HAL
open science

Towards a Term Clustering Framework for Modular Ontology Learning

Ziwei Xu, Mounira Harzallah, Fabrice Guillet, Ryutaro Ichise

To cite this version:

Ziwei Xu, Mounira Harzallah, Fabrice Guillet, Ryutaro Ichise. Towards a Term Clustering Framework for Modular Ontology Learning. Knowledge Discovery, Knowledge Engineering and Knowledge Management, pp.178-201, 2020, 10.1007/978-3-030-49559-6_9 . hal-03063773

HAL Id: hal-03063773

https://hal.science/hal-03063773v1

Submitted on 14 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards a Term Clustering Framework for Modular Ontology Learning

Ziwei Xu[1], Mounira Harzallah[1], Fabrice Guillet[1], and Ryutaro Ichise[2]

[1] LS2N, Ecole Polytechnique de l'Universit de Nantes, Nantes, 44300, France
{ziwei.xu, mounira.harzallah, Fabrice.Guillet@univ-nantes.fr}
[2] National Institute of Informatics, Tokyo,101-8430, Japan
ichise@nii.ac.jp

Abstract. This paper aims to analyze and adopt the term clustering method for building a modular ontology according to its core ontology. The acquisition of semantic knowledge focuses on noun phrase appearing with the same syntactic roles in relation to a verb or its preposition combination in a sentence. The construction of this co-occurrence matrix from context helps to build feature space of noun phrases, which is then transformed to several encoding representations including feature selection and dimensionality reduction. In addition, word embedding techniques are also presented as feature representation. These representations are clustered respectively with K-Means, K-Medoids, Affinity Propagation, DBscan and co-clustering algorithms. The feature representation and clustering methods constitute the major sections of term clustering frameworks. Due to the randomness of clustering approaches, iteration efforts are adopted to find the optimal parameter and provide convinced value for evaluation. The DBscan and affinity propagation show their outstanding effectiveness for term clustering and NMF encoding technique and word embedding representation are salient by its promising facilities in feature compression.

Keywords: Text Mining · Feature Extraction · Ontology Learning · Term Clustering.

1 Introduction

Ontology building is a complex process composed of several tasks: term or concept acquisition, concept formation, taxonomy definition, ad-hoc relation definition, axiom definition [17]. The ever-increasing access to textual sources has motivated the development of ontology learning approaches based on techniques of different fields, like natural language processing, data mining and machine learning. Many works are focused on the taxonomy definition and more especially on the hypernym relation extraction. A term t1 is a hypernym of a term t2 if the former categorizes the later. This relation is also known as a terminological is-a relation. For its extraction from texts, several approaches based on Harris distributional hypothesis are proposed. This hypothesis states that words/terms in the same context can have similar meanings [27]. Then each term can be represented by a numeric vector in a vector space by taking into account the context, with different word embedding techniques (e.g. co-occurrence matrix, word2vec, NMF, etc.)

Based on the geometric similarity in a vector space, non-supervised methods are applied for term clustering. Due to the concerns about the semantic relation between terms upon the construction of vector space, each cluster is expected to include semantically similar terms (i.e. synonyms or related by the hypernym relation) or semantically connected terms.

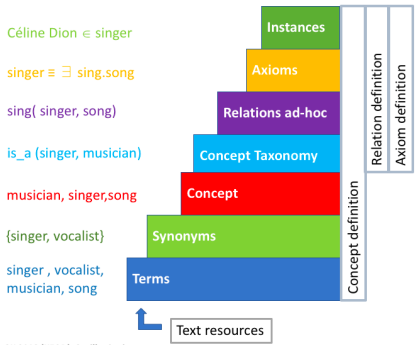
In case that the semantic meaning of clusters could not match any existing concepts of ontology, these clusters are not suitable for ontology building. Moreover, these approaches may have poor performance due to the sparsity of the co-occurrence matrix [4]. Dimensionality reduction becomes a crucial issue. It can be performed by feature selection. In the statistical stage, feature selection could be achieved by the frequency of terms or the weighting of Tf-Idf (term frequency-inverse document frequency).

Clustering terms under the core concepts of ontology are demonstrated to be productive to build a modular ontology [33]. A core ontology of a domain is a basic and minimal ontology composed only of the minimal concepts (i.e core concepts) and the principal relations between them that allow defining the other concepts of the domain [40, 5]. This step (i.e. term clustering under core concepts) is the first stage towards a taxonomy definition. Indeed, a term of each cluster is expected to be synonym or hyponym of the core concept that corresponds to its cluster. Later, inside of each cluster, other hypernym relations between terms have to be extracted.

In this paper, we will group terms under core concepts through clustering algorithms and to evaluate these clustered terms whether they are synonym, hypernym or semantically close to core concepts. Accordingly, we define and evaluate several frameworks for term clustering by varying feature representations (i.e. co-occurrence representation, weighted co-occurrence representation, NMF representation, and word embedding representation) and clustering techniques(i.e. k-means, k-medoids, affinity propagation, DBscan and co-clustering). We present the ontology building steps from core ontology in section 2. The related works are discussed in terms of term clustering for ontology building in section 3. We then describe the corpus and the pre-processing steps served for feature representation in section 4. Sequentially, we discuss the parameters setting of these five clustering techniques, analysis their results. Finally, we conclude with the term clustering techniques recommendation for ontology building purpose.

The main differences between this paper and previous work [52] can be summarized regarding to the extension of content and the augmentation of experiments. We progressively describe the ontology building procedures from 'core ontology' to 'modular ontology', and to our proposed ontology in section 2. Furthermore, we detail the interesting clustering methods about their advantages and disadvantages and show their utility over ontology building in previous work. In terms of experiments, we extend our operation with three additional clustering techniques: k-medoids, DBscan and co-clustering, and update the existing experiments with a much enlarged gold standard. Ultimately, five different clustering techniques are compared together with their fresh results in order to offer a broader comparison upon term clustering techniques.

