ranking weight helps us also to consider the importance of the extracted word in the evaluation of the cluster explanation too. The above measure is referred to each cluster, and in order to summarize the accuracy among the different clusters, we have used the average of *WMRR@K* among all the clusters.

In the case of *imperfect clustering*, where each cluster can contain documents belonging to different target labels, we introduce *WMRR**, which is modification of the above *WMRR*. The *WMRR** considers the proportion of the documents assigned to each target label. It is defined as follows:

$$WMRR^*@K(L^p, L^g) = \sum_{l^g \in L^g} \sum_{p_j^p \in L^p} w_d(l^g) \frac{1}{j} semr(p_j^p, l^g) \tag{11}$$

L^g is the set of the target labels assigned to a cluster, and $w_d(l^g)$ is the proportion of documents assigned to the target label l^g.

We note that in the case of clusters explanations with supervision the *semr* function is based on the exact match between the predicted word and target label instead of the cosine similarity between their embeddings. This is due to the fact that we are interested to evaluate cluster explanations based on an ordered set of target labels instead of words.

7. Experiments

We divide the discussion of the experiments in different sections. In Section 7.1, we present the experiments related to the score-based and the external knowledge-base methods using the 20NG dataset; in Section 7.3 we present a comparison between the knowledge-base method and the Latent Dirichlet Allocation (LDA) [29] using the BBC dataset; in Section 7.4 we present the results related to the supervised approach using the ItentC dataset. We present the results in terms of the metrics presented in Section 6.6 along with a *confidence interval*, which is computed as +/- twice the standard deviation of the metrics values. In the case of noisy clusters, we repeat the experiments three times and we average the results in order to take into account the random sampling strategy used to swap documents across the clusters based on the noise levels. In the tables related to the knowledge-base approach (EDES algorithm Section 4), we

[7] We use the Wikipedia API — with version 01-04/2020.
[8] http://www.roget.org/.
[9] https://github.com/clinc/oos-eval.
[10] The selected words embeddings model can be downloaded from http://magnitude.plasticity.ai/glove/heavy/glove.6B.300d.magnitude.
[11] https://commoncrawl.org/.

also used the following notations: *KB@(a-b-c)*, where *KB* is the external knowledge-base used in the algorithm, and we use *a*, *b*, and *c* to refer to the value of the parameters used in the constraints defined in Section 4.2: *a* is for the α value, *b* is for the β value, and *c* is for the γ value. For all the evaluations of the EDES algorithm, we also set $\eta > 0.5$ as lower bound threshold to accept the edges among the clusters in the weighted graph. These edges are used to enforce the semantic coherence among the tags and this threshold helps us to consider only the edges that really express their semantic connection. In the case of a score-based method, we used *x* in the *@x* notation in the results to refer to the *K* in the *WMRR@K* and *WMRR*@K*. For all the experiments related to the EDES algorithm, we use the Top-20 CDTF score-based method as input for the initial explanation, which means that we set Top-E equal to 20 in the EDES algorithm. We select the CDTF score-based explanations because they provide better results compared to the baselines as described in Section 7.1.1. For the optimization stage (Section 4.2) of the EDES algorithm, we used the formulation without relaxing the constraints if not stated differently. We note also that in some experiments for extracting the initial labels we use score mechanisms different from CDTF like CC and LDA to have a more comprehensive comparison.

7.1. Experiments with 20NG dataset

We used the 20NG dataset to evaluate the clusters explanations methods presented in Sections 3 and 4. We also present a user study to evaluate the differences between a score-based approach versus a knowledge-base one from the perspective of the user's preferences. For the evaluations, we use the metrics introduced in Section 6.6. In particular, in this dataset is evident the limit of having the exact match between the predicted and the target labels as proposed in [3]. For example, in Table 1 we report in two different contexts the ratios between the number of documents whose words match at least once the target label and the total number of documents. The first ratio considers the documents from perfect clusters (column a), while the second takes into account the documents from the whole dataset (column b). Table 1 shows that the presences of the target labels are quite low in both cases: in the case of perfect clustering, we have an average around 10%, while for the whole dataset we have an average around 2%. This means that target labels are simply rare words within the 20NG datasets and associating a label to a document is not a process driven by the frequency of the words candidates. For each target label, we considered all the words (i.e. talk.politics.guns becomes three words "talk", "politcs", and "guns") to compute the above statistics. The introduction of the *semr* function in our metric has the role to overcome the above limitation considering the "semantic overlap" between the target labels and the words selected as explanations. The *semr* function also uses the embeddings trained on large datasets (Wikipedia and Common Crawl), and the selected model[12] covers $\sim$ 91% of the words extracted from the 20NG dataset. This high coverage guarantees us to always have a semantic representation for the cluster explanations. In all the experiments involving the K-means clustering of this dataset, we estimated the number of clusters (K) equal to 15 using the Elbow method.

[12] The selected words embeddings model can be downloaded from http://magnitude.plasticity.ai/glove/heavy/glove.6B.300d.magnitude.

Table 1
The table depicts the proportion of documents in the 20NG dataset whose words match at least once the target label. The proportions are computed grouping the documents in two different contexts: (a) documents within the perfect clusters and (b) documents in the whole dataset.

	Local Ratio (a)	Global Ratio (b)
Average	**0.118**	**0.026**

7.1.1. Evaluation of the cluster explanations based on scores method

In this subsection, we describe the experiments to evaluate the different approaches for extracting cluster explanations based on scores. In particular, we report the results for the CDTF 3.1, RF 3.2, SJD 3.3, CC 3.4 scores and we also include other two baselines: Embedding Similarity (ES) and Inverse Document Frequency (IDF). The Embedding Similarity baseline is computed by selecting as cluster explanations the words that have the highest similarity with the target label. The similarity is computed using the *semr* function introduced in the above section. The Inverse Document Frequency baseline is computed by selecting as cluster explanations the words that have the highest Inverse Document Frequency. The Inverse Document Frequency is computed considering only the documents within the same cluster. We note that the ES baseline implicitly maximizes the WMRR metric, while the IDF baseline prefers rare words as cluster explanations. We report the results for all the three clustering procedures: perfect clustering (Table 2), clustering with noise level $p = 0.1$ (Table 3), clustering with noise level $p = 0.4$ (Table 4), K-Means (Table 5). We also evaluate the statistical differences using an independent t-test. We report the *WMRR@K* and *WMRR*@K* results for different values of $K \in \{5, 10, 20\}$ (i.e. number of words for explanations). In particular, we highlight the value of *K* equal to 10 to facilitate the visual comparison across the tables. In those experiments we found that:

- In the case of the *Perfect Clustering*, the CDTF and CC performed better than the other methods with statistical significance (p-values<0.05), while the differences among the RF, SJD and IDF results are not statistically robust enough (p-values≥ 0.05).
- In the case of the *Clustering with noise*, when the noise level (p) is equal to 0.1, the CDTF and CC method performs better than the others with a statistical significance (p-values<0.05) across all the explanations of different sizes. By contrast, the results of the RF, SJD, and IDF do not present statistically significant differences in the *WMRR* values. We note also that the *WMRR* results across all the methods start to be closer (the differences are not statistically significant) if we increase the noise level ($p = 0.4$) and we increase the number of words in the explanations (K) to 10, 20.
- In the case of the *K-means*, we observe a drop in the results due to the fact that we have less homogeneous clusters (i.e. documents belonging to different labels). In this context, we used the *WMRR** and we did not find a statistically significant difference among CDTF, CC, RF, and SJD (p-values≥ 0.05), while the results are statistically different from the IDF baseline.
- We did not find statistical significance differences between CDTF and CC (p-values≥ 0.05) across the different clustering strategies in this dataset. We observe that the average value across all the clusters of CDTF is slighter higher than the CC ones.

We also notice that the confidence intervals in *WMRR* and *WMRR** are quite large, which means that in some clusters the explanations are not semantically similar to the human labels of

Table 2
The table describes the statics of the *WMRR* (average value +/- twice the standard deviation) computed across all the clusters explanations extracted from the **perfect clusters** of the 20NG dataset. The results are related to the scores methods presented in Section 3 and to the two baseline Embedding Similarity (ES) and Inverse Document Frequency (IDF) described in Section 7.1.1. We use x in the @x notation to refer to the K in the *WMRR@K*.

