Mohsen Pourvali* and Salvatore Orlando

Enriching Documents by Linking Salient Entities and Lexical-Semantic Expansion

https://doi.org/10.1515/jisys-2018-0098
Received April 13, 2017; previously published online December 4, 2018.

Abstract: This paper explores a multi-strategy technique that aims at enriching text documents for improving clustering quality. We use a combination of entity linking and document summarization in order to determine the identity of the most *salient entities* mentioned in texts. To effectively enrich documents without introducing noise, we limit ourselves to the text fragments mentioning the salient entities, in turn, belonging to a *knowledge base* like Wikipedia, while the actual enrichment of text fragments is carried out using WordNet. To feed clustering algorithms, we investigate different document representations obtained using several combinations of document enrichment and feature extraction. This allows us to exploit ensemble clustering, by combining multiple clustering results obtained using different document representations. Our experiments indicate that our novel enriching strategies, combined with ensemble clustering, can improve the quality of classical text clustering when applied to text corpora like The British Broadcasting Corporation (BBC) NEWS.

Keywords: Document clustering, document enriching.

2010 Mathematics Subject Classification: 68U15, 68P20.

1 Introduction

Clustering algorithms are a common method to organize huge corpora of textual digital documents. In traditional text clustering, the vector-based representations of texts are purely based on terms occurring in documents. Other information, in particular latent ones, should be included in the document representation to improve the quality of document similarity metrics. In this paper, we investigate a combination of two techniques to discover such latent information. First, we select some important words in a text document, by identifying the *text fragments* mentioning the *most salient entities* linked to a *knowledge base* (articles of Wikipedia). Second, we enrich such subset of important text fragments using common semantic concepts based on a lexical-semantic database (WordNet).

To identify entities, we employ an entity linking (EL) [4, 6, 8, 13, 14] technique, aimed at identifying entities from their *mentions* or *spot* (i.e. small fragments of text referring to any entity in a knowledge base) occurring in a large corpus. More precisely, we use Wikipedia as the referring knowledge base of entities and associated mentions. The method exploited returns, for each mention selected, the *entity*, namely, a Wikipedia page with a unique URL, its title, and a set of semantic categories (or types) of the page as defined in Wikipedia. We combine such EL technique with text summarization to finally arrive at identifying the most salient entities/topics discussed in a document. Indeed, we adopt a graph-based ranking summarization algorithm [17] to create a summary and finally identify the most salient entities.

We finally use the original text fragments mentioning such salient entities to enrich the final document vector representation of vectors. To this end, WordNet is used to enrich the text fragments identified, so far, with ontology-based latent information. Indeed, we take advantage of predecessor/successor concepts within

***Corresponding author: Mohsen Pourvali,** Università Ca' Foscari Venezia, Venice 30172, Italy, e-mail: mohsen.pourvali@unive.it
Salvatore Orlando: Università Ca' Foscari Venezia, Venice, Italy

four semantical relations in WordNet to expand the original text. Finally, as semantic enrichment allows us to produce different vector representations of documents, thus, entailing different similarity measures between them, we exploit a clustering ensemble approach applied to BBC NEWS articles to validate our technique and assess the improvement in the clustering quality obtained.

The rest of the paper is organized as follows. The related works are presented in Section 2. Section 3 discusses our unsupervised approach, based on salient entities, for enriching documents and clustering them. In Sections 4 and 5, we discuss the experimental results, and Section 6 draws some conclusions.

2 Related Work

Given a plain text, EL aims at identifying the small fragments of text (also called spots or mentions) possibly referring to any named entity that is listed in a given knowledge base like Wikipedia. The ambiguity of natural language makes it a nontrivial task. Among the most influential work in the field, WikiMiner [14] exploits a novel relatedness measure [14] within a machine learning framework for disambiguating. TAGME [8] focuses on efficiency and effectiveness for processing short texts (e.g. micro-blogs), but it was proved to be effective also for longer texts. Ceccarelli et al. introduced a new machine-learned entity relatedness function that improves all the previous methods [4].

There are several services/tools for EL like AlchemyAPI (http://www.alchemyapi.com/) and The Wiki Machine (http://thewikimachine.fbk.eu/), which provide API calls to automatically annotate text with respect to the external knowledge (like Wikipedia pages and DBpedia). A notable API, which is the one used in this work, is Dandelion Entity Extraction (https://dandelion.eu/). Dandelion is based on and enhances TAGME [8].

In the text clustering domain, latent information is exploited in different ways during the clustering process. Some contributions consider the latent information of documents by only considering the attributes of the named entities [3, 11, 15]. In Ref. [3], the authors propose an entity-keyword multi-vector space model, which represents a document by a vector on keywords (i.e. the words of original documents used in the traditional VSM model) and four vectors on named entity features (i.e. entity names, types, name-type pairs, and identifiers). The main idea in this work is to generate a trade-off between named entity features and traditional vector space model depending on the importance of entities and keywords among the collection.

Besides, there are contributions in which the authors propose to exploit an ontology of common concepts like WordNet [18, 22]. The common idea behind the different approaches is to try to expand the latent information, which is hidden among the terms of a document, in order to improve somewhat the quality of text clustering. The obtained results by these approaches indicate such improvement. Intuitively, if we want to cluster a collection of documents based on their *contents*, the aim of clustering may be defined to group those documents in which their *main topics*, being discussed in each one, are in common. However, each document contains several topics, for each of which there are relevant terms in documents [2]. Therefore, not all the terms appearing in a document have the same relevance and utility in understanding the main topic being discussed. Expanding latent information, by exploiting the terms that are relevant to the main topic of the document, is more efficient in finding similar documents rather than expanding all terms of included topics, which may contrariwise cause increasing noises coming from irrelevant information.

3 Document Enriching and Ensemble Clustering

In this section, we present our unsupervised approach, called Salient Entities for Enriching Documents (SEED), to enrich documents before clustering. The aim of SEED is to identify significant fragments of text concerning the main topics discussed in each document, overcoming the issues of using term/document frequency to identify such fragments, to finally enrich the vectorial document representation.

The brief description of our approach is as follows. First (1), we extract all the entities implicitly linked in a document. In order to extract such entities from text, we use Dandelion Entity Extraction (https://dandelion.eu/). Then (2), we exploit the NG-Rank algorithm [17] for summarizing text. The entities appearing in both the summary and the original text are selected as the most salient entities. We (3) utilize the semantic relations in the WordNet ontology to expand the knowledge associated with such salient entities, by carefully disambiguating the sense of specific terms, namely, the terms implicitly mentioning the salient entities in the original text (Section 3.3). Finally, as diverse representations of documents can be generated by combining in different ways the expanded sets of features, we (4) exploit ensemble clustering to combine multiple clustering results, in turn, obtained using the diverse document representations.

3.1 Document Enrichment

In the following, we discuss the various steps for document enrichments.