Fig. 1. The Ontology learning cake from [3] with modification



2 Ontology Building

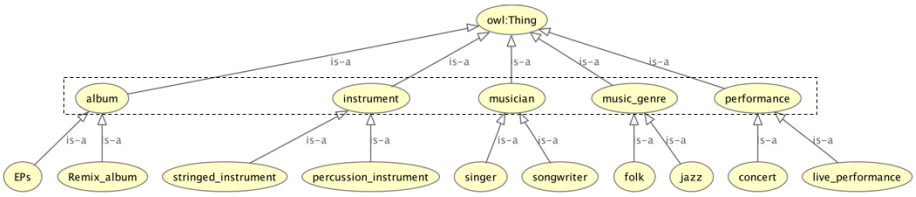
Ontology building from text could be achieved by various approaches, it could be performed manually, automatically or semi-automatically. During the ontology construction procedures, respecting to the sequence of manipulation, ontology building is able to be divided into the bottom-up approach, top-down approach, and mixed approach. However, the step of human validation is irreplaceable at the end of ontology building, to ensure the accuracy of knowledge representation in the constructed ontology.

Ontology conceptualization is the core part of ontology building. It can be simplified into this ontology learning layer cake [3] in Fig. 1. As shown in this cake, starting with terms from text, several steps are followed to explore concepts and their corresponding relations. For example, in the music domain, the terms 'singer', 'vocalist', 'musician' and 'song' are extracted. Then, term synonyms are identified and grouped to form concepts (e.g. the synonyms terms 'singer' and 'vocalist' are grouped and constitute the concept). From these isolated terms, we can find their synonyms. At the same time, we can infer the relations between them. It could be the simple is-a relation or more complex ad-hoc relation. Once enough relations are dug out, it is interesting to find the axioms between these relations. In our approach, we concentrate on the bottom three steps, from term extraction to synonyms identification and to concept definition. From these stages, we are allowed to cluster the extracted terms to form concepts where each cluster includes synonyms or hypernyms of core concepts. Then within each cluster, further synonym or is-a relations between terms of the same cluster can be extracted.

2.1 Core Ontology

To steer the learning process of a domain ontology, we benefit from a domain core ontology. A core ontology of a domain is the basic and minimal concepts (i.e. core concepts) and the principle relations between them that allow defining the other concepts of the domain [5, 38, 21]. Scherp [46] considers that a core ontology should be characterized by a high degree of axiomatization and formal precision. Nevertheless, it could be presented by a concept taxonomy structure with is-a relation as Fig. 2 shows.

Fig. 2. The core ontology and its sub concepts.



Furthermore, in a core ontology, generally, each core concept (except 'Thing') represents (conceptualize) a sub-domain (a topic) of the ontology domain and it could be specialized on sub-concepts in order to define the sub-domains (see Fig. 3).

A core ontology could be considered as an upper ontology (i.e. top-level ontology or foundation ontology [5]) of domain ontology, which provides the high possibilities to be reused for extensive purpose. In most cases, the core ontology is predefined by a domain expert, in order to provide guidelines in terms of domain ontology construction.

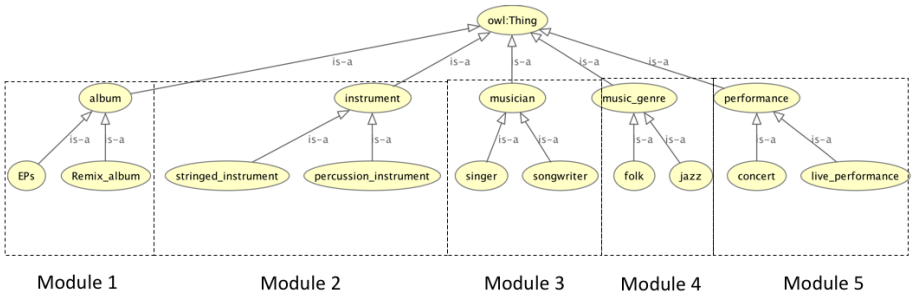
On the basis of core ontology, Gruber [25] suggests using core ontology of a domain to build domain ontology. Additionally, several works define or reuse a core ontology to identify and further define the domain concepts by specialization. For instance, on the one hand, almost all OBOs (Open Biomedical Ontologies) have been originated by importing the BFO (Basic Formal Ontology) and the RO (Relation Ontology); Opdhal et al. [39] used BWW (Bunge Wand Weber) ontology to build the UEML ontology; Chulyadyo et al. [7] improved the ontology flatness by inferring hypernym relation between extracted terms and core concepts. On the other hand, some works map a core ontology to a given domain ontology, so as to better define the concepts of the domain and superimpose a structure of one domain ontology. For example, Deprs et al. [11] map the Core Legal Ontology (CLO) to legal Ontologies; Burita et al. [5] map NEC (Network Enabled Capabilities) core ontology to the NEC domain ontology.

2.2 Modular Ontology

Modular ontology is considered as a major topic to facilitate and simplify the ontology engineering process in the field of formal ontology developments [29]. If it is required to alter the structure of the ontology, we can just remove, add or enrich the target modules in modular ontology, without interference to other remaining parts of ontology. Moreover, the modular representations are easier to understand, reason with, extend and reuse [24]. Therefore, using these representations tends to reduce the complexity of designing and to facilitate ontology reasoning, development, and integration [13].