Method	Perfect Clustering - *WMRR*
CDTF	@1: 0.435 +/- 0.588 — @5: 0.864 +/- 0.826 **@10: 1.029 +/- 0.914** — @20: 1.149 +- 0.991
RF	@1: 0.180 +/- 0.273 — @5: 0.494 +/- 0.350 **@10: 0.621 +/- 0.405** — @20: 0.727 +- 0.471
SJD	@1: 0.160 +/- 0.227 — @5: 0.389 +/- 0.407 **@10: 0.492 +/- 0.467** — @20: 0.599 +- 0.563
ES	@1: 0.975 +/- 0.153 — @5: 1.797 +/- 0.328 **@10: 2.154 +/- 0.412** — @20: 2.479 +- 0.459
IDF	@1: 0.180 +/- 0.273 — @5: 0.431 +/- 0.490 **@10: 0.549 +/- 0.531** — @20: 0.724 +- 0.595
CC	@1: 0.377 +/- 0.563 — @5: 0.777 +/- 0.73 **@10: 0.939 +/- 0.847** — @20: 1.07 +-0.947

Table 3
The table describes the statics of the *WMRR* (average value +/- twice the standard deviation) computed across all the clusters explanations extracted from the **noise clusters** ($p = \mathbf{0.1}$) of the 20NG dataset. The results are related to the scores methods presented in Section 3 and to the two baseline Embedding Similarity (ES) and Inverse Document Frequency (IDF) described in Section 7.1.1. We use x in the @x notation to refer to the K in the *WMRR@K*.

Method	Noise Clustering ($p = 0.1$) - *WMRR*
CDTF	@1: 0.445 +/- 0.634 @5: 0.765 +/- 0.863 **@10: 0.846 +/- 0.921** @20: 1.086 +/- 1.090
RF	@1: 0.180 +/- 0.273 @5: 0.449 +/- 0.365 **@10: 0.574 +/- 0.411** @20: 0.689 +/- 0.474
SJD	@1: 0.142 +/- 0.173 @5: 0.375 +/- 0.405 **@10: 0.476 +/- 0.502** @20: 0.587 +/- 0.602
ES	@1: 0.979 +/- 0.141 @5: 1.801 +/- 0.312 **@10: 2.157 +/- 0.394** @20: 2.479 +/- 0.434
IDF	@1: 0.180 +/- 0.273 @5: 0.401 +/- 0.497 **@10: 0.526 +/- 0.525** @20: 0.686 +/- 0.591
CC	@1: 283 +/- 0.648 @5: 0.423 +/- 0.895 **@10: 0.48+/−0.842** @20: 0.536+/- 1.083

Table 4
The table describes the statics of the *WMRR* (average value +/- twice the standard deviation) computed across all the clusters explanations extracted from the **noise clusters** ($p = \mathbf{0.4}$) of the 20NG dataset. The results are related to the scores methods presented in Section 3 and to the baselines Embedding Similarity (ES) and Inverse Document Frequency (IDF) described in Section 7.1.1. We use x in the @x notation to refer to the K in the *WMRR@K*.

Method	Noise Clustering ($p = 0.4$) - *WMRR*
CDTF	@1: 0.335 +/- 0.649 @5: 0.568 +/- 1.032 **@10: 0.583 +/- 1.126** @20: 0.568 +/- 1.159
RF	@1: 0.180 +/- 0.273 @5: 0.381 +/- 0.450 **@10: 0.487 +/- 0.544** @20: 0.593 +/- 0.608
SJD	@1: 0.152 +/- 0.198 @5: 0.381 +/- 0.400 **@10: 0.499 +/- 0.515** @20: 0.611 +/- 0.624
ES	@1: 0.980 +/- 0.177 @5: 1.793 +/- 0.326 **@10: 2.142 +/- 0.395** @20: 2.458 +/- 0.425
IDF	@1: 0.180 +/- 0.273 @5: 0.338 +/- 0.402 **@10: 0.432 +/- 0.462** @20: 0.584 +/- 0.542
CC	@1: 0.283 +/- 0.695 @5: 0.423 +/- 1.085 **@10: 0.550+/- 1.266** @20: 0.59 +/- 1.363

Table 5
The table describes the statics of the *WMRR* (average value +/- twice the standard deviation) computed across all the clusters explanations extracted from the clusters of the 20NG dataset computed with the **K-means** algorithm. The results are related to the scores methods presented in Section 3 and to the baselines Embedding Similarity (ES) and Inverse Document Frequency (IDF) described in Section 7.1.1. We use x in the @x notation to refer to the K in the *WMRR*@K*.

Method	K-means - *WMRR**
CDTF	@1: 0.067 +/- 0.123 — @5: 0.131 +/- 0.195 **@10: 0.153 +/- 0.231** — @20: 0.231 +/- 0.479
RF	@1: 0.054 +/- 0.175 — @5: 0.125 +/- 0.269 **@10: 0.145 +/- 0.276** — @20: 0.174 +/- 0.321
SJD	@1: 0.045 +/- 0.086 — @5: 0.111 +/- 0.253 **@10: 0.138 +/- 0.293** — @20: 0.169 +/- 0.369
ES	@1: 0.820 +/- 0.471 — @5: 1.494 +/- 0.679 **@10: 1.781 +/- 0.748** — @20: 2.047 +/- 0.812
IDF	@1: 0.162 +/- 0.256 — @5: 0.317 +/- 0.390 **@10: 0.382 +/- 0.450** — @20: 0.461 +/- 0.524
CC	@1: 0.038 +/- 0.137 @5: 0.098 +/- 0.285 **@10: 0.116+/−0.327** @20: 0.204+/- 0.0351

the documents. This is due to two factors. The first one is related to the distribution of the *semr* outputs, which are squashed at the extremities of the interval [0, 1]. The second factor is that the true labels are quite different in terms of specificity and generality, for example, we have target labels that are quite general "auto" and "politics" or quite specific like "encrypt", while the extracted explanations based on scores are more specific. These differences between specific and general words can have an impact on the distribution of the similarity scores. The condition underlying the first factor can be clearly seen in Fig. 4, where we plot the average similarity for the 200 most similar words for each target label (Top-200) in the context of perfect clustering. In other words, for each perfect cluster, we select the 200 most similar words to the target label, and then we average the result across all the clusters. The plot shows that we already have a significant drop in the similarity values in the first 10 or 20 words. Therefore, according to the selected embeddings model the semantic similarity score does not have multiple degrees of interpretations, it mostly assumes that "two words are similar or not". In Tables 2 and 4, we saw that the CDTF and CC results in terms of average values are superior respect to the other methods. This difference is less evident when the clusters are noisier, for example in Tables 4 and 5. In Fig. 3, we plot the results of the *VMRR* vs the Adjusted Rand Index (ARI) for the perfect and noise clustering approaches. The Adjusted Rand Index is a measure used to evaluate the quality of the clusters using annotated data [38]. From this comparison, we can deduce as a rule of thumb that CDTF and CC are robust techniques for extracting clusters explanations when the clustering procedure has an Adjusted Rand Index (ARI)$\geq$ 0.4. For lower values a simpler IDF-based approach could still provide some insights of the content of the cluster. We are aware that the computation of the Adjusted Rand Index requires annotated data, but an approximation of the ARI can be obtained by labelling a subset of the data.

7.1.2. Evaluation of the cluster explanations based on external knowledge-base

In this subsection, we present the evaluation of the EDES algorithm described in Section 4. The experiments above show that the CDTF and CC provides better explanations compared to the other score-based methods, and the differences are statistically significant in the context of good quality clusters (ARI$\geq$ 0.4) with CDTF having average values higher than CC. Therefore we choose the first 20 words from the CDTF method as input for the EDES algorithm (this input is named initial explanation in Section 4.1). For the external knowledge-base, we use Wikipedia, and we consider two cases referred to as Wiki-1 and Wiki-5, which means that we select the Top-1 or the Top-5 results from the Wikipedia search respectively. According to the notation

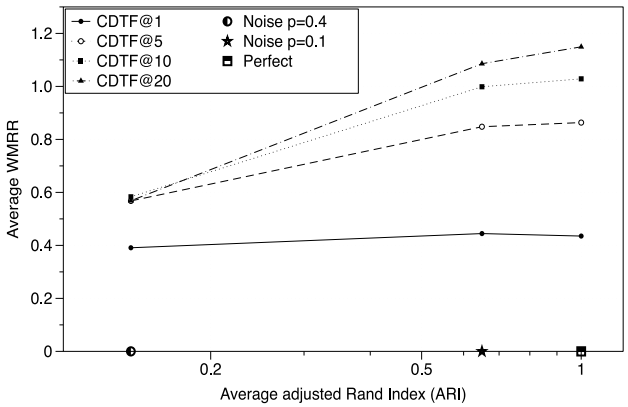

Fig. 3. The figure shows the relationship between the *WMRR* results and the *ARI* for the CDTF (Section 3.1) method in three different contexts: perfect clustering, clustering with noise level 0.1 and 0.4. We note that we did not include the K-means results because the *WMRR* is computed differently (*WMRR**) due the proportion of documents assigned to each label.