3.1.1 Entity Extraction

Wikipedia has emerged as an important repository of semi-structured, collective knowledge about notable entities [10], already linked to many existing formal ontologies through efforts like DBpedia and Semantic MediaWiki. We use the Dandelion Entity Extraction API to obtain, given an input text, the Wikipedia entities (titles and URIs) possibly cited within the text, along with their mention (or spot) and some other relevant information. The mention of an entity indicates the fragment of text that is identified as a reference to the detected entity, like the anchor text of a hyperlink. More formally, let $D = \{D_1, D_2, \dots, D_m\}$ be a collection of documents, and let $Ent(D_i) = \{(e_1, m_1), (e_2, m_2), \dots, (e_n, m_n)\}$ be the set of all pairs of *entities* and *associated mentions* (e_i, m_i) occurring in D_i. While each e_i is identified by a URI and/or a unique title, a mention m_i is indeed an *n-gram*, i.e. a contiguous sequence of n terms referring to e_i.

3.1.2 Salient Entity Selection

To select the most salient entities, we exploit the NG-rank summarization algorithm [17] to create a summary S_i for each document $D_i \in D$. In principle, only the entities mentioned in both S_i and D_i are selected for further semantic expansions. However, as each S_i is a keyword-based summary, an *n-gramm* that is recognized as a mention to an entity in the original document D_i can appear only partially in S_i, or the terms of m can be scattered over the text of S_i. If all the terms of the n-gram m are completely discarded during the summarization and, thus, do not appear in S_i, the associated entity is not considered salient, but what if the terms of m appear partially or are spread over the summary?

To illustrate our method, we consider each summary S_i and each spot m as a multiset (bag) of words. So, the salient entities $\widehat{Ent}(D_i)$, where $\widehat{Ent}(D_i) \subset Ent(D_i)$, are identified as follows:

$$\widehat{Ent}(D_i) = \{(e, m) \in Ent(D_i) \mid m \cap S_i \neq \emptyset\}$$

where for each $(e, m) \in Ent(D_i)$, we have by definition that $\forall x \in m, x \in D_i$. We argue that this method allows us to enrich a document by only expanding important portions of the document, without introducing noise, which could come from a method that semantically enriches terms of irrelevant phrases, too, namely, salient and not salient ones. Finally, the set of mentions to salient entities occurring in a document D_i is denoted by $\widehat{M}(D_i)$ and defined as follows:

$$\widehat{M}(D_i) = \bigcup_{(e,m) \in \widehat{Ent}(D_i)} m$$

For example, consider a document D from which we extract $\widehat{Ent}(D)$, where $\widehat{Ent}(D) = \{$(Profit (accounting), profit), (Market (economics), market), (United States dollar, dollar), (Telecommunication, telecoms)$\}$. The former element of each pair, e.g. “*Profit (accounting)*”, is the *title* of Wikipedia articles, while the latter one, e.g. “*profit*”, is the corresponding n-gram mention. In this example, all the mentions are simple 1-grams. Finally, we have $\widehat{M}(D) = \{\text{profit, market, dollar, telecoms}\}$.

3.1.3 Expanding Salient Entities

This step regards the final enrichment, which concerns the topical terms identified by $\widehat{M}(D)$. The enrichment leverages the lexical-semantic database *WordNet*.

In WordNet, words that denote the same concept (synonyms) and are interchangeable in many contexts are grouped into unordered sets (*synsets*). Therefore, a word related to n synsets in WordNet has n possible senses. These senses may cover multiple parts of speech; for example, if a word has eight distinct synsets, it might have five noun senses, two verb senses, and an adjective sense. More specifically, in this work, we only use noun senses of words. Additionally, for each synset in WordNet, there is a brief definition (*gloss*), in which the use of the synset members is illustrated by one or more short sentences.

Before enriching the terms in $\widehat{M}(D)$, we need to identify their senses (meanings). This corresponds to selecting one of the possible WordNet synsets, which is chosen on the basis of the context in which each term occurs. To disambiguate the senses of terms in $\widehat{M}(D)$, we exploit the word sense disambiguation (WSD) algorithm illustrated in Section 3.3, which assigns a sense to each word in $\widehat{M}(D)$. Note that our WSD technique uses a very small but unnoisy context for disambiguating, as $\widehat{M}(D)$ only includes the words related to the most salient entities (topics) discussed in the document.

Generally, WordNet includes several semantic relations, in which the most important relation among synsets is the super-subordinate relation (also called hyperonymy, hyponymy, or ISA relation). It links more general synsets to increasingly specific ones that generates semantic hierarchies in either direction, from general to specific or from specific to general concepts. Another important relation is meronymy, also called the part-whole relation, which indicates inheritance between concepts [7].

We exploit four available forms (http://wordnet.princeton.edu) of these relations to disambiguate the sense of words and to finally enrich documents:

- *hypernym* (kind-of or is-a): Y is a hypernym of X if every X is a (kind of) Y (e.g. *motor vehicle* is a hypernym of *car*).
- *member meronym* (member of): Y is a member meronym of X if Y is a member of X (e.g. *professor* is a member meronym of *faculty*).
- *part meronym* (part of): Y is a part meronym of X if Y is a part of X (e.g. *camshaft* is part meronym of *engine*).
- *substance meronym* (contains, used in): Y is a substance of X if Y contains (used in) X (e.g. *water* is a substance meronym of *oxygen*).

Each of these relations is used to extract a *rooted directed acyclic graph* (DAG) from WordNet, where nodes are synsets, and directed edges model one of the above semantic relations. We use such DAGs in several steps, namely, to disambiguate sense of words, and finally, to enrich the document vectorial representation of documents to be clustered.

In the following, we finally discuss how we identify the WordNet elements, in turn, used to prepare the enriched vector representation of documents. For each word in $\widehat{M}(D_i)$, we exploit the synsets (senses of words) identified by the WSD algorithm, along with the further synsets in WordNet that are related to the first ones through semantic relations of types *hypernym*, *part meronym*, *member meronym*, and *substance meronym*, respectively. Let $Syns(D_i)$ denote the senses (synsets) of the words in $\widehat{M}(D_i)$, as identified by WSD. For a given $s_i \in Syns(D_i)$, and for each type of semantic relation, e.g. for *hypernym*, we can distinguish between synsets that are *direct predecessor* and *direct successor* in the hypernym DAG extracted from WordNet. If an edge (s_i, s_{succ}) exists in the DAG, s_{succ} is a direct successor, whereas an edge (s_{pred}, s_i) identifies s_{pred} as a direct predecessor. At the end of this process, by considering all the words in $\widehat{M}(D_i)$ and the four types of relations, we can associate with each D_i three sets of synsets: $Syns(D_i)$, $PredSyns(D_i)$, and $SuccSyns(D_i)$. While

$Syns(D_i)$ includes the sense synsets of the words in $\widehat{M}(D_i)$, the other two sets contain, respectively, the direct predecessor and direct successor synsets according to all the four types of WordNet relations.