Gangemi [22] and Kutz [33] suggest to map core ontology to domain ontology for improving modularity. On this basis of core ontology, it is interesting to obtain a well-structured taxonomy where each sub-domain is defined by a separate module (Fig. 3). Then it becomes easier to define a modular regarding each core concept that represents its sub-domain. The constituted several main topics (i.e. core concepts) in a specific domain will lead the extension of sub-concepts (bottom layer in Fig. 3).

Fig. 3. The domain modular ontology.



2.3 Our Target Ontology

Concisely, we aim to build modular ontology from text using term clustering derived by core ontology. Following the top-down approaches from the core ontology to modular ontology in order to build an ontology, we would like to enrich each module by concepts/terms, through extracting terms and clustering them where each cluster should correspond to the terms/concepts of a module. We start by analysing term clustering frameworks and comparing their suitability to put terms semantically close to a core concept (i.e. synomym or hypernym terms of a core concept) in the same cluster. For that, it is required to evaluate whether the resulted clusters are close to a manual term classification.

In our work, a clustering framework concerning NPs as terms and it depends of three main components: 1) feature representation approaches, 2) dimension reduction techniques, 3) and clustering algorithms. These components allow to be substituted by the different related techniques, which brings the high flexibility for the entire term clustering framework.

3 Related Work

3.1 Feature Representations

In the field of knowledge acquisition from text, it is apparent that the functional entities of sentences and their clauses constitute the dominant linguistic elements for syntagmatic information collection. Cimiano [9] describe the local context by extracting triples of nouns, their syntactic roles, and co-occurred verbs. They consider only verb/object relations, so as to emphasize partial features of terms working as an object by a conditional probability measure, which calculate the conditional probability that a certain term appears as head of a certain argument position of a verb. Similarly, Jiang [31] and Rios-alvarado [45] formed the triple term structure of noun phrases and verbs, in the shape as subject of the noun, verb, the object of the noun. Moreover, ASIUM [16] acquires semantic knowledge from the following canonical syntactic frames which include the verb, and their preposition or syntactic roles and the headword of noun phrases:

$$< to\ verb > \quad ((< preposition > | < syntactic\ role >) \quad < headword >)$$

For examples, the instantiated syntactic frame of the clause, "Bart travels by a huge boat", we get:

$$< to\ travel > \quad < subject > \quad < \mathbf{Bart} >$$
$$< by > \quad < \mathbf{boat} >$$

It is evident that their focus is based on the dependency between the verb (i.e. 'to travel') and features of the verb (i.e. 'Bart' with syntactic roles 'subject'; 'boat' with preposition 'by'). Except for the extraction of nouns and verbs, some work consider the involvement of adjectives as well, which would be considered as keywords of ontology learning [50, 43].

Besides syntactic dependency, one recent work [19] extracts co-occurring couples of entities and present their semantic relations with pattern-based representation. To interpret these appearances, terms (entities) are presented by vectors with the frequent sequential pattern as components. Then pattern-based feature space is constructed for relation discovery. Moreover, according to Word2vec [36], a term is statistically encoded with analogies from its appearance in a different context, where the similarity of encoding vectors reflect the semantic relations between terms.

3.2 Dimensionality Reduction Techniques

After the choice of the feature representation and the building of term-feature matrix, often we have to deal with matrix sparsity problem using dimensionality reduction techniques. Church et al. [8] proposed to apply PMI weighting (pointwise mutual information) to reduce bias in rare contexts, in which values below 0 are replaced by 0. Tf-Idf (term frequency-inverse document frequency) also contribute to weight terms by their specificity to documents. The computational complexity grows exponentially with the size of the lattice, where NMF (non-Negative Matrix Factorization) [34] is dedicated to solving the dimensionality reduction problem by performing feature compression.

3.3 Clustering Techniques

K-Means The most typical clustering technique is k-means, which starts with randomly selected centroids and performs iterative calculations to optimize the positions of the centroids for partition purpose [28]. It is easy to be implemented and widely used as a simple clustering solution. However, its drawbacks are also evident that 1) k-means is quite sensitive to the initial set of seeds; 2) its performance could be strongly impacted by the noisy elements. Despite that, k-means is always regarded as the baseline to compare with other clustering algorithms.

K-Medoids Similar to k-means clustering algorithm, k-medoids also attempts to minimize the distance between centroids. In contrast to k-means, k-medoids choose the starting centroids as priori before calculation [32]. K-medoids provides many favorable properties: 1) it presents no limitations on attributes types, which means it is capable of numerical, categorical and binary attributes. 2) the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster and, therefore, it is

lesser sensitive to the presence of outliers. Briefly, it is more robust to noise and outliers as compared to k-means. However, this algorithm suffers from the negative effects of unsuitable initial seeds, because it does not allow reassigning seeds while changing mean values. Nevertheless, it could be a preferable clustering algorithm for us once we acknowledge the proper starting seed for each cluster.

Affinity Propagation Like k-medoids, affinity propagation clustering algorithm finds centroids to represent their located clusters during iterations. Differ from the dissimilar distance in k-medoids, affinity propagation uses graph distance that performs in a message passing way between data points [18]. With this approach, 1) it is not required to determine the number of clusters in advance and 2) the centroid of each cluster is specified after calculation, which turns out to be helpful for cluster interpretation. However, this algorithm is not friendly with big datasets because the time complexity of calculation increases dramatically along with the amount of clustered elements. Nevertheless, affinity propagation is still interesting as clustering algorithm for normal-size datasets.