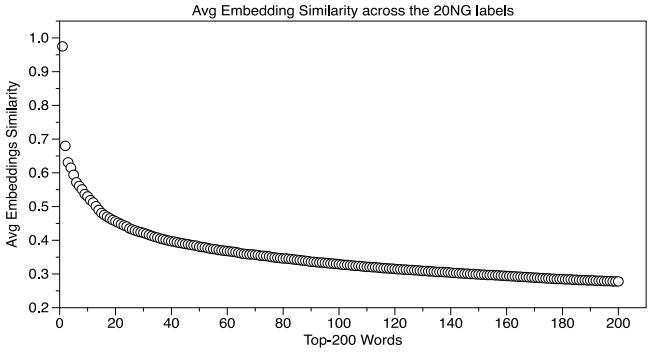

Fig. 4. The figure shows the average similarity of the Top-200 words across the labels in the 20NG dataset. We note that the embedding similarity values are squashed on the extremes of the range [0,1].

KB@a-b-c introduces in Section 7, KB can be Wiki-1 or Wiki-5. In the tables, we report also the results of the pairwise t-test to validate the statistical significance in the differences. We used the column Null Hypothesis (N.H.) to report if we accept (True, p-value ≤ 0.05) or not (False, p-value < 0.05) the Null Hypothesis. The Null Hypothesis is accepted (True) if there are not statistically relevant differences in the value of the *WMRR* between the two considered methods. The statistical test is done using an independent t-test. The solver used for the solution of the ILP problem is the default one in the Python library Pulp.[13] In Tables 6–9 we report our results. The results show that the EDES algorithm provides average results that are higher than the CDTF, and this is consistent among the different clustering contexts. Due to the high variance in the results (large confidence intervals), the t-test accepts the null hypothesis (N.H.=True), indicating that the distribution of the *WMRR* scores across all the labels is not different from the CDTF. At the same time, we note that the results of the EDES algorithm have higher averages, which means that the cluster explanations are closer to the target labels or they are made of words that are more general therefore semantically closer to the target labels. This aspect is evident in Tables 10 and 11, where we report in the context of perfect clusters and clusters with noise ($p = 0.4$) the Top-5 explanations from the EDES algorithm, the CDTF and CC methods as well as the method described in [11]. In particular, we choose the ones related to the

13 https://pythonhosted.org/PuLP/.

Table 6
The table describes the average *WMRR* for the clusters explanations from the 20NG dataset using the Algorithm in Section 4 based on the initial explanation extracted from the CDTF method (Section 3.1). In these results, we refer to the **perfect clustering** method. N.H. column refers to acceptance (True, p-value ≤ 0.05) or not (False, p-value < 0.05) of the null hypothesis. A null hypothesis is accepted if there is no statistical significant difference in the *WMRR* values between the selected method based on an external knowledge-base and the one based on the CDTF score. The notation used in the methods is described in Sections 7 and 7.1.2.

Method	Average *WMRR* - PC	N.H.
Wiki-1@1-1-1	0.507 +/- 0.625	True
Wiki-1@5-5-2	0.943 +/- 0.919	True
Wiki-1@10-10-5	**1.122 +/- 0.984**	**True**
Wiki-5@1-1-1	0.498 +/- 0.716	True
Wiki-5@5-5-2	0.936 +/- 0.966	True
Wiki-5@10-10-5	**1.117 +/- 1.062**	**True**
Roget@1-1-1	0.071 +/- 0.267	False
Roget@5-5-2	0.221 +/- 0.540	False
Roget@10-10-5	**0.253 +/- 0.566**	False

clusters with best and worst explanations in terms of *WMRR*. In Table 11, we report the clusters with the related target labels that are consistently in the top and bottom of the *WMRR* rank at least in three runs out of five, while the *WMRR* result refers to the single run with a *WMRR* lower value. The above choice is due the fact that the each run can have different clusters due to different injected noise, therefore different explanations. We choose these clustering approaches because they have an *ARI* ≥ 0.4, which is the rule of thumb that we have found in our previous analysis as condition to have more accurate explanations based on CDTF. In these latest tables, the EDES algorithm based on Wikipedia is able to provide explanations closer to the target labels than the CDTF, CC and [11], but for the target labels that are too general (i.e. "politics") or too specific (i.e. "encrypt") we observe a reduction of the *WMRR* due to the limit of using the selected embedding model to measure the semantic relatedness between two words as discussed in Section 7.1.1. In Table 11, we also included the results for the perfect clustering (PC) for the same label. The negative values of the *WMRR* are due to the fact that the semantic similarity function is providing negative values in the presence of words that are not contained in the embedding model. In those results, we also found that considering more search outputs from the retrieval of the Wikipedia pages (i.e. Top-5) does not improve the *WMRR*. We also note that the structure and the richness of the considered external knowledge base are making the difference, for example, the Roget knowledge-base was not providing enough content to improve the explanations of the EDES algorithm respect to the CDTF. We also compare the EDES algorithm with the approach described in [11] based on WordNet. The results of this comparison are presented in Table 12 for different clustering strategies. In order to have a fair comparison, we use the same knowledge-base (WordNet) also for the EDES algorithm. For the input tags of the EDES algorithm, we retrieve all the hypernyms of the initial labels up to levels two (the WordNet root is level 0), such that we have an input for the EDES algorithm that is similar to the ones used in [11]. In Table 12, we also report the results of the EDES algorithm in the case where we use CC instead of CDTF for the initial labels as used in [11]. The results shows that EDES algorithm outperforms in terms of *WMRR* the approach presented in [11] for all the clustering strategies. We also notice that the use of WordNet provides lower *WMRR* values in comparison of Wikipedia. We note that our approach can be applied to multiple knowledge-bases while the [11] approach is tied up to the properties of WordNet.

Table 7
The table describes the average *WMRR* for the clusters explanations from the 20NG dataset using the Algorithm in Section 4 based on the initial explanation extracted from the CDTF method (Section 3.1). In these results, we refer to the **noise clusters** with noise level (p) equal to 0.1. N.H. column refers to acceptance (True, p-value ≤ 0.05) or not (False, p-value < 0.05) of the null hypothesis. A null hypothesis is accepted if there is no statistical significant difference in the *WMRR* values between the selected method based on an external knowledge-base and the one based on the CDTF score. The notation used in the methods is described in Section 7 and Section 7.1.2.

Method	Avg *WMRR* - NC($p = 0.1$)	N.H.
Wiki-1@1-1–1	0.467 +/- 0.606	True
Wiki-1@5-5–2	0.884 +/- 0.921	True
Wiki-1@10-10-5	**1.04 +/- 1.03**	**True**
Wiki-5@1-1–1	0.626 +/- 0.794	True
Wiki-5@5-5–2	0.769 +/- 0.986	True
Wiki-5@10-10-5	**0.818 +/- 0.939**	**True**
Roget@1-1–1	0.07 +/- 0.244	False
Roget@5-5–2	0.231 +/- 0.522	False
Roget@10-10-5	**0.239 +/- 0.536**	**False**

Table 8
The table describes the average *WMRR* for the clusters explanations from the 20NG dataset using the Algorithm in Section 4 based on the initial explanation extracted from the CDTF method (Section 3.1). In these results, we refer to the **noise clusters** with noise level (p) equal to 0.4. N.H. column refers to acceptance (True, p-value ≤ 0.05) or not (False, p-value < 0.05) of the null hypothesis. A null hypothesis is accepted if there is no statistical significant difference in the *WMRR* values between the selected method based on an external knowledge-base and the one based on the CDTF score. The notation used in the methods is described in Sections 7 and 7.1.2.