3.1.4 Feature Extraction

Finally, for each document $D_i \in D$, we can pick from a large set of sources to extract the features of the vector representing D_i. For example, we can enrich the vector representation by only using the words appearing in the senses (synsets) of the terms in $\widehat{M}(D_i)$ ($Syns(D_i)$), or we can also exploit their predecessors/successors in the WordNet graph. In particular, to create this enriched vector representation, we exploit the following multisets/bags of word (we stem words after removal of stop words):

Or_i — $OrigDoc(D_i)$, denoted in short by Or_i, is the multiset of words associated with the original document D_i;

Sum_i — $SummDoc(D_i)$, denoted in short by Sum_i, is the multiset of words occurring in the summary extracted from D_i by NG-rank [17];

Na_i — $NamesEN(D_i)$, denoted in short by Na_i, is the multiset of words appearing in the titles of the salient entities in $\widehat{Ent}(D)$, formally defined by

$$Na_i = \bigcup_{(e,m)\in \widehat{Ent}(D_i)} e$$

Me_i — $MentionsEN(D_i)$ is exactly the multiset of words containing the mentions of the salient entities in $\widehat{Ent}(D)$, formally defined as follows:

$$Me_i = \widehat{M}(D_i) = \bigcup_{(e,m)\in \widehat{Ent}(D_i)} m$$

Sy_i — $Syns(D_i)$, denoted in short by Sy_i, is the multiset of words occurring in the identified senses (synsets) of the words in Me_i, i.e. in the mention, possibly refers to the most salient entities in D_i;

Pre_i — $PredSyns(D_i)$, denoted in short by Pre_i, is the multiset of words occurring in all the synsets that directly precede the ones in Sy_i, according to any of the four types of WordNet relations *hypernym*, *part meronym*, *member meronym*, and *substance meronym*;

Suc_i — $SuccSyns(D_i)$, denoted in short by Suc_i, is the multiset of words including all the synsets that are the direct successors of the ones in Sy_i, according to any of the four types of WordNet semantic relations above.

3.2 Feature Selection and Ensemble Clustering

We utilize a *clustering ensemble* method, which combines different clustering results to finally partition documents. Although we exploit the same clustering algorithm to partition the input document corpus, as we can adopt different enrichment and associated vector representations of documents, the final clustering results may differ. The rationale of using ensemble clustering is that each single enrichment strategy may generally work for the whole corpus, but may introduce noise in the representations of a few documents that are eventually clustered badly. Ensemble clustering permits us to exploit many possible document enrichments and finally remove possible noisy results through a consensus method.

A cluster ensemble method consists of two steps: *Generation*, which creates a set of possible partitions of the input objects (in our case, a document corpus), and *Consensus*, which computes a new partition by integrating all the partitions obtained in the generation step [21].

In our experiments for the generation step, we adopt a hybrid function $\mathcal{F}$; indeed, many different instances $\{\mathcal{F}^h\}_{h=1,\dots,n}$ of this function entail different *feature selection* methods, thus, generating different subsets of features and vectorial representation of documents. Therefore, $\mathcal{F}$ models different possible enrichment strategies.

Hence, we consider the above multisets of words for each document D_i, denoted by $SES(D_i) = \{Or_i, Sum_i, Na_i, Me_i, Sy_i, Pre_i, Suc_i\}$ and combine them using different instances of function $\mathcal{F}$.

Let $\mathbb{C} = \{\mathcal{C}^1, \mathcal{C}^2, \dots, \mathcal{C}^n\}$ be the different clusterings of the document corpus $\mathcal{D}$, where each clustering $\mathcal{C}^h$ is obtained by first applying the instance $\mathcal{F}^h$ of the feature selection function over the corpus's documents and, then, by running over them a given clustering algorithm. In our case, we exploit *k-means*, a well-known algorithm that takes the input document corpus and produces k disjoint clusters. Thus, each $\mathcal{C}^i$ is a partition required for the ensemble method. Formally, the enriched bag-of-words representation of D_i, obtained by $\mathcal{F}^h$, is denoted by D_i^h, while the instance $\mathcal{F}^h$ of the combining function is due to the different settings of six integer parameters, namely, $\alpha^h, \beta^h, \gamma^h, \varepsilon^h, \delta^h$, and η^h:

$$\begin{aligned} D_i^h &= \mathcal{F}^h(D_i|\alpha^h, \beta^h, \gamma^h, \varepsilon^h, \delta^h, \eta^h) \\ &= \{Or_i\} \cup (\alpha^h \cdot Sum_i) \cup (\beta^h \cdot Na_i) \cup (\gamma^h \cdot Me_i) \cup (\varepsilon^h \cdot Sy_i) \cup (\delta^h \cdot Pre_i) \cup (\eta^h \cdot Suc_i) \end{aligned}$$

where $\alpha^h, \beta^h, \gamma^h, \varepsilon^h, \delta^h, \eta^h \in \{0, 1, 2, \dots, v\}$ indicate the number of times we replicate the elements of $SES(D_i)$ to generate a new bag-of-words document representation D_i^h. More formally, $\alpha^h \cdot Sum_i = \biguplus_{j=1}^{\alpha^h} Sum_i$, where the operator $\uplus$ denotes the *multiset union*, and thus, D_i^h will contain α^h replicas of the document summary Sum_i. In case a parameter equals zero, for example, $\alpha^h = 0$, then $\alpha^h \cdot Sum_i$ is equal to $\emptyset$. In our experiments, we varied these parameters and used a different maximum value v for every parameter.

It is worth remarking that by varying the parameter setting to generate a different $\mathcal{F}^h$, we may change the vocabulary used to identify the dimensions of document vectors, but we may also modify the term frequency and, thus, the *tf.idf* weights used in the vectors. As a consequence, if we enrich and represent a corpus D according to different $\mathcal{F}^h$, we produce different partitions of the corpus even if we run the same clustering algorithms.

For the consensus step, we apply the *objects co-occurrence* approach, which is based on the computation of how many times two objects are assigned to the same cluster by the various clustering instances of the ensemble. Like in the *cluster-based similarity partitioning algorithm* (CSPA) [20], we, thus, build $m \times m$ similarity matrix (the co-association matrix), which can be viewed as the adjacency matrix of a weighted graph, where the nodes are the elements of the document corpus D, and each edge between two objects (documents) is weighted with the number of times the objects appear in the same cluster, for each instance of the clustering ensemble. Then, the graph partitioning algorithm METIS is used for generating the final consensus partition.