DBscan Despite those distance-based clustering methods, DBscan (Density-based spatial clustering of applications with noise) [15] is distinguished as a density-based clustering method. It groups together closely packed points and marks the low-density points as outlier points, in order to accentuate the high-density points into clusters and get rid of the negative impacts of outliers. DBscan clustering algorithm has some special benefits: 1) it is capable to find arbitrarily shaped clusters, because of the reduced single-link effect (different clusters being connected by a thin line of points) 2) no demand to specify the number of clusters as that of affinity propagation. In opposite, DBscan allows for points to be part of more than one cluster, which might induce overlapping between clusters. It requires the knowledge of domain expert during the selection of key parameters, such as the minimum number of points required to form a dense region (i.e. minPts) and the radius of a neighborhood with respect to some points (i.e. eps). It is desirable to apply DBscan clustering algorithm even with several pre-experiments for the selection of parameter.

Co-clustering In co-clustering algorithm (also called bi-clustering, block clustering), not only the targets but also the features of the targets can be clustered simultaneously, which preserves the existing relation between targets and their features. We are interested in the bi-clustering over contingency table [23]. Typically, the input matrix would be arranged as a two-way contingency table. This algorithm shows the encouraging performance on the contingency outcomes. The co-clustering has practical importance in gene research and also document classification. The resulted co-clusters are expected to overlap with each other, where these overlaps themselves are often of interest. It has two major shortcomings: 1) the problem of local optimization to each co-cluster individually; 2) the lack of a well-defined global objective during each iterations [40]. Despite these facts, the co-clustering algorithm is attractive because it takes into account the relation between clustered elements and the features of them.

In previous work, Clustering techniques have shown their favorable properties in terms of ontology learning. The K-means clustering algorithm was implemented to

separate the domain knowledge for the purpose of domain ontology learning [47]. One adaptive k-medoids clustering method [14]could be applied to identify clusters by these medoids, which are representing the concepts of ontology from the knowledge database. Except for the typical clustering algorithm, there are many calculation approaches used for clustering purpose. The Weka data mining tool [51] helps to implement many algorithms for clustering purposes, such as viz., EM, Farthest First and k-Means. The previous research showed that the Farthest First clustering technique yielded rather better performance than the others in the attempt of concept clustering. The Farthest-First [37] is a variant of K-Means that differs in the initial centroid assignment, which places the cluster center at the point furthest from the existing centers. on the other hand, Hao [26] aimed to construct a hierarchy of ontology by using EM algorithm [10] to cluster the keywords from domain corpus. EM computes the distribution of parameters for each cluster by the maximum likelihood criteria. Hao [26] implemented EM several times to select the appropriate number of clusters and then summarized the subject of the cluster for the convenience of hierarchy construction and organization.

Briefly, many clustering algorithms have participated in the procedures of ontology learning. In previous work, the output of automatic term clustering for ontology building is hard to recognize the meaning of each cluster and label it relating to ontology domain. In the same time, the quality of clusters is not satisfying. In our work, our approach is based on a core ontology and aims at obtaining clusters, where each one includes terms that are synonyms, hypernyms of a core concept or strongly related to it. Meanwhile, little effort has been done in term clustering for ontology learning using core ontology.

4 Frameworks Comparison Approach

For the purpose of ontology building, we established a workflow for the comparison of approaches of term. The workflow is comprised of 5 stages to gradually transform corpus into the dedicated clusters of terms. The corpus (stage 1) provides resources for relation extraction of terms (stage 2). It brings two basic feature representations, co-occurrence representation and word embedding representation. With respect to feature transformation and dimension reduction techniques, the two initial features could be transformed into 4 extensive feature representations (stage 3). Based on those representations of terms, various clustering algorithms are employed to gather together the semantic similar terms (stage 4). Finally, the quality of clusters is assessed according to evaluation indices (stage 5).

4.1 Corpus Selection

With the aim of term clustering experiments, we choose two corpora in different domains: music domain and ontology learning domain. Each corpus possess the gold standard, which includes a set of extracted terms that are classified manually over the core concepts of the domain.

Music Corpus [6], is composed of 100M-word documents, includes Amazon reviews, music biographies and Wikipedia pages about theory and music genres. We deliberately selected 2000 documents from 105,000 documents, ensuring that the chosen

Fig. 4. The term clustering workflow. Adapted from Xu et al. [52]

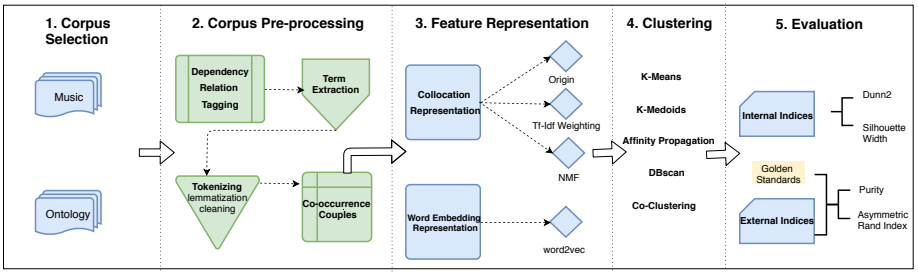


Table 1. The Corpus Size and Statistics.

Corpus	# Docs	Sampling	# Sentences	# occurrence	# unique tokens	$\frac{\#tokens}{docs}$	# docs containing CC	$\frac{\#CCs}{docs}$
Music	105,000	2,000	28,286	703,519	51,327	351	1,879	4.9
Ontology	16	16	4,901	112,628	7,700	7,040	16	198.7

content includes the great proportion of terms in the predefined gold standards. The Ontology Learning Corpus comprises of 16 scientific articles in the domain of ontology learning. As shown in table 1, it presents the statistics of the number of documents, the number of documents after sampling, the number of sentences, the occurrence of tokens, the number of unique tokens, the number of tokens divided by the number of sampled documents, the number of documents containing a core concept(CC) and the number of core concepts(CCs) divided by the number of sampled documents. These two corpora are different in terms of domain and the amounts of docs, however, their evident contrast could help researchers to figure out whether it exists a relation between corpus and term clustering techniques.