Method	Avg *WMRR* - NC($p = 0.4$)	NH.
Wiki-1@1-1–1	0.319 +/- 0.644	True
Wiki-1@5-5–2	0.605 +/- 0.103	True
Wiki-1@10-10-5	**0.688 +/- 1.185**	**True**
Wiki-5@1-1–1	0.406 +/- 0.927	True
Wiki-5@5-5–2	0.520 +/- 1.039	True
Wiki-5@10-10-5	**0.558** +/- 1.075	**True**
Roget@1-1–1	0.08 +/- 0.255	False
Roget@5-5–2	0.204 +/- 0.557	False
Roget@10-10-5	**0.247 +/- 0.562**	**False**

Table 9
The table describes the average *WMRR** for the clusters explanations from the 20NG dataset using the Algorithm in Section 4 based on the initial explanation extracted from the CDTF method (Section 3.1). In these results, we refer to the K-means clusters. N.H. column refers to acceptance (True, p-value ≤ 0.05) or not (False, p-value < 0.05) of the null hypothesis. A null hypothesis is accepted if there is no statistical significant difference in the *WMRR** values between the selected method based on an external knowledge-base and the one based on the CDTF score. The notation used in the methods is described in Sections 7 and 7.1.2.

Method	Avg *WMRR** - KMeans	N.H.
Wiki-1@1-1–1	0.092 +/- 0.199	True
Wiki-1@5-5–2	0.186 +/- 0.351	True
Wiki-1@10-10-5	**0.219 +/- 0.401**	**True**
Wiki-5@1-1–1	0.088 +/- 0.196	True
Wiki-5@5-5–2	0.184 +/- 0.351	True
Wiki-5@10-10-5	**0.217 +/- 0.395**	**True**
Roget@1-1–1	0.01 +/- 0.04	False
Roget@5-5–2	0.02 +/- 0.05	False
Roget@10-10-5	**0.03 +/- 0.07**	**False**

7.1.3. User study

In this subsection, we report the results of the user study that analyses the preferences of the users between the cluster explanations provided by the CDTF score-based method (Section 3.1) and the EDES algorithm that uses Wikipedia as the external knowledge-base (Section 4). We note that from the user perspective a cluster explanation could be evaluated with respect to attributes such as the level of specificity and/or generality of its words that could be not fully captured by the *WMRR* value.

Table 10
The table contains the Top-5 explanations for some clusters from the 20NG dataset computed from the EDES Algorithm (Section 4) and the CDTF method (Section 3.1). In particular, we report only the clusters with the best *WMRR* and worst *WMRR* respect to the EDES algorithm. We include the Top-5 explanations with the highest embedding similarity with the target labels (ES). The explanations are related to the clusters extracted in the context of the **perfect clustering**.

Target labels	Method	Explanations	WMRR
christian (soc.religion. christian) (Best WMRR)	CDTF	god, jesus, church, christians, christ	0.977
	CC	sandvik, christian, jesus, kendig, kent	0.234
	Wiki-1 @5-5-2	christian, religious, christianity, god, church	1.657
	Tseng [11]	person, causal, belief, creation, handbook	0.633
	ES	christian, christianity, catholic, religious, faith	1.762
encrypt (sci.crypt) (Worst WMRR)	CDTF	key, encryption, clipper, chip, privacy	0.657
	CC	clipper, encryption, key, escrow, chip	0.401
	Wiki-1 @5-5-2	security, agency, cryptography, legal, protection	0.080
	Tseng [11]	device, act, instrumentality, event, artefact	0.131
	ES	encrypt, encryptor, encryp, encrypter, encrypted	2.028

However, the *WMRR* is designed to measure how semantically close the explanations are with the target labels, which have been chosen by humans, therefore *WMRR* is also used as a proxy for measuring how the explanations are related to the user preferences. In this user study, we compare two explanations presented in the previous analysis in the context of perfect clustering: the CDTF@10 in Table 2 and the Wiki-5@10-10-5 in Table 6. Both these approaches presented good performances in terms of *WMRR* but the t-test did not find a statistically significant difference in the values of *WMRR* (p-value > 0.05). Our user study consists of a set of 40 questions. In each question, we present to the user a document belonging to a cluster and ask them to choose between the two explanations (i.e. set of words). We select 10 target labels from the 20NG dataset related to topics of general knowledge like "religion", "politics", "auto", and we randomly sub-sample 4 documents for each question related to the same cluster (i.e. target label). The random subsampling of the documents within the clusters is constrained to use only the set of documents that have a similar length, and have at least one word in common with the presented explanations. We also constraint the sub-sampling of the documents by the condition that the words that are in common between the documents in the

Table 11
The table contains the Top-5 explanations for some clusters from the 20NG dataset computed from the EDES Algorithm (Section 4) and the CDTF method (Section 3.1). In particular, we report only the clusters with the best *WMRR* and the worst *WMRR* respect to the EDES algorithm. We include the Top-5 explanations with the highest embedding similarity with the target labels (ES). The explanations are related to the clusters extracted in the context of the **noise clustering** with noise level equal to 0.4. We also include Top-5 explanations for the same labels in the context of the perfect clustering denoted with PC.

Target labels	Method	Explanations	WMRR
baseball (rec.sport. baseball) (Best WMRR)	CDTF	game, baseball, player, team, pitch	1.391
	CDTF (PC)	game, baseball, team, player, pitch	1.389
	CC	baseball, pitcher, phillies, sox, hitter	1.651
	CC (PC)	baseball, pitching, pitcher, hitter, phillies	1.605
	Wiki-1 @5-5-2	baseball, pitching, team, game, summer	1.584
	Wiki-1 @5-5-2 (PC)	baseball, team, pitching, league, musician	1.564
	Tseng [11]	activity, person, act, artefact, contestant	0.268
	Tseng [11] (PC)	act, person, activity, event, instrumentality	0.267
	ES	baseball, basketball, mlb, football, hockey	1.842
politics (talk.politics. misc) (Worst WMRR)	CDTF	salonica, aiu, aloysius, talaat, slick	−0.22
	CDTF (PC)	president, gay, clinton, tax, government	0.687
	CC	boomer, kaldis, consent, clayton, slick	0.077
	CC (PC)	cramer, clayton, gay, petaluma, optilink	0.112
	Wiki-1 @5-5-2	aeolian, coastal, dune, geography, sedimentology	−0.04
	Wiki-1 @5-5-2 (PC)	homosexuality, lgbt, same-sex, sexuality, american	0.444
	Tseng [11]	person, ray, elasmobranch, instrumentality, cartilaginous	0.186
	Tseng [11] (PC)	district, person, country, region, administrative	0.424
	ES	politics, political, religion, culture, ideology	1.782

questions and the presented explanations need to be different, in order to ensure that the users express an unbiased choice between two different options without being driven by intersections happened by chance. We also found that the selected

Table 12
In this Table we compare the explanations in terms of *WMRR* for clusters from 20NG dataset using the EDES 4 approach and the one presented in [11]. We use different clustering strategies. In order to have a fair comparison, we use the same knowledge-base (WordNet) used in [11]. We also report the result of the EDES algorithm based on the labels extracted from the correlation coefficient presented in [11].

PC	
Method	Average WMRR@10
Wordnet@10-10-5 with CDTF	0.6273 +/- 0.544
Wordnet@10-10-5 with CC	0.6584 +/- 0.595
Tseng [11]	0.2858 +/- 0.268
NC (p=0.1)	
Method	Average WMRR@10
Wordnet@10-10-5 with CDTF	0.6735 +/- 0.590
Wordnet@10-10-5 with CC	0.6774 +/- 0.611
Tseng [11]	0.2829 +/- 0.279
NC (p=0.4)	
Method	Average WMRR@10
Wordnet@10-10-5 with CDTF	0.6214 +/- 0.472
Wordnet@10-10-5 with CC	0.6553 +/- 0.657
Tseng [11]	0.2769 +/- 0.261
K-Means	
Method	Average WMRR@10
Wordnet@10-10-5 with CDTF	0.0552 +/- 0.108
Wordnet@10-10-5 with CC	0.0559 +/- 0.103
Tseng [11]	0.0267 +/- 0.040

10 target labels are also representative of the full dataset for this study due to the fact that we did not find any statistically significant differences in the average *WMRR* between these labels and our selection (p-value < 0.05). The user study involved 3 persons and it was done in a constraint environment in order to avoid interferences between users during the study. We provide to each user an Excel file with the 40 questions. For each user, we randomize both the order of the questions and the position of the columns assigned to the explanations. We ask the users to select 5 words among the explanations to better justify their choice. The results of this user study are presented in Table 13. The results show a substantial agreement (Fleiss' Kappa value between 0.61–0.80) in preferring the CDTF explanations with 85% preferences. We also compute for each question a ratio between the 5 words selected by the raters and the Top-5 words contained in the selected explanation (named Average RWI - Ratio Words Intersection), and we found an average of $0.680 +/- 0.142$ for the CDTF method across all the questions compared to the $0.297 +/- 0.355$ for the Wiki-5@10-10-5. These results show that users prefer the CDTF explanations due to the specificity of their words respect to the generality of the Wikipedia tags (0.85 vs 0.15), and the ranking of the CDTF is also meaningful for the users (0.68 vs 0.297). This users' preference could not be well captured by the *WMRR* results and by the t-test due to the fact that if a word is more specific its embedding representation is on average closer to fewer words increasing the variance in the *WMRR* results. This study also shows that the evaluation of the explanations could include different factors such as the user preferences versus a specific content, while an explanation algorithm is mostly designed with the purpose to generalize the semantic description of the clusters. These experiments also show what are aspects that the CDTF or the EDES algorithm favour in the selection of the clusters explanations.