3.3 Words Sense Disambiguation

In this section, we discuss our unsupervised WSD method that uses WordNet as a knowledge base. Given a *target word* to disambiguate, we utilize the four above-mentioned semantic relations of WordNet to identify its best sense (meaning) among all the possible senses in WordNet. The disambiguation WSD strategy takes advantage of the *word context*, i.e. a portion of document that surrounds each word. The size of word contexts may be different, e.g. Unigram, Bigrams, Trigrams, Sentence, Paragraph, or different size of a window [16]. Determining such word context for a target word is crucially important because wrong relations between the target word and other words in the context may affect the best sense selection.

The novel idea of our approach is to limit the word sense disambiguation task to the terms included in $\widehat{M}(D_i)$, while $\widehat{M}(D_i)$ is also used as the context used by our WSD algorithm. As our word context $\widehat{M}(D_i)$ is extracted by a summary including terms closely related to the *main topic* of documents, semantically related to each other, this should hopefully favor a fair selection of the appropriate senses for each target word.

Our approach proceeds as follows: given $\widehat{M}(D_i) = \{w_1, w_2, \dots, w_{n_i}\}$ as the word context, where w_i is a word (noun) included in a mention of a salient entity, we create n_i semantic trees, one for each $w_k \in \widehat{M}(D_i)$, as illustrated in Figure 1A. The method works as follows:

(i) First, for each $w_k \in \widehat{M}(D_i)$, associated with the root of a tree, we identify $S(w_k) = \{s_1, s_2, \dots, s_{m_k}\}$, which includes all the possible senses (*synsets*) of w_k. From the root w_k, the tree is thus grown by adding $m_k = |S(w_k)|$ children, where each child corresponds to a distinct synset $s_j \in S(w_k)$;

(ii) For each sense s_j in $S(w_i)$, currently a leaf of the tree, we denote by $G(s_j) = \{wg_1, wg_2, \dots, wg_{h_j}\}$ all the terms within the associated WordNet's *gloss*. From each leaf s_j, we further grow the tree, by adding $h_j = |G(s_j)|$ children, where each child corresponds to a distinct word $wg_t \in G(s_j)$;

(iii) For each $wg_t \in G(s_j)$, and for all $s_j \in S(wg_t)$, we repeat step 1, and add another level to the tree, where the new leaves are the possible synsets associated with word wg_t.

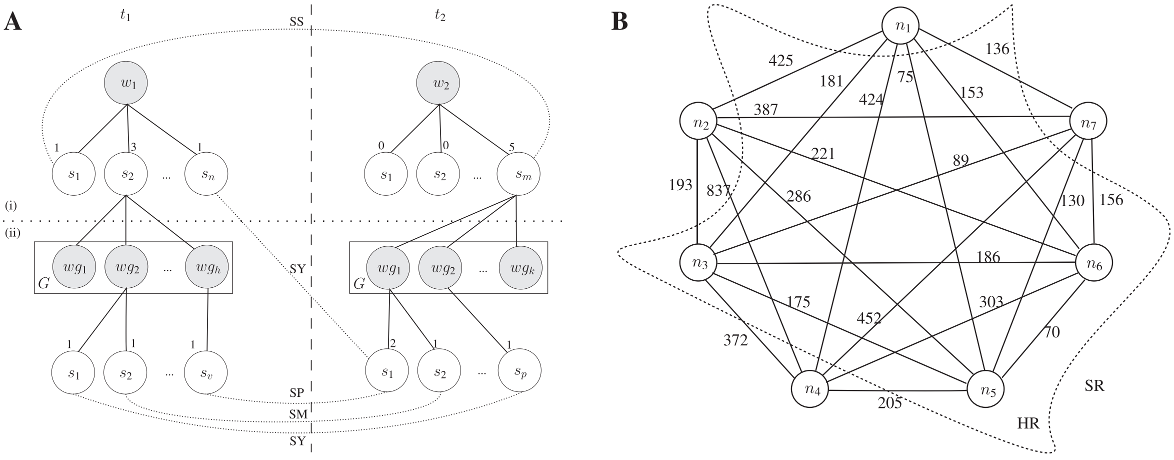


Figure 1: The conceptual shapes to visualize the relationships and information in (A) WordNet-Based Trees t_1 and t_2, Created for Words w_1 and w_2. (B) Graph G Build on the WSD Output for $\widehat{M}(D_i)$, with the Cuts by METIS Algorithm.

Finally, our technique creates a *forest of n_i indirected trees* $T(D_i) = \{t_1, t_2, \ldots, t_{n_i}\}$, each of three levels and each associated with a distinct word of context $\widehat{M}(D_i)$.

As explained above, we exploit four semantic relations of the WordNet ontology, namely, hypernym (*SY*), member meronym (*SM*), part meronym (*SP*), and substance meronym (*SS*). For each of these four relations, we can extract a directed graph (rooted DAG), where the nodes are synsets, and the edges are the semantic relations.

Returning to consider the forest $T(D_i)$, for each pair of synsets s_i and s_j occurring in two distinct trees of $T(D_i)$, if a directed edge between them exists in one of the four semantic DAGs, we add an *undirected inter-tree edge* between s_i and s_j, labeled as either *SY*, *SP*, *SM*, or *SS*. In our example in Figure 1A, these new undirected inter-tree edges are represented as *dotted (labeled) links* between pairs of synsets. These edges indicate that the two connected synsets are semantically related.

To extract the best sense for each word in $\widehat{M}(D_i)$, we proceed through a voting process, using these semantic relations between pairs of synsets as a sort of "mutual vote" between them. The final goal is to rank, for each $w_k \in \widehat{M}(D_i)$, the synsets $S(w_k) = \{s_1, s_2, \ldots, s_{m_k}\}$ occurring at depth 1 of each tree, for finally selecting the synset that obtains the highest vote. The voting mechanism works as follows:

(i) First, we assign an initial vote to each synset s_i occurring in the forest of trees. This initial vote is simply the *degree* of the corresponding node, by only considering the dotted edges, labeled by either *SY*, *SP*, *SM*, or *SS*. The intuition is that a synset is important if it is related to others synsets occurring in other trees, in turn modeling the context $\widehat{M}(D_i)$.

(ii) Second, we assign the final vote to each synset in $s_i \in S(w_k)$, by summing up the votes of all the synsets that belong to the subtree rooted at s_i, indeed the leaves at depth 3 of this subtree.

If the voting strategy discussed, so far, is not able to select the best sense for $w_k \in \widehat{M}(D_i)$, for example, because all the votes assigned to the synsets in $S(w_k)$ are zero, then, we select the sense that was tagged for the highest number of times in the semantic concordances (Actually, it corresponds to the most common sense of w_k.).

Looking at the example in Figure 1A, the vote of s_m in tree t_2 should be equal to 1 only considering the *inter-tree relations*, as the *degree* of s_m is equal to 1 if we only consider its dotted edges. The final vote of s_m becomes 5, by also considering the contribution of the synsets in the subtree rooted at s_m – namely, s_1, s_2, and s_p – which contributes to the final vote by the quantity $2 + 1 + 1 = 4$.