The aforementioned core concepts are predefined for each domain in the gold standard. As shown in table 2, the gold standard of Music Corpus is composed of 4,382 relevant terms (i.e nouns relevant for the music domain) labeled with one of the core concepts of music domain, while in the gold standard of Ontology Learning Corpus, 2953 terms (as nouns) are labeled with one of the core concepts of the ontology learning domain.

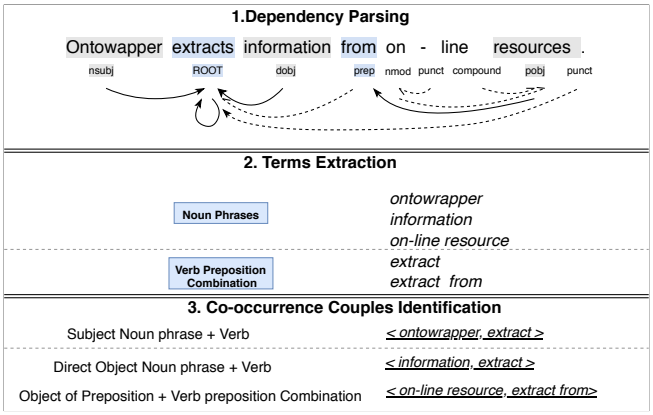
4.2 Corpus Pre-processing

Considering that only semantic similar terms are interesting to be clustered, it turns to be essential to extract the relations between terms from their context. Regarding to the utility of syntactic roles, the skeleton of a sentence is supposed to comprise the subject, the object and their related verb. In other words, terms with important syntactic roles are assumed to cover the most descriptive information in a sentence. Thus noun phrases (NP), acting as subject or object, are worth to be highlighted in concept extraction, while

Table 2. The Gold Standard.

Corpus	# core concepts	# terms	Labels of Core Concepts
Music	5	4,382	Album, Musician, Music Genre, Instruments, Performance
Ontology Learning	8	2,953	Component, Technique, Ontology, Domain, Tool, User, Step, Resource

Fig. 5. The instantiated co-occurrence couples extraction. Adapted from Xu et al. [52]

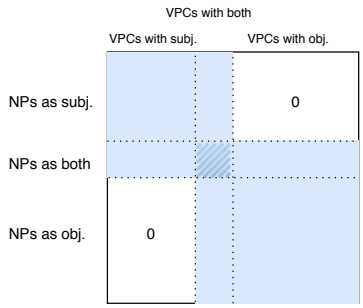


their contextual components, i.e. verbs, could present the concrete connection between NPs.

In the procedure of relation tuples extraction based on dependency tagging, as shown in the stage 2 of Fig. 4, syntactic information is extracted to help identify NPs acting as a subject or object and their co-occurred verbs. In our experiment, we propose to use spaCy [48] as a parser tool. It could decompose an entire typical syntactic tree into structured information, which shows the overwhelming convenience in post-processing, comparing to other parser tools, such as cleanNLP [2] and coreNLP [35].

To explain how noun phrases (NPs) with subject and object role and verb-preposition combinations (VPCs) are extracted during the POS tagging, we provide an instance about co-occurrence couples extraction in Fig. 5. After the tokenization of a sentence, tokens will be cleaned and lemmatized. Following the pre-processing steps, we start with the recognition of skeleton terms. As shown in the top of Fig. 5, terms in a sentence are presented with dependency relation, where the shaded terms have been tagged as subject (nsubj), ROOT and object (dobj, pobj). The subject (ontowrapper) and direct object (information) point to the ROOT (extract) with the solid lines, while the proposition object (on-line resource) indirectly points to ROOT (extract from) with the relay of dashed lines and solid lines. As for the non-skeleton dependency, they are connected in dashed lines. Furthermore, we need to pay attention to the distinction between the passive and active sentences. To simplify the composition of sentences, it is practical to

Fig. 6. The merged co-occurrenc matrix. Adapted from Xu et al. [52]



record passive subject (nsubjpass) as direct object (dobj). With the help of head pointers, noun phrases (NPs) and verb-preposition combinations (VPCs) could be gathered and extracted in the compound format. Finally, the pairs of ROOT (verbs) and skeleton terms are tagged and recorded as the reconstruction resource taking the place of the raw corpus.

4.3 Feature Representation

After the pre-processing, the feature representation stage are following as shown in Fig. 4. We plant to experiment with 5 distinct strategies to build the word representation in a scalar vector space where each word is encoded as a numeric tuple/vector. We begins with two disparate approaches to build the basic feature representations. One of the fundamental vector spaces takes advantage of the frequency of NPs-VPCs pairs, while another feature representation uses the entire context to acquire the word embedding. They differ from each other in the range of terms co-location, for which the fundamental method facilitates syntactic roles for co-occurrence pairs within a sentence, while the word embedding method takes into account a certain length of context of all appearance places of a term. Additionally, to tackle the sparseness problem of numeric vectors, dimensionality reduction techniques are employed to condense feature representation.

Co-occurrence Representation To build up the co-occurrence representation, the aforementioned pairs are extracted and transformed into a co-occurrence frequency matrix, where VPCs are considered as the features of NPs. Since we notice that it exists a big gap in terms of the syntactic functionality between subject and object, their representation are supposed to be separated into different co-occurrence pairs, named subject co-occurrence and object co-occurrence.