7.2. Experiments with CLUTO datasets

We run a set of experiments with the CLUTO datasets to evaluate the differences between the CDTF 3.1 and CC 3.1 scores. The

Table 13
Results of the User Study related to user preferences between the CDTF method in Section 3.1 (@10) and EDES algorithm in Section 4 (Wiki-5@10-10-5) both in the context of perfect clustering using the 20NG dataset. RWI (Ratio Words Intersection) is the ratio between the 5 words selected by the users and the Top-5 words belonging to the selected explanation.

Metrics	Results
Inter-rater agreement (Fleiss' Kappa for 3 Raters)	0.673, p-value <0.0001
Preferences for @10 CDTF hline Preferences for Wiki-5@10-10–5	0.85 0.15
Average RWI in CDTF	0.680+/- 0.142
Average RWI in Wiki-5@10-10–5	0.297 +/- 0.355

Table 14
The table describes the statics of the *WMRR* (average value +/- twice the standard deviation) computed across all the Top-10 clusters explanations for the CLUTO datasets extracted using the CDTF 3.1 and CC 3.4 scores across all the **noise clusters** ($p = \mathbf{0.1}$).

DB	CDTF	CC	*p-value*	ADRI
reviews	2.475 +/- 1.146	2.518 +/- 1.008	0.939	0.585
wap	2.661 +/- 0.727	2.174 +/- 1.514	0.032	0.785
re1	2.876 +/- 0.185	2.859 +/- 0.221	0.617	0.775
hitech	2.364 +/- 0.803	2.383 +/- 1.262	0.967	0.625
lal2	2.901 +/- 0.096	2.778 +/- 0.264	0.179	0.667
k1b	2.929 +/- 0.000	2.766 +/- 0.327	0.134	0.709
sports	2.929 +/- 0.000	2.842 +/- 0.261	0.217	0.704
mm	1.933 +/- 0.200	2.879 +/- 0.100	0.013	0.472
news20	2.773 +/- 0.530	2.608 +/- 0.932	0.275	0.694

Table 15
The table describes the statics of the *WMRR* (average value +/- twice the standard deviation) computed across all the Top-10 clusters explanations for the CLUTO datasets extracted using the CDTF 3.1 and CC 3.4 scores across all the **noise clusters** ($p = \mathbf{0.4}$).

DB	CDTF	CC	*p-value*	ADRI
reviews	1.111 +/- 1.595	2.404 +/- 1.485	0.168	0.387
wap	2.323 +/- 1.289	1.784 +/- 1.779	0.067	0.568
re1	2.656 +/- 1.137	2.541 +/- 1.349	0.584	0.528
hitech	1.413 +/- 1.471	2.085 +/- 2.218	0.415	0.469
lal2	1.508 +/- 1.084	2.740 +/- 0.295	0.009	0.430
k1b	2.537 +/- 0.743	2.747 +/- 0.364	0.412	0.408
sports	1.214 +/- 2.065	2.299 +/- 2.301	0.197	0.462
mm	0.000 +/- 0.000	2.731 +/- 0.111	0.0004	0.198
news20	2.778 +/- 0.409	2.595 +/- 1.004	0.275	0.517

results are presented in Tables 14 and 15. For each dataset, we report the ADRIs and the *p-values*. We only run the experiments with noise clusters to have a more realistic pictures, and we not use the K-means clustering strategy because it is not able to provide meaningful clusters for those datasets (ADRI is lower than 0.3 in most of the datasets), and finding the best clustering algorithms for each dataset is not in the scope of this work. According to these results, we do not find a statistical significance difference between the CDTF and CC scores using a t-test. In some datasets, we observe that CDTF is able to be more specific (which means more focused on entities) in the selection of the labels across the datasets which can be useful to retrieve useful tags from an external knowledge. Potentially, this is due the fact that it considers a term-frequency component. We are not able to use these datasets for evaluating the EDES approach 4 because the words are stemmed, therefore they cannot be used as search keywords for the knowledge base. We note that for this evaluation we manually annotate the extracted labels (i.e. stemmed words) as relevant respect to the ground true labels in the cluster. Then, we use an indicator function that outputs 1 or 0 depending on whether the label is relevant or not as *semr* function in the *WMRR*.

7.3. Experiments with BBC dataset

In this section, we describe the results related to the cluster explanations of the BBC dataset computed using our methods and the Latent Dirichlet Allocation (LDA). In particular, we report the results of the following experiments:

- In Table 16, we compare the CDTF-based explanations (Section 3.1) vs the EDES-based one (Section 4) using Wikipedia as knowledge-base. The clusters are extracted using the K-means algorithm with K (i.e number of clusters) equals to 6 (estimated with the Elbow method). We report the *WMRR**@10 results for the CDTF, and we use the Wiki-1@10-10-5 setting for the EDES algorithm. The meaning of the Wiki-1@10-10-5 setting is explained in Section 7. In particular, Wiki-1 means that we select the Top-1 results from the Wikipedia search, while 10-10-5 means that we fix the length of the explanations to 10 words for each cluster, and each cluster can share not more than 5 words across the other clusters. We also set Top-E equal to 20 in the EDES algorithm as discussed in Section 7. We select these configurations because they provide much more context for the explanations as discussed in the experiments with the 20NG dataset. The clusters have an Adjust Random Index (ARI) of 0.431. For each cluster, we report also the proportion of the target labels and the explanations.
- In Table 17 we compare the results of the CDTF-based method with the LDA. LDA is a well-known technique applied in the industry to extract topics from the collection of documents. LDA can also be considered as a score based approach to extract explanations (i.e. ordered set of words), but clustering and explanations are part of the same algorithm. We need also to consider that LDA is a "soft" clustering technique because a document can be assigned to multiple topics, therefore to have a fair comparison we apply an "hard" clustering approach where we select the most probable topic for each document. We name this clustering approach *LDA with the most probable assignment*. Using this approach, we have a set of clusters, where each cluster has a topic (i.e. words with probabilities) that we assume as cluster explanation (*named LDA-based explanation*), and we use for the comparison. In Table 17, we also report the target labels for each cluster along with the Top-10 explanations for both the CDTF and the LDA methods. In the case of the LDA, we use the words assigned to the topics. The results are reported in terms of *WMRR**. In the LDA, we set the number of topics equal to 6 similar to what we did for the K-means. In the LDA code[14] we set the number parameters like the number of iterations and passes such that the quality of the clusters in terms of ARI was greater than 0.4, and we also have lower perplexity and higher coherence [39]. In particular with our settings (i.e iteration=50, passes=100) we got an ARI equal to 0.478.
- In Table 18 we compare the explanations obtained from the EDES algorithm using Wikipedia as an external knowledge-base with two different initial explanations (i.e. input). The first one is based on the CDTF method, while the second one is based on the LDA. We assume the same length of initial explanations for both the CDTF-based and LDA-based approaches. For the EDES algorithm, we use the same settings described above. We report the results in terms of *WMRR**.
- In Table 19 we also compare the EDES algorithm with the approach described in [11] in terms of *WMRR*. In order to

[14] https://radimrehurek.com/gensim/models/ldamodel.html.

Table 16
The Table reports the results of the CDTF and the EDES algorithm for the BBC dataset using Wikipedia as an external knowledge-base. We also use the CDTF-based approach for the initial explanation of the EDES algorithm. The documents are clustered using K-means with K=6 (estimated with the Elbow method). The Table presents the results in terms of *WMRR**@10 for the CTDF-based method and the EDES algorithm. For the EDES algorithm, we use the setting Wiki-1@10-10-5 which is explained in Section 7.3.