The reason for introducing a new WSD technique instead of using other methods, already proposed in the literature, is the different contexts used to disambiguate the senses of the target words, i.e. the words for which we have to identify the best senses. Instead of using the original sentence(s) including these target words, we limit ourselves to the words that are the most relevant with respect to the salient entities mentioned in the text. This context is less noisy, even if we have to pay an initial expensive step to first identify the salient entities and their mentions. Another reason is that our WSD method works on a WordNet semantic network, and the same elements of the WordNet networks, used to disambiguate and identify the best senses of words, are also used to enrich the vector representation of the original documents.

In Section 5.2, we compare our WSD approach with a lesk-based algorithm as a baseline, i.e. the adapted lesk algorithm [1], in order to evaluate the accuracy of our approach dealing with the contexts commonly used for evaluation.

3.3.1 Removal of Noisy Terms

Using only the mentions of the salient entities for expanding is an efficient way to reduce noises originating from irrelevant terms. However, noises may still be transferred from some terms in $\widehat{M}(D_i)$. To reduce this possible noise, we take advantage of the output of the WSD algorithm, by exploiting the relations between pairs of synsets (mutual votes) occurring between the semantic trees used by the WSD. From $\widehat{M}(D_i)$, we first build a weighted graph G_i, whose nodes correspond to the words in $\widehat{M}(D_i)$. Two nodes n_1 and n_2 of G_i, in turn, associated with words w_1 and w_2 in $\widehat{M}(D_i)$, are connected by an edge if at least a mutual vote there exists between two senses occurring in the WSD semantic trees of w_1 and w_2 (Figure 1A). The weight of the edge between n_1 and n_2 is computed by summing all these mutual votes. Figure 1B shows graph G_i for $\widehat{M}(D_i)$, where D_i is a document in the BBC corpus belonging to class *Business*. In Figure 1B, the entities and their spots/mentions (duplicated spots are removed) within $\widehat{Ent}(D_i)$ are, respectively, as follows: *Entities* = {*Property, Realestateeconomics, Realestateappraisal, EnglandandWales, Market(economics), Financialtransaction, Sales, Fiscalyea*}, and *Spots* = {*properti, hous, market, land, transact, sale, quarter*}.

It is worth recalling that the words in $\widehat{M}(D_i)$ are those filtered by a summarization technique, indeed, the NG-rank method, that should only keep relevant words. However, we conjecture that not all the words in $\widehat{M}(D_i)$ are relevant in the same way. To this end, we partition $\widehat{M}(D_i)$ into two sets: *Hard Relevant* (*HR*) and *Soft Relevant* (*SR*) ones. For example, in Figure 1B, *HR* = {*market, transact, sale*} and *SR* = {*properti, hous, land, quarter*}. To this end, we first bisect the graph G_i using algorithm METIS [12], with the aim of minimizing the sum of the weights associated with all the edges crossing the cut. At the end, we obtain two clusters of words, where each cluster consists of terms that are semantically relevant to each other. Discarding SR words should avoid the extra noises that expanding such words may cause, thus, affecting the quality of the clustering results. To distinguish HR words from SR words, we utilize the scores assigned to the keywords of the document summary by our algorithm NG-rank (see Ref. [17] for more details). Let $NG(D_i) = \{(w_1, g_1), (w_2, g_2), \ldots, (w_m, g_m)\}$ be the summary of document D_i extracted by NG-rank, where $\widehat{M}(D_i) \subseteq NG(D_i)$, and g_i is the score assigned to the keywords w_i of the summary. Hence, considering that METIS partitions $\widehat{M}(D_i)$ and produces two sets P_1 and P_2, we identify either P_1 or P_2 as the HR set as follows:

$$HR = \begin{cases} P_1 & \text{if } \dfrac{\sum\limits_{P_1 \in G} g_{i|w_i \in P_1}}{|P_1|} > \dfrac{\sum\limits_{P_2 \in G} g_{i|w_i \in P_2}}{|P_2|} \\ P_2 & \text{otherwise} \end{cases}$$

In practice, we compute the average score of P_1 and P_2, and then identify as the HR set the one with the highest average score.

Finally, given document D_i, we denote by $\overline{M}(D_i)$ the HR words in $\widehat{M}(D_i)$. We only use $\overline{M}(D_i)$ to expand the document D_i by exploiting the semantic relations in WordNet.

4 Experimental Setup

The principal idea of the experiments is to show the efficacy of the document-enriching method on clustering results through a manually predefined categorization of the corpus. We used “BBC NEWS” to test the effect of using our document-enriching method on clustering quality. Moreover, we also exploited Document Understanding Conferences “DUC 2002” dataset for testing the quality of the summarization method (NG-rank), which, in turn, is used to extract salient entities from text, and "Senseval" to evaluate our proposed WSD approach.

The three corpora are discussed below, along with the preprocessing applied to them and, finally, the evaluation measures used in the experimental tests.

BBC NEWS: This dataset consists of 2225 documents from the BBC News website corresponding to stories in five topical areas, which are named Business, Entertainment, Politics, Sport, and Tech, from 2004 to 2005 [9]. We use two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ of 500 documents each, obtained by randomly selecting about 100 documents for each topical area. BBC News is a dataset full of entities to be linked. Moreover, the documents can be split in paragraphs, a feature that is needed to apply our summarization technique NG-rank [17].

DUC2002: DUC2002 contains 567 document summaries, where documents are clustered into 59 topics, and each topic contains about 10 documents. For each topic, there are seven summaries, namely, *10*, *50*, *100*, *200*, *200e*, *400e*, and *perdocs*, which are written by experts. The summary 10 is a 10-word summary of all the documents included in a topic. Similarly, summaries 50, 100, and 200 are 50-word, 100-word, and 200-word summaries of all the documents included in a topic. Summaries 200e and 400e are created by extracting important sentences from the documents of each topic. The last type of summaries is *perdocs*, which is a single separate summary of 100 words for each single document of a given topic. For our evaluation, we only used summaries 10, 50, 100, and 200 words, along with *perdocs*.

Senseval: Senseval is the international organization devoted to the evaluation of WSD systems. Its activities, which started in 1997, aim at organizing and running evaluation of WSD systems for different words, different aspects of language, and different languages [5]. Senseval-1 (1998) focused on the evaluation of WSD systems on a few major languages (English, French, Italian) that were available in corpus and computerized dictionary. The successive Senseval extended the language coverage and increased the number of tasks considered. We use Senseval-1 for our experiments on the English language.

4.1 Preprocessing and Evaluation Measures

Besides stop word removal, lower case conversion, and stemming (http://tartarus.org/martin/PorterStemmer/), we also identify sentences and paragraphs in each document. We also preprocess the WordNet ontology, to create data structures that allow a fast navigation of semantic relations.