As a ground truth, one kind of co-occurrence pairs, either subject or object, could only convey the partial linguistic knowledge from a sentence. It is profitable to deliberately combine subject and object co-occurrence pairs, with the intention of an entire coverage of context. Thus, we propose the merged co-occurrence matrix (in Fig. 6). In this model, we differentiate NPs and VPCs into pure subject, pure object and common

part. The common part means NPs and VPCs appear in both subject and object. On the whole, the merged matrix comprises 9 sub-parts, where the non-existing pairs present to be all zero (blank rectangles) and the pure pairs (subject or object) present their frequency respectively in two blue rectangles. Common couples (shaded rectangles), the overlaps between subject rectangle and object rectangle, are filled with the accumulative frequency of subject pairs and object pairs. From any objective perspective, as long as subject and object co-occurrence pairs join together, the merged matrix theoretically encompasses complete linguistic information. Hence, this merged model will work as a primary representation in the following part.

Table 3. The dimensionality reduction after the threshold. Adapted from Xu et al. [52]

	#NPs			#VPCs			Reduction with Frequency		Reduction with Tf-Idf	
							#NPs	#VPCs	#NPs	#VPCs
corpus	subj.	obj.	both	subj.	obj.	both	Threshold δ_1:Summation of frequency		Threshold δ_2:Summation of value	
music	3,138	7,272	1,560	254	3,054	532	$\delta_1 > 8$		$\delta_2 > 7$	
							573	660	582	456
Ontology	401	1,643	281	80	889	219	$\delta_1 > 3$		$\delta_2 > 4$	
							602	505	563	502

Dimensionality Reduction The sparseness of a merged co-occurrence matrix becomes a significant issue, where the dimension reduction technique can be applied to solve it. For a sparse matrix, the reduction over row and column are both required to decrease the noise effect. In table 3, we apply with the frequency-based thresholds to eliminate the most common and rare NPs and VPCs. On the right hand, Tf-Idf encoding representation also provides bi-directional selection respecting to NPs and their tf-df features.

- **Weighted Co-occurrence Representation.** Based on the co-occurrence representation, we would like to weight values to differentiate the importance of co-occurrence pairs. Tf-Idf, is designed with this discriminative purpose. Basically, this algorithm could extract the most descriptive terms from documents, which is able to be extended to weight the most significant NPs to specific VPCs, instead of documents. With certain thresholds in rows and columns, only prominent NPs and their co-occurred VPCs are kept at last. Owing to the derivation of Tf-Idf, the close connected NPs and VPCs are preserved through the thresholds in table 3 so that the weighted co-occurrence matrix gets refined from the reduced dimensionality.
- **NMF Co-occurrence Representation.** Term co-occurrences could be separated into 3 levels according to the identity of words in context [20]. In the first-order co-occurrence, terms appear together in the identical context. As for two terms are associated by means of second-order co-occurrence, they share at least one-word context and have strong syntactic relations. Besides, terms do not co-occur in context with the same words but between words that can be related through indirect co-occurrences, namely third (higher) order co-occurrence. To capture those co-occurrences, NMF [34] is applied to condense the isolated VPCs into some encoded

features. In this way, NPs with indirect co-occurrence are presented in the new dense feature space. We set the number of features to be 100 during experiments.

Word Embedding Representation On the basis of contextual information, it allows to build feature vectors that are adapted for semantic similarity tasks. Word embedding representation was trained using word2vec [36] algorithm under the skip-gram model. In the local aspect, terms can be represented by vectors of its co-located words within certain window size, called co-locating vectors. The sum of co-locating vectors around the appearance place of a term constitutes the context vectors. In the global aspect, the sum of context vectors at all appearance places of a term gives the construction of word vectors. It integrates all the contextual features of a word and presents by the encoded similarity statistically. One of the advantages of word2vec is that it achieves dimension reduction purposes by indicating the required amount of features. To be comparable with NMF encoding technique, the number of features with word2vec is also given by 100.

4.4 The Clustering over Feature Representation

Heretofore, we have introduced all the alternative algorithms in the term clustering workflow, involving four different feature representations and five diverse clustering approaches. The composition of these alternatives is under interests for term clustering exploration, this effort assists to present a clear comprehension for the dominant possibilities of term clustering workflows.

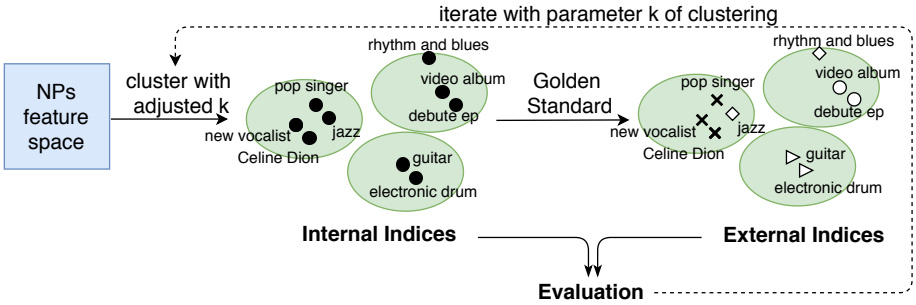
In this stage, we analysis the combination of the different feature representation and clustering algorithms. The four feature representations have various concerns about relations between terms. As we discussed in previous section, the *co-occurrence representation* offers the co-occurrence relation between NPs and VPCs; the *weighted co-occurrence representation* discriminates the principle co-occurrence from the rare or extreme frequent pairs of NPs and VPCs; the *NMF co-occurrence representation* takes into account the indirect co-occurrence of pairs by encoded features; the *word embedding representation* emphasizes the co-occurrence within certain windows in sentences. These distinct features would generate different compactness with these five clustering algorithms, including *k-means*, *k-medoids*, *affinity propagation*, *DBscan* and *co-clustering*.