Cluster index	Target labels	Explanations CDTF Top-10 Words	WMRR* CDTF	Explanations Wiki-1@10-10-5 Top 10 Words (Input: CDTF)	WMRR* Wiki-1@10-10-5 (Input: CDTF)
0	"entertainment": 0.970, "tech": 0.029	film, actor, award, oscar, actress, star, aviator, comedy, festival, hollywood	0.394	television, theatrical, art, award, aviation, los, angeles, oscar, american, star	0.502
1	"business": 0.416, "entertainment": 0.242, "sport": 0.046, "politics": 0.230, "tech": 0.064	yukos, airline, album, lse, gazprom, glazer, fiat, boerse, worldcom, ebbers	0.051	american, comedian, actress, university, fiat, living, people, yukos, industry, rosneft	0.114
2	"business": 0.009, "politics": 0.990	labour, election, party, blair, brown, howard, tory, tories, chancellor, tax	0.438	party, election, american, labour, politics, government, parliamentary, voting, minister, leadership	0.630
3	"business": 0.005, "entertainment": 0.003, "sport": 0.914, "politics": 0.005, "tech": 0.0701	game, match, cup, player, injury, rugby, coach, chelsea, season, arsenal	0.171	game, equipment, german, hungarian, match, player, pain, trauma, rugby, coach	0.154
4	"business": 1.0	economy, growth, rate, rise, price, deficit, economist, fall, dollar, export	1.113	economics, economy, growth, consumption, rate, rise, deficit, science, economist, social	1.138
5	"business": 0.0129, "entertainment": 0.006, "politics": 0.006, "tech": 0.974	user, phone, technology, software, broadband, device, mobile, gadget, program, apple	0.195	technology, mobile, telecommunication, software, digital, telephony, user, phone, radio, science	0.294
AVG +/-2*STD			0.3942 +/- 0.695		0.4726 +/- 0.698

have a fair comparison we used WordNet as knowledge-base for the EDES algorithm as in [11]. In this dataset, the search process on WordNet is done similar to what we describe in the 20NG evaluation 7.1.2. We observe that our approach outperforms the one presented in [11] in line of what we observe in the 20NG evaluation.

In Table 16, we observe the same results of the 20NG dataset, where the EDES algorithm has an *WMRR** average higher than the CDTF-based approach, however, the confidence interval is too large to derive a clear difference in terms of *WMRR**. In Table 17 the LDA-based approach presents a small improvement (i.e. absolute difference $\sim$ 0.02) in terms of *WMRR** compared to the CDTF-based technique. We note also that the words in the LDA-based approach are less coherent than the CDTF-based due to the fact that in the LDA results documents can be part of multiple clusters (i.e topics). For example, if we observe the target labels proportions in Table 17, clusters 2, 4, and 5 are mostly described by one topic, and in these cases, the differences in terms of *WMRR** between the CDTF-based and LDA-based techniques are smaller (i.e absolute difference $\sim$ 0.01). We also observe the same similar difference in Table 18. We also note that LDA favours words that have a higher frequency in the documents because its inference process is mostly driven by the number of co-occurrence of the words, therefore LDA results are biased toward more frequent words in the corpus, which we can assume that are more general because they are used across the domains. These differences between the different levels of specificity and generality in the explanations can be inferred by the words presented in Tables 17 and 18. In order to further analyse this aspect, we

Table 17
The Table compares the *WMRR** results between two score-based approaches the CDTF and the Latent Dirichlet Allocation (LDA) for the BBC dataset . The documents are clustered using the **LDA and the most probable assignment** described in Section 7.3. The results refer to the Top-10 explanations.

Cluster index	Target Labels	Explanations CDTF Top-10 Words	*WMRR** CDTF	Explanations LDA Top 10 words	*WMRR** LDA
0	"business": 0.024, "entertainment": 0.860, "politics": 0.006, "tech": 0.109	film, award, actor, oscar, actress, aviator, star, festival, oscars, comedy	0.177	film, award, domain, include, star, people, director, browser, actor, release	0.196
1	"business": 0.002, "entertainment": 0.224, "sport": 0.760, "politics": 0.005, "tech": 0.0059	cup, match, coach, game, injury, rugby, win, chelsea, champion, play	0.140	play, win, game, time, player, england, team, match, club, set	0.161
2	"business": 0.022, "entertainment": 0.192, ,"politics": 0.007, "tech": 0.777	phone, broadband, mobile, technology, dvd, handset, camera, speed, video, service	0.188	phone, people, service, technology, dvd, mobile, broadband, video, gadget, offer	0.187
3	"business": 0.498, "entertainment": 0.060, "sport": 0.004, "politics": 0.415, "tech": 0.021	government, labour, election,minister, blair, economy, party,tax, tory, chancellor	0.175	government, company, people, minister, labour, country, party, plan, election, law	0.198
4	"business": 0.045, "entertainment": 0.053, "tech": 0.900	game, device, console, gaming, music, nintendo, gamer, gadget, xbox, sony	0.252	game, music, people, player, device,consumer, technology,play, market,sony	0.276
5	"business": 0.0144, "entertainment": 0.007, "tech": 0.978	user, software, virus, mail, program, search, blog, security, spam, google	0.194	people, software, user, firm, search, file, network, mail, security, program	0.200
AVG +/-2*STD			0.1881+/- 0.067		0.2036 +/- 0.070

investigate how the level of specificity/generality of the words in the explanations change according to the heterogeneity of the clusters (i.e. multiple topics — more noise). First, we compute the Inverse Document Frequency (IDF) for each word in the dataset which is correlated with the "specificity" of the words. In particular, the IDF reflects a "specificity" measure that is mostly defined in the opposite terms: words with higher IDF are not common words. We are interested to have a specificity measure that is closer to the definition that takes into account how a word reflects a piece of particular information about a domain, and to estimate this "specificity" or "generality" measure, we quantize the IDF (Q-IDF) in three main bins: "Low", "Medium" and "High". In particular, we assume that the level of the "specificity" is computed as the fraction (%) of words that belongs to the "High" and "Medium" bins of the Q-IDF, while the words in the "Low" bin are more generic because they are spreads over all the collections of documents, therefore they are more domain agnostic. We plot in Fig. 5 how the % of the words in the "Low" bin (i.e. Y-axis) changes respect to the level of the heterogeneity of the clusters that are correlated to their level of noise (i.e. X-axis). In particular, we assign to each cluster a position in the X-axis that reflects his level of noise based on the entropy values of the target labels used to annotate the clusters. Clusters with uniform distributions are more noise so they are ranked first. The plot shows that the differences among the % of generic words measured using the "Low" bin of the Q-IDF between the CDTF and LDA approach increases as the clusters become less noisy. This means that in the presence of coherent clusters the CDTF is selecting words that are more specific of the ones extracted using LDA. For example, for cluster 6 (most coherent cluster), we have a difference of 0.4 between the % of words in the "Low" bin of the Q-IDF for the CDTF and LDA approaches, with the CDTF having a lower proportion, therefore it has more specific words.

7.4. Experiments with IntentC dataset

In this section, we describe the results of the approach described in Section 5. The approach extracts clusters explanations using a supervised dataset. In particular, we used the IntentC dataset introduced in Section 6.1. The results are reported in Tables 21 and 22 in terms of *WMRR* and *WMRR** respectively. We note that in this case, the *semr* function used in the *WMRR* and *WMRR** is the exact match because the scope is to evaluate how the predicted target labels used for the explanations are the

Table 18
The Table reports the results of the EDES algorithm for the BBC dataset using Wikipedia as an external knowledge-base using two different score-based methods to extract the initial explanations for the BBC datasets. The first one is the CDTF, while the second id the LDA. We use the same length for the initial explanations (i.e.Top-E=20). For the EDES algorithm, we use the setting Wiki-1@10-10-5 which is explained in Section 7.3.