For evaluating the clustering results, we use the well-known *Purity* measure [19]. For evaluating our WSD systems and comparing with baseline, we simply use *accuracy*, which is the percentage of correctly identified word senses.

5 Experimental Results

As previously stated, we first evaluate *NG-rank* as a method for extracting salient entities. Then, we assess the quality of our proposed WSD approach. Finally, we assess the quality of results obtained from the document clustering after applying document enrichment.

5.1 Assessing the quality of the salient entities extracted by NG-rank

For a given document d belonging to a topic within DUC2002, we first create a set including all the words of the mentions related to the entities appearing in d, namely, $DP(d)$. This step, which still leverages the Dandelion API, aims to prepare the *ground truth* used to assess our method to identify entities that are salient. We then take the five summaries $SU_i^{Exp}(d)$ of document d created by experts and included in DUC2002. Each $SU_i^{Exp}(d)$, $i = \{1, \ldots, 5\}$, has a different length, as we use the DUC2002 summaries of 10, 50, 100, and 200 words, and *perdocs* one, respectively. From the summaries, we produce five corresponding sets of words $SP_i^{Exp}(d)$, where $SP_i^{Exp}(d)$ includes those words of SU_i^{Exp} that are in common with the ones of $DP(d)$. As a consequence, we can also rank the saliency of entities mentioned in d by considering the length of the DUC2002 summary where each word in $DP(d)$ appears. An entity whose mention occurs in a 10-word summary is more salient than the one whose mention only appears in a 50-word summary of d.

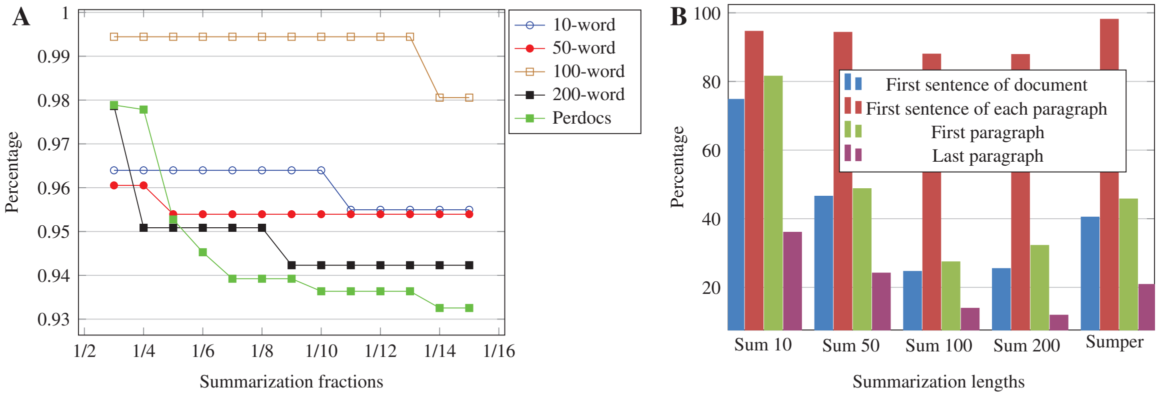


Figure 2: The plots of the obtained results described (A) Covering of the Salient Entities Mentioned in the Documents. (B) Percentage of Mentioned Salient Entities in the Sections of DUC Documents.

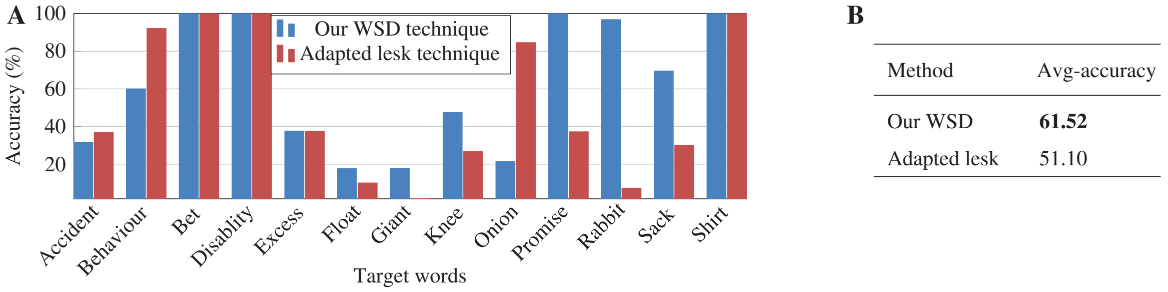
To evaluate our summarization method, we created 13 summaries of different sizes by applying NG-rank, where $SU_j^{NG-Rank}(d)$ denotes each summary. Afterward, for each document d, we can compare each $SU_j^{NG-Rank}$ with the terms included in the various SP_i^{Exp}.

Figure 2A shows the obtained results, where we plot the measure R to evaluate the *covering* of the salient entities mentioned in the document. R is maximum (equal to 1.0) when all the entity spots in $SP_i^{Exp}(d)$ are found in the summary $SU_j^{NG-Rank}(d)$. The abscissas of the plot in Figure 2A indicate the summarization fractions of $SU_j^{NG-Rank}$. The ordinates are, indeed, the *average R* (*percentage*) for all the documents of the dataset, where each curve of the plot refers to a given SP_i^{Exp}, i.e. the entity mentions occurring in expert-produced summaries of a given length in DUC2002. Note that the average numbers of words mentioning salient entities in the summaries generated by the NG-rank method are in the range of 7.78 words (summarization fraction $= 1/3$) to 37.26 words (summarization fraction $= 1/15$). These words cover the actual salient entities, as identified in the ground truth, for more than 93% in the worst case. For example, the text included in the NG-rank summaries of size 1/10 covers more than 99% of the mentions appearing in the expert-based 100-word summaries.

Because NG-rank, i.e. the summarization method used in this paper, emphasizes the *first* and *last* sentences of each paragraph to summarize a document, we also investigated the average percentage of mentions to entities occurring in the different sections of the DUC2002 documents. As you can note from Figure 2B, the first sentences of each paragraph are remarkable, the section of a document that includes most of the mentions to entities present in the document.

5.2 Assessing the Quality of our WSD Method

To assess the quality of our WSD technique, we use a Python implementation of the well-known baseline (Liling Tan. 2014. Pywsd: Python Implementations of Word Sense Disambiguation (WSD) Technologies [software]. Retrieved from https://github.com/alvations/pywsd) adapted lesk algorithm [1], which makes use of the lexical items from semantically related senses within the WordNet hierarchies to generate more lexical items for each sense. Because in our WSD method the context used for disambiguating a target word only consists of nouns (i.e. spot nouns of salient entities within the summary of the document), for all tests operating on Senseval-1 (http://www.senseval.org/), we limit ourselves to only noun words for which there are WordNet mappings in Senseval. We compare our selected senses for the various target words with the "gold standard" senses, which are the sense that the human sense-tagging team considered correct for each corpus instance. The detailed results, concerning the accuracy of our method in comparison with the baseline for each target noun word in the gold standard, are shown in Figure 3A. The average accuracy for all the target nouns is shown in Figure 3B.