4.5 Evaluation Indices

A large number of indices provide possibilities to assess the clustering quality [1]. In order to simplify the discrimination process, we select two distinct indices respectively for internal evaluation and external evaluation.

Indices for Internal Evaluation To evaluate the observations that are aggregated into clusters, one intuitive approach is to measure their compactness and separateness by their geometric similarity. Without any assistance of extra knowledge, the cluster could

Fig. 7. The utility of gold standard. Adapted from Xu et al. [52]



Note: The symbol 'black dots' in green circle represent the unlabeled terms, on the right hand, the 'cross', 'diamond', 'circle' and 'triangle' symbols in green circles show that the terms are marked with different labels.

be evaluated with some distance-based indices, given the name as the internal evaluation. In Fig. 7, after applying the clustering algorithm over one of the NPs feature spaces, the terms could be directly evaluated by internal indices. However, in this situation, the clusters are difficult to be labeled with some human understandable concepts.

Silhouette width [44] and adjusted Dunn Index are chosen as indices of internal evaluation. Silhouette method specifies how well each observation lies within its cluster.

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{1}$$

In equation 1, i represents one observation in clusters, $a(i)$ represents average dissimilarity between i and all other observations of the cluster to which i belongs. For each cluster C, $d(i,C)$ denotes average dissimilarity of i to all observations of C. In this basis, $b(i)$ is set by the smallest $d(i,C)$ and can be considered as the dissimilarity between an observation i and its neighbor cluster. A high average silhouette width indicates a good clustering according to features.

Adjusted Dunn Index proposed by Pal and Biswas [41] overcomes the presence of noise comparing to original Dunn Index [12]. In general, they are both dedicated to the identification of compact and well-separated clusters. Higher values are preferred, which shows a good performance of compactness. Notably, the Dunn Index family does not exhibit any trend with respect to the number of clusters, of which this property is exceedingly welcomed since the number of clusters varies in different iterations.

Indices for External Evaluation In the case of external evaluation, the indices are slightly different from the former because of the use of a gold standard. In the external indices section of Fig. 7, the terms are marked with different labels by the classes of gold standard, which becomes human interpretable. For instance, in the displaying of clusters, the left cluster includes terms as 'pop singer', 'new vocalist', 'Celine Dion' and 'jazz'. With the assistance of labels, it is straightforward to explain that this cluster

is composed of 75% musician class and 25% music genre class. The top-right cluster is constituted by 33% music genre class and 66% album class. The bottom-right cluster is labeled with 100% instruments class. Further, in the approach of external evaluation, the clusters are capable to be label by classes from external evaluation, which provides the possibility for cluster labelling issues. According to the expected core concept classes, Purity and Asymmetric Rand Index are representative of clustering quality measurement.

Purity is one of the most simple and widely used indices. Each cluster firstly is assigned with a label that is most frequent, according to the gold standard, then this assignment is calculated by counting the number of correctly assigned elements dividing by all elements. High purity is easy to achieve when the number of clusters is large, because the number of terms in each cluster will significantly decease and the percentage of terms with the same label probably increases. A larger amount of clusters may refine the branches of structure in ontology building, however, it incurs complexity to label clusters with core concepts, performing as the first step of ontology learning. Thus we could not use only purity to trade off the quality of clustering against the number of clusters.

The Asymmetric Rand Index proposed by Hubert and Arabie [30] is also considered, for which it provides the comparison between the result of a classification and a correct classification. This index is developed from the idea of typical Rand Index (RI). Instead of counting single observation, the typical Rand Index (RI) counts the correctly classified pairs of observations. Then the rand index [42] is calculated by:

$$RI = \frac{a+b}{\binom{n}{2}} \tag{2}$$

, where $\binom{n}{2}$ is the number of un-ordered pairs in a sets of n observations. The a in the formula refers to the number of times that the pair of observation belongs to the same classification but exists in different clusters and the b indicates the opposite way, in which a pair belongs to different classification and exists in different clusters. Hence RI depends on both, the number of clusters and the number of observations [49].

However, we cannot get the lowest value (e.g. zero) for two random partitions by typical Rand Index. Thus Hubert and Arabie [30] made a modification with the null hypothesis, which means the value of Adjusted Rand Index (ARI) is expected to under the null hypothesis, with 0 for independent clustering and 1 for identical clustering. The Adjusted Rand Index (ARI) [30] is defined as follows:

$$ARI = \frac{\sum_{i=1}^{k}\sum_{j=1}^{l}\binom{m_{i,j}}{2} - t_3}{\frac{1}{2}(t_1+t_2) - t_3} \tag{3}$$

, where $t_1 = \sum_{i=1}^{k}\binom{|C_i|}{2}$, $t_2 = \sum_{j=1}^{l}\binom{|C'_j|}{2}$, $t_3 = \frac{2t_1t_2}{n(n-1)}$. In general, the i and j represents the cluster i and classification j. The m_{ij} indicates the number of observations in cluster i matching to classification j. The $|C_i|$ and $|C'_j|$ represent the total number of observations for each cluster i or for each classification j, respectively.

Additionally, ARI allows that the number of clustering can be different with the number of classification of gold standard. During experiments, the number of partitions

Table 4. The Parameters of 5 clustering algorithm.

clustering	similarity measure	R library	function	k selection	parameter selection	other parameters
k-means	cosine	stats	kmeans()	2-50	-	-
k-medoids	cosine	cluster	pam()	2-50	-	medoids
affinity propagation	cosine	apcluster	apclusterK()	2-50	-	maxits=2000;convits=200
DBscan	cosine	fpc	dbscan()	-	dbscan::kNNdistplot()	eps=0.2;minpts=3
co-clustering	-	blockcluster	coclusterContingency()	9	-	-

are always larger than that of classification from gold standard, while the application of Asymmetric Rand Index allows for a more accurate analysis.