Cluster index	Target labels	Explanations Wiki-1@10-10-5 Top 10 Words (Input: CDTF)	WMRR* Wiki-1@10-10-5 (Input: CDTF)	Explanations Wiki-1@10-10-5 Top 10 Words (Input: LDA)	WMRR* Wiki-1@10-10-5 (Input: LDA)
0	"business": 0.024, "entertainment": 0.860, "politics": 0.006, "tech": 0.109	television, theatrical, award, art, oscar, rapper, star, festival, social, academy	0.230	french, technology, browser, astronomy, people, english, acting, television, cultural, digital	0.138
1	"business": 0.002, "entertainment": 0.224, "sport": 0.760, "politics": 0.005, "tech": 0.0059	winner, cup, equipment, german, hungarian, match, coach, game, trauma, pain	0.112	concept, metaphysics, theory, country, britain, play, win, game, science, idea	0.129
2	"business": 0.022, "entertainment": 0.192, ,"politics": 0.007, "tech": 0.777	technology, mobile, video, television, digital, telecommunication, audio, telephony, equipment, consumer	0.282	technology, mobile, video, television, digital, telecommunication, audio, telephony, radio, concept	0.274
3	"business": 0.498, "entertainment": 0.060, "sport": 0.004, "politics": 0.415, "tech": 0.021	government, party, political, labour, politics, election, minister, bank, banking, legal	0.217	party, government, political, company, legal, people, minister, labour, country, management	0.208
4	"business": 0.045, "entertainment": 0.053, "tech": 0.900	game, video, television, brand, software, console, culture, company, technology, device	0.274	technology, television, consumer, video, company, game, music, performing, sound, people	0.355
5	"business": 0.0144, "entertainment": 0.007, "tech": 0.978	google, security, software, computing, internet, user, science, virology, virus, blog	0.237	technology, people, science, software, user, business, network, enforcement, security, cultural	0.373
AVG +/-2*STD			0.2259 +/- 0.112		0.2467 +/- 0.192

real target labels used to annotate the documents in the clusters. We design these experiments in order to evaluate how the results change respect to three factors: cluster strategy, number of target labels retrieved from the search, size of the labelled data. In Table 21 we present the results related to the perfect clustering strategy, while in Table 22 we present the results respect to the K-means clustering strategy. In the K-means, we set the number of clusters to 100 (K) using the Elbow method, and we also found that the ARI for the obtained clusters was greater than 0.4. In both tables, we present different results varying the other two factors: TopK-Ann and the Test Size. The TopK-Ann variable indicates the number of results retrieved by the ANN search module. In our experiments, we set TopK-Ann equal to 2 and 5, which means we select only the first two and five results from the search respectively. The Test Size refers to the size of the un-labelled data, we vary it from 10% (large quantity of labelled data) to 70% (a small quantity of labelled data). We use as external knowledge-base for the EDES the Ontology defined for the IntentC dataset, which is mainly a taxonomy used to organize the target labels in terms of domain concepts, an example of this taxonomy is reported in Table 20. We used this taxonomy to enforce the semantic coherence among the candidates' target labels. In particular, in the graph used by the EDES algorithm, we create an association between tags if they share the same

Table 19
The Table reports the results of the EDES algorithm in terms of *WMRR* applied to the BBC dataset clustered using K-Means. We report the results for using both CDTF and CC as score mechanism to extract the initial labels. WordNet is the external knowledge-base. We also include the *WMRR* results using the approach described in [11].

K-Means	
Method	Average WMRR@10
Wordnet@10-10-5 with CDTF	0.191 +/- 0.222
Wordnet@10-10-5 with CC	0.182 +/- 0.208
Tseng [11]	0.1118 +/- 0.101

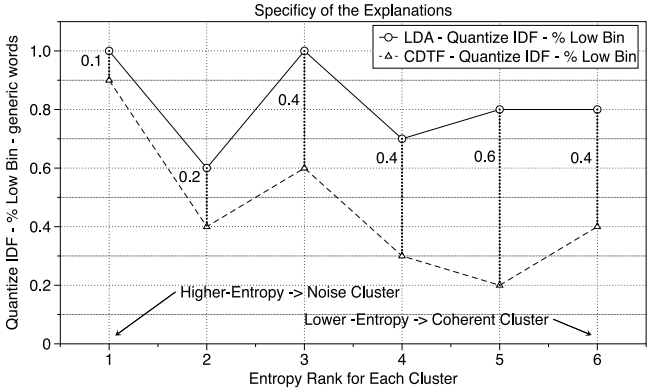

Fig. 5. The Figure shows how the level of "generality" of the words in the explanations related to clusters of the BBC dataset changes according to the heterogeneity/noise level in the clusters. The level of the "generality" is computed as the fraction (%) of words that in the "Low" bin of the Quantize IDF (Q-IDF), while the clusters are ranked in the *X*-axis based on the entropy. A detail description of these measures is described in Section 7.3.

domain concepts. We note that in this context the nodes referred to as tags in the graph used by the EDES algorithm are the target labels derived from the labelled documents. We also introduce a baseline for comparison. In the baseline, the predicted target label is the most retrieved target labels based on the words in the initial explanations, which is an approach similar to what [3] proposed in their work. In Table 21 we report the results in terms of *WMRR*@1, which means we consider only the first predicted target labels. We observe that our approach has superior accuracy compared to the baseline. The perfection in the clusters results compensates also the effect of having a small number of labelled data. In this setting, we did not observe any gain in considering larger results from the retrieval process. In Table 21, we report the results in terms of *WMRR**@1, but we use the Ontology@10-10-5 setting for the parameters of the EDES algorithm, which means that we constraints the solver to find exactly 10 candidates for the target labels for each cluster, and at most half of them could be shared across the cluster. The full explanation of this notation in terms of the EDES algorithm parameters is described in Section 7. Due to a large number of clusters (100), we force the solver to explore at least 10 candidates for each cluster, but then for the evaluation, we select only the first one in order to evaluate is the first predicted target label was equal to the main target label within the cluster. In this context, we used the optimization model with a relaxed constraint using the formulation presented in Eq. (9). In particular, we found that the constraint related to the shared condition (7) was causing an infeasible solution, therefore we had to relax it introducing the slack variables. We observe that also in this context the quality of our predicted target labels is also superior respect to the baseline. We also note that restricting the results of the retrieval process to only the first two results is helping in improving the *WMRR**@1 results. This suggests that regulating the results of the retrieval process has an impact on the quality of the clusters explanations. We note that the confidence intervals in these results are much smaller compared to the ones presented in the previous analysis because in this context we use the exact much in the *semr* function, therefore we do not observe the effect of the embedding similarity measure in the *WMRR*@1 and *WMRR**@1 as discussed in the previous sections. We note that if we label only the 30% of the document we have a drop of the *WMRR**@1 of only 9% compared to a situation where we label 90% of the data. This shows that by using the unsupervised knowledge extracted from data (i.e. clustering), the domain knowledge (i.e. ontology), the human annotation (i.e. labelled data), and our model we can group and explain the information stored in the data.

Table 20
Example of the Taxonomy used as Ontology for the cluster explanations process described in Section 5, and evaluated in Section 7.4 for the IntentC dataset.

Domain concepts	Target labels
Banking	Transfer Balance Pay Bill
Credit Cards	New Card International Bill Credit Limit

Table 21
The table describes the *WMRR*@1 results of the clusters explanations approach described in Section 5 in the context of the perfect clustering strategy. The settings of the experiments and the baseline are described in Section 7.4.

TopK ANN	Test size	*WMRR*@1 Ontology@1-1-1	*WMRR*@1 Baseline(1)
2	0.1	1.000+/−0.000	0.757+/−0.040
2	0.3	1.000+/−0.000	0.829+/−0.049
2	0.5	1.000+/−0.000	0.862+/−0.069
2	0.7	1.000+/−0.000	0.944+/−0.000
5	0.1	1.000+/−0.000	0.592+/−0.042
5	0.3	0.984+/−0.056	0.780+/−0.049
5	0.5	1.000+/−0.000	0.770+/−0.105
5	0.7	1.000+/−0.000	0.852+/−0.128

Table 22
The table describes the *WMRR**@1 results of the clusters explanations approach described in Section 5 in the context of the K-means clustering. The setting of the experiments and the baseline are described in Section 7.4.