Method	Avg-accuracy
Our WSD	**61.52**
Adapted lesk	51.10

Figure 3: The obtained results of assessing the quality of our WSD method that show (A) the Accuracy Obtained by our WSD Method and the Adapted Lesk Algorithm on 13 Noun Target Words. (B) Average Accuracy for all the Target Words, indicating a considerable improvement in result (shown in bold) compared to the Adapted lesk method.

Using the Senseval contexts, made of one or more sentences including a given target word, the results obtained by our WSD method are very good and are substantially better than the baseline.

5.3 Clustering Results

In this section, we finally evaluate how our overall algorithm (SEED) is able to improve the quality of text clustering. The algorithm adopted for clustering is always the k-means, while the vectorial representation of documents is based on a classical *tf-idf* weighting of terms, and the measure of similarity between vector pairs is the Cosine one. We utilize *RapidMiner* (https://rapidminer.cosm/products/studio/), which is an integrated environment for analytics and also providing tools for text mining. As stated in Section 4, for testing clustering quality, we use two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ of BBC NEWS.

In the first experiment on $\mathcal{S}_1$, whose results are reported in Table 1, we evaluate the quality of clustering when we only use salient entities and the associated categories and spots to enrich the original documents. First, we show in Table 1(a) the results obtained when we exploit the state-of-the-art approach NEKW [3], which uses all the entities mentioned in a document, not only the salient ones as SEED.

From Table 1(b), we observe a slight improvement (1.37%) in the total purity of clustering when we use salient entities and their categories only, instead of using all the entities and their categories as NEKW does. The utility of only using the mentions of salient entities is shown in Table 1(d), in which we observe an improvement (0.9%) over a method that exploits categories of salient entities. However, the utility of using spots of salient entities in improving the quality of clustering is finally reported in Table 1(f), in which we can see a considerable improvement (3.67% in the total purity) over NEWK when we use the mentions of salient entities only or even using the spots of all the entities in Table 1(e) (2.72% in the total purity).

In the second experiment, we evaluate the quality of the document clustering obtained by our full SEED method, which also exploits WordNet to expand the mentions of salient entities. The results are reported in Tables 2 and 3. Table 2 shows 14 distinct partitions of the same collection, indeed, $\mathcal{S}_1$ or $\mathcal{S}_2$, obtained by clustering with document representations obtained using different instances of function $\mathcal{F}$, namely, $\{\mathcal{F}^h\}_{h=1,2,..,14}$. The first column of Table 2 indicates the $\mathcal{F}$-based representation of documents, whereas the other columns are the results of clustering using the specified representation. The range of parameters $\alpha, \beta, \gamma, \varepsilon, \delta, \eta$ in $\mathcal{F}^h$ is bounded by an empirical value v that takes into account the document lengths and the sizes of the various bags of words extracted from (or enriching) the documents, considering a threshold

Table 1: Clustering Results on Subset $\mathcal{S}_1$ Using Original Documents Plus.

	(a)	(b)	(c)	(d)	(e)	(f)
Total purity	0.873	0.885	0.877	0.893	0.881	0.905

(a) All the entities and their categories (NEKW method); (b) salient entities and their categories; (c) all the entities and their spots; (d) salient entities and their spots; (e) all the spots of the entities; (f) the spots of salient entities.

Table 2: Partitions of BBC NEWS (Subsets $\mathcal{S}_1$ and $\mathcal{S}_2$) Obtained by k-Means with Different Function $\mathcal{F}$, Entailing Different Document Representations.

	Purity of clusters										Total purity subset	
	Subset $\mathcal{S}_1$					Subset $\mathcal{S}_2$						
Representation	**C0**	**C1**	**C2**	**C3**	**C4**	**C0**	**C1**	**C2**	**C3**	**C4**	$\mathcal{S}_1$	$\mathcal{S}_2$
$PYCEM^2O$	1.0	0.925	0.827	0.843	0.913	0.588	0.569	0.568	0.888	0.857	0.897	0.687
$PYCUEM^3O$	0.947	0.855	0.894	0.943	0.898	0.645	0.634	0.596	0.991	1.0	0.905	0.756
$\{PYC\}^2EM^5UO$	0.989	0.868	0.902	0.898	0.940	0.634	0.606	0.568	1.0	1.0	0.917	0.704
$P^4Y^2C^4EM^{11}O$	1.0	0.690	0.955	0.836	0.881	0.597	0.586	0.573	0.905	0.867	0.850	0.697
$\{PYC\}^5EM^{11}U^2O$	0.931	0.756	0.924	0.914	0.920	0.797	0.620	0.565	0.981	0.95	0.880	0.737
$\{PYC\}^4EP^7U^2O$	0.934	0.840	0.917	0.898	0.923	0.705	0.598	0.592	0.991	0.987	0.901	0.762
$\{PYC\}^3E^2M^7UO$	0.937	0.861	0.935	0.917	0.941	0.618	0.598	0.598	0.893	0.897	0.917	0.714
$P^2\{YC\}^3E^2M^6O$	0.957	0.908	0.835	0.905	0.950	0.629	0.602	0.576	0.872	0.918	0.909	0.706
$PYCEMUO$	1.0	0.978	0.874	0.817	0.979	0.792	0.596	0.636	1.0	1.0	0.921	0.768
$PYCEM^2O$	1.0	0.978	0.883	0.803	0.979	0.770	0.619	0.591	0.983	1.0	0.919	0.768
$\{PYC\}^2EMUO$	1.0	0.927	0.941	0.814	0.979	0.729	0.604	0.588	0.991	0.987	0.927	0.752
$\{PYC\}^2EM^2UO$	1.0	0.927	0.950	0.816	0.970	0.677	0.636	0.610	0.991	1.0	0.927	0.764
$\{PYC\}^3EM^2UO$	0.979	1.0	0.896	0.773	0.975	0.717	0.607	0.582	0.982	0.987	0.911	0.748
EUO	1.0	0.937	0.9	0.768	0.979	0.770	0.609	0.510	0.990	0.987	0.907	0.731

O, OrigDoc; M, MentionsEN; P, PredSyns; Y, Syns; C, SuccSyns; U, SummDoc; E, NamesEn.

Table 3: Clustering Results.

	(a)	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**
Total purity	0.858	0.656	0.873	0.941	0.681	0.779

(a) Original documents of subset $\mathcal{S}_1$; (b) original documents of subset $\mathcal{S}_2$; (c) using *NEKW* on dataset $\mathcal{S}_1$; (d) using SEED on dataset $\mathcal{S}_1$; (e) using *NEKW* on dataset $\mathcal{S}_2$; (f) using SEED on dataset $\mathcal{S}_2$.

of minimum occurrence of words in the documents in order to limit repetition of the bags. The summaries produced by the NG-rank method are 1/3 of the original documents.