5 Experiment Settings and Evaluation

However, before the examination of the combination between clustering algorithms and feature representations, it is inevitable to preset the parameters regarding clustering methods. How to choose the optimal number of clusters? And is it valuable to choose the number of clusters according to core concepts? These puzzles would be tackled in the following subsections. On the grounds of these prepared settings, the related experiments are executed to provide the evaluation for each examination. The analysis of those outcomes will bring in the recommendation about the alternative algorithms for term clustering.

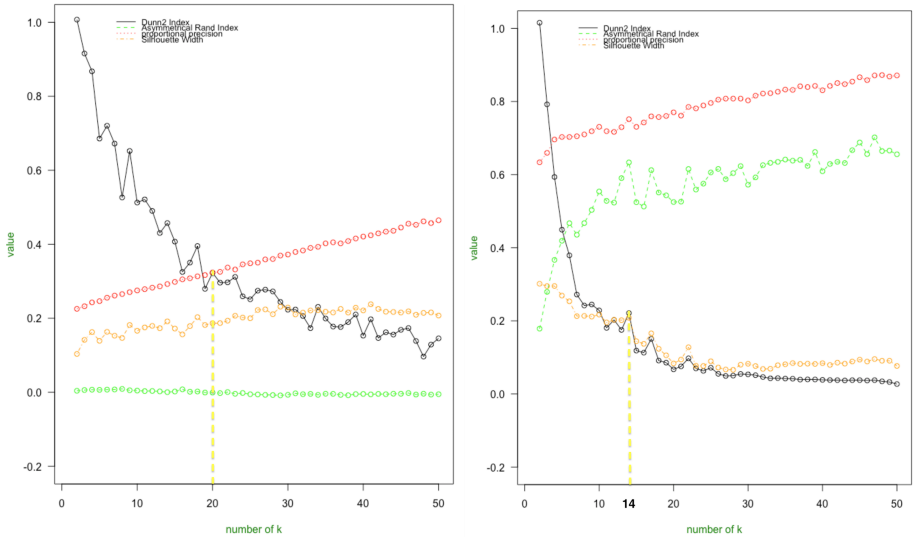
5.1 Parameter Setting of Clustering

On the basis of various feature representation of terms, the inner relations between terms are able to be discovered by the clustering algorithms. The clustering algorithms differ from each other with regards to both their distance measurements and their preference on the optimal number of clusters.

Similarity Calculation of Clustering Before applying these representations to clustering algorithms, it is essential to illustrate the choice of similarity/dissimilarity measure for each algorithm. For example, k-means, k-medoids and DBscan clustering algorithm make use of cosine distance for each representation. The cosine measurement has an outstanding favorable property as normalization, which fits well to the multi-nominal probability distributions in Bag-of-word assumption. On the contrary, the affinity propagation employs cosine similarity calculation as required by the executing algorithms. However, the similarity or dissimilarity calculation is skipped for the co-clustering approach, because its concentration is to explore the contingency of raw data with row and column effects.

Repetitions of Clustering To weaken the impact of the randomness of clustering, the repetition of experiments is necessary as a proof for the subsequent analysis. Generally, to serve our purpose about selecting the optimal number of clusters, each experiment goes through all the parameters of k (the number of clusters) ranging from 2 to 50

Fig. 8. The examples of parameter selection with K-Means



(a) Ontology Learning corpus (b) Music corpus

Note: All values are statistically averaged from 10 times

for 10 repetitions. To get the convincing results, each index is statistically averaged to mean values for evaluation. As presented in Fig. 7, each iteration allows for analysis of clustering performance respecting to internal indices and external indices.

However, this method is not suitable for all of the clustering methods. On the one hand, as we mentioned that some clustering algorithms do not require the pre-setting of the number of clusters, instead, they are able to provide the choice of optimal number of cluster. As shown in table 4, it depicts the experiment parameters for these five clustering algorithms. Theoretically, affinity propagation can be implemented without such prerequisite. However, from the experiments we obtained very poor performance based on the automatic assignment of k. In order to acquire the optimal setting for such clustering algorithm, we apply the k selection procedure to vary the number of clusters from 2 to 50. As for DBscan clustering algorithm, it could implement without such prerequisite of k, but it needs the parameter of the minimum number of points required to form a dense region (i.e. minPts) and the radius of a neighborhood with respect to some points (i.e. eps). Fortunately, it exist a function (parameter selection column of table 4)to assist us to find a suitable value for DBscan by calculating the k-nearest neighbor distances in a matrix of points. In the aspect of co-clustering, the selection of these two parameters brings many complexity. To solve this problem, the expert knowledge of domain assists us to settle down these numbers. Hence, we can directly use the optimal number of clusters the same as the number of core concepts in the different corpus.

Eventually, k-means, k-medoids and affinity propagation clustering are capable to find their optimal number of clusters from a large range of candidates. To select the op-

timal amount of clusters, we attempt to solve the multi-criteria optimization problem. As Fig. 8 shows, it represents the evaluation results of k-means clustering with the co-occurrence feature representations, for Ontology Learning Corpus and Music Corpus separately. The two plots depict the fluctuation of all evaluation indices along with the increasing of the number of clusters. In order to address the multi-criteria optimization problem, we plan to find some evident peaks of one of the most fluctuating line and choose one from these candidate peaks to assure a rather higher summation over the entire indices. For instance, in the left sub-figure, we select the first 10 peaks of Dunn Index as candidates, and calculate the summation of all indices for those 10 candidates. Then we can choose the candidate with the highest summation as the optimal number of cluster. In Fig. 8, the dashed