TopK ANN	Test size	*WMRR**@1 Ontology@10-10-5	*WMRR**@1 Baseline (10)
2	0.1	0.722+/−0.019	0.070+/−0.067
2	0.3	0.719+/−0.035	0.057+/−0.097
2	0.5	0.719+/−0.015	0.127+/−0.054
2	0.7	0.698+/−0.023	0.072+/−0.088
5	0.1	0.646+/−0.058	0.077+/−0.096
5	0.3	0.643+/−0.034	0.071+/−0.094
5	0.5	0.629+/−0.016	0.189+/−0.207
5	0.7	0.607+/−0.018	0.065+/−0.110

8. Conclusion

Besides some interesting advances in textual clustering, the interpretation of the clustering results still relies on the manual inspection of their contents. The manual review of the cluster not only is a time-consuming activity but is also limiting the use of the clusters in decision support systems where the explainability is a requirement. In this paper, we proposed different approaches to extract cluster explanations based on scores, external knowledge-base, and labelled data. In particular, we found that two score-based methods provided specific enough explanations and they performed (with statistical significance) better than other methods such as the one based on topic analysis like Latent Dirichlet Allocation or well-adopted natural language processing methods like Inverse Document Frequency. We also proposed

an extendible framework where we expanded the score-based explanations encoding the domain knowledge as constraints in the optimization problem. We demonstrated how the proposed framework provide more conceptual and general explanations. In our experiments, we showed how this injection of an external knowledge-base improves on average the explanation results for clusters that are more diverse in terms of content. We also found in the presented user study, that users prefer the score-based explanations respect to the one based on external knowledge bases along with their related rank with 85% preferences (with Fleiss' Kappa value between 0.61–0.80) because they are more focus on finding explanation throughout the details rather than general concepts. We also showed how the two methods (score-based and external knowledge-base) can be combined to provide meaningful insights into the clusters in the context of semi-supervised textual datasets. In particular, we found that if we label only 30% of the document we have a drop of only 9% in our metric compared to a situation where we label 90% of the data. We also extended the metric adopted in previous works in order to evaluate the semantic content of the cluster explanations, and we presented an analysis of how the nature of the explanations could change in the presence of noisy clusters. In particular, we suggested to monitor the Adjusted Rand Index ensuring to be greater than 0.4 in order to have meaningful explanations. We also presented an evaluation protocol/measure where the evaluations of the cluster explanations can be done not only based on the target labels but also considering external characteristics such as user preferences. This user-study evaluation can be beneficial in applications that are directly targeting users. We also discussed the limitations of the evaluation metrics used in previous literature and we suggest solutions to overcome some of these limitations. In the end, we also note that in the current machine learning literature there is not a common and accepted definition of what should be a protocol to evaluate the quality of explanations. This could be an impactful space where the whole machine learning community could collaborate, and the proposed solutions/benchmarks could lead to interesting open questions and challenges.

CRediT authorship contribution statement

Antonio Penta: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resource, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Anandita Pal:** Data curation, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We are indebted to Rory Timlin, Paul Walsh, Norin Duggan, Md Faisal Zaman, Daphne Tsatsoulis for reviewing the work and providing important feedbacks. We would also like to express thanks to Peter Walsh for his help in planning user evaluations. We would also like to thank Accenture The Dock for supporting this research.

References

[1] C.C. Aggarwal, C. Zhai, A survey of text clustering algorithms, in: Mining Text Data, Springer US, 2012, pp. 77–128.
[2] P. Treeratpituk, J. Callan, Automatically labeling hierarchical clusters, in: Proceedings of the 2006 International Conference on Digital Government Research, dg.o '06, 2006, pp. 167–176.
[3] D. Carmel, H. Roitman, N. Zwerdling, Enhancing cluster labeling using Wikipedia, in: Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'09, 2009, pp. 139–146.
[4] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, 2004, pp. 74–81.
[5] H. Roitman, S. Hummel, M. Shmueli-Scheuer, A fusion approach to cluster labeling, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '14, 2014, pp. 883–886.
[6] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma, A survey on ensemble learning, Front. Comput. Sci. 14 (2) (2019) 241–258.
[7] Z. Li, J. Li, Y. Liao, S. Wen, J. Tang, Labeling clusters from both linguistic and statistical perspectives: A hybrid approach, Knowl.-Based Syst. 76 (2015) 219–227.
[8] L.A. Lopes, V.P. Machado, R.A. Rabêlo, R.A. Fernandes, B.V. Lima, Automatic labelling of clusters of discrete and continuous data with supervised machine learning, Knowl.-Based Syst. 106 (2016) 231–241.
[9] V.D. Orangeville, M.A. Mayers, M.E. Monga, M.S. Wang, Efficient cluster labeling for support vector clustering, IEEE Trans. Knowl. Data Eng. 25 (11) (2013) 2494–2506.
[10] F. Role, M. Nadif, Beyond cluster labeling: Semantic interpretation of clusters' contents using a graph representation, Knowl.-Based Syst. 56 (2014) 141–155.
[11] Y.-H. Tseng, Generic title labeling for clustered documents, Expert Syst. Appl. 37 (3) (2010) 2247–2254.
[12] G.A. Miller, WordNet: A lexical database for English, Commun. ACM 38 (11) (1995) 39–41.
[13] I. Ruthven, M. Lalmas, A survey on the use of relevance feedback for information access systems, Knowl. Eng. Rev. 18 (2) (2003) 95–145.
[14] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Found. Trends Inf. Retr. 3 (4) (2009) 333–389.
[15] S.E. Robertson, The probability ranking principle in IR, J. Doc. 33 (4) (1977) 294–304.
[16] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.
[17] S.E. Robertson, K.S. Jones, Relevance weighting of search terms, J. Am. Soc. Inf. Sci. 27 (3) (1976) 129–146.
[18] D. Carmel, E. Yom-Tov, A. Darlow, D. Pelleg, What makes a query difficult? in: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2006, pp. 390–397.
[19] E. Momeni, C. Cardie, N. Diakopoulos, A survey on assessment and ranking methodologies for user-generated content on the web, ACM Comput. Surv. 48 (3) (2015).
[20] P. Xiao, H. Toivonen, O. Gross, A. Cardoso, J.a. Correia, P. Machado, P. Martins, H.G. Oliveira, R. Sharma, A.M. Pinto, A. Díaz, V. Francisco, P. Gervás, R. Hervás, C. León, J. Forth, M. Purver, G.A. Wiggins, D. Miljković, V. Podpečan, S. Pollak, J. Kralj, M. Žnidaršič, M. Bohanec, N. Lavrač, T. Urbančič, F.V.D. Velde, S. Battersby, Conceptual representations for computational concept creation, ACM Comput. Surv. 52 (1) (2019).
[21] Z.S. Harris, Distributional structure, WORD 10 (2–3) (1954) 146–162.
[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, 2013, pp. 3111–3119.
[23] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[24] J.W. Chinneck, Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods, Springer, 2007.
[25] M. Aumüller, E. Bernhardsson, A. Faithfull, ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms, 2018, arXiv:1807.05614.
[26] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques, in: KDD Workshop on Text Mining, 2000.
[27] Y. Zhao, G. Karypis, Empirical and theoretical comparisons of selected criterion functions for document clustering, Mach. Learn. 55 (3) (2004) 311–331.
[28] S. Zhong, Semi-supervised model-based document clustering: A comparative study, Mach. Learn. 65 (1) (2006) 3–29.
[29] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.

[30] S. Larson, A. Mahendran, J.J. Peper, C. Clarke, A. Lee, P. Hill, J.K. Kummerfeld, K. Leach, M.A. Laurenzano, L. Tang, J. Mars, An evaluation dataset for intent classification and out-of-scope prediction, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.
[31] G. Miller, E. Newman, E. Friedman, Length-frequency statistics for written English, Inf. Control 1 (4) (1958) 370–389.
[32] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, HuggingFace's transformers: State-of-the-art natural language processing, 2019, ArXiv:abs/1910.03771.
[33] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31st International Conference on Machine Learning, 2014, pp. 1188–1196.
[34] S. Arora, Y. Liang, T. Ma, A simple but tough-to-beat baseline for sentence embeddings, in: International Conference on Learning Representations, 2017.
[35] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. 34 (1) (2002) 1–47.
[36] A.C. Benabdellah, A. Benghabrit, I. Bouhaddou, A survey of clustering algorithms for an industrial context, Procedia Comput. Sci. 148 (2019) 291–302.
[37] R.L. Thorndike, Who belongs in the family? Psychometrika 18 (4) (1953) 267–276.
[38] L. Hubert, P. Arabie, Comparing partitions, J. Classification 2 (1) (1985) 193–218.
[39] F. Morstatter, H. Liu, In search of coherence and consensus: Measuring the interpretability of statistical topics, J. Mach. Learn. Res. 18 (1) (2017) 6177–6208.