Table 2 also shows the utility of enriching documents by expanding mentions of salient entities. Consider that the best purity measures we can obtain by clustering with the original vectorial representations (plain clustering) of the two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ are 0.858 and 0.656, respectively. We can see the positive effect of using the *PYC* representation (WordNet-based enrichment) in the final clustering quality by looking at the 11th row: we improved by 8% on $\mathcal{S}_1$ and by 14.63% on $\mathcal{S}_2$ over plain clustering. Moreover, the use of document summaries and salient entities (see the *EU* representation at the 14th row) also improves the clustering quality: we obtained 5.7% and 11.43% of improvements on $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, over plain clustering. In order to improve uniformly the purity of all the clusters obtained, SEED combines the various clustering results by exploiting an ensemble method. The final clustering is, thus, a consensus one, whose results obtained by combining the 14 distinct partitions of $\mathcal{S}_1$ and $\mathcal{S}_2$ (shown in Table 2) are reported in Table 3(d) and (f).

For clustering $\mathcal{S}_1$, as many document representations can be generated by replicating the bags of words in *SES*, we selected the top-N partitions (with the highest *total purity*) obtained using these representations. In this way, we obtained the 14 partitions shown in Table 2, where each partition is associated with a different instance of $\mathcal{F}$. We can consider this phase as a sort of *learning phase*, used to determine the best document enrichments. The same learned $\mathcal{F}$ configurations were used to obtain the 14 partitions of $\mathcal{S}_2$, also shown in Table 2.

SEED finally obtains 9.67% and 18.75% improvements of the overall purity – shown in Table 3(d) and (f) – on subsets $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, over the results obtained using the original document representations – shown in Table 3(a) and (b). We also observed that the results obtained by *NEWK* on subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ improved by 7.8% and 14.4%, shown in Tables 3(c) and (e). It is worth noting that the best improvements were obtained by the consensus algorithm on subset $\mathcal{S}_2$, where the types of enrichment, used for producing the various partitions to combine, were decided on subset $\mathcal{S}_1$.

6 Conclusion

This paper presented a multi-strategy algorithm to extract the most salient entities cited in a document and, in turn, exploited to semantically enrich the document. Our experiments indicate that we can exploit this knowledge about the most salient entities to effectively enrich documents with the latent information extracted from WordNet, thus, finally improving clustering quality. As a future work, as salient entities are a way to highlight the main topics in a document, we plan to extend our approach to investigate the automatic cluster labeling.

Bibliography

[1] S. Banerjee and T. Pedersen, An adapted Lesk algorithm for word sense disambiguation using WordNet, in: *Int. Conf. on Intel. Text Processing and Computational Linguistics*, pp. 136–145, Springer, Berlin, Heidelberg, 2002.

[2] D. M. Blei, Probabilistic topic models, *Comm. ACM* **55** (2012), 77–84.

[3] T. H. Cao, T. M. Tang and C. K. Chau, Text clustering with named entities: a model, experimentation and realization, in: *Data Mining: Foundations and Intelligent Paradigms*, pp. 267–287, Springer, Berlin, Heidelberg, 2012.

[4] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego and S. Trani, Learning relatedness measures for entity linking, in: *Proc. of CIKM '13*, pp. 139–148, ACM, San Francisco, California, USA, 2013.

[5] P. Edmonds, SENSEVAL: the evaluation of word sense disambiguation systems, *ELRA Newsletter* **7** (2002), 5–14.

[6] M. van Erp, P. N. Mendes, H. Paulheim, F. Ilievski, J. Plu, G. Rizzo and J. Waitelonis, Evaluating entity linking: an analysis of current benchmark datasets and a roadmap for doing a better job, in: *Proc. of LREC'16*, 2016.

[7] C. Fellbaum, *WordNet*, Wiley Online Library, 1998.

[8] P. Ferragina and U. Scaiella, Tagme: on-the-fly annotation of short text fragments (by wikipedia entities), in: *Proc. of CIKM'10*, pp. 1625–1628, ACM, Toronto, ON, Canada, 2010.

[9] D. Greene and P. Cunningham, Practical solutions to the problem of diagonal dominance in kernel document clustering, in: *Proc. of ICML'06*, pp. 377–384, ACM, Pittsburgh, Pennsylvania, USA, 2006.

[10] B. Hachey, W. Radford, J. Nothman, M. Honnibal and J. R Curran, Evaluating entity linking with Wikipedia, *Artif. Intell.* **194** (2013), 130–150.

[11] R. Kadlec, M. Schmid, O. Bajgar and J. Kleindienst, Text understanding with the attention sum reader network, In: *Proc. of ACL'16*, pp. 908–918, Berlin, Germany, 2016.

[12] G. Karypis and V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM J. Sci. Comput.* **20** (1998), 359–392.

[13] R. Mihalcea and A. Csomai, Wikify!: linking documents to encyclopedic knowledge, in: *Proc. of CIKM '07*, pp. 233–242, ACM, Lisbon, Portugal, 2007.

[14] D. Milne and I. H. Witten, Learning to link with Wikipedia, in: *Proc. of CIKM '08*, pp. 509–518, ACM, Napa Valley, California, USA, 2008.

[15] S. Montalvo, R. Martnez, V. Fresno and A. Delgado, Exploiting named entities for bilingual news clustering, *J. Assoc. Inf. Sci. Technol.* **66** (2015), 363–376.

[16] R. Navigli, Word sense disambiguation: a survey, *ACM Comp. Surveys* **41** (2009), 10.

[17] M. Pourvali, S. Orlando and M. Gharagozloo, Improving clustering quality by automatic text summarization, *Inf. Retrieval Technology*, pp. 292–303, Springer, Cham, 2015.

[18] D. Reforgiato Recupero, A new unsupervised method for document clustering by using WordNet lexical and conceptual relations, *Inf. Retrieval* **10** (2007), 563–579.

[19] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. P. L. F. de Carvalho, and J. Gama, Data stream clustering: a survey, *ACM Comput. Surveys* **46** (2013), 13.

[20] A. Strehl and J. Ghosh, Cluster ensembles – a knowledge reuse framework for combining multiple partitions, *J. Mach Learn. Res.* **3** (2003), 583–617.

[21] S. Vega-Pons and J. Ruiz-Shulcloper, A survey of clustering ensemble algorithms, *J. Pattern Recognit. Artif. Intell.* **25** (2011), 337–372.

[22] X. Zhang, L. Jing, X. Hu, M. Ng and X. Zhou, A comparative study of ontology based term similarity measures on PubMed document clustering, in: *Advances in Databases: Concepts, Systems and Applications*, pp. 115–126, Springer, Berlin, Heidelberg, 2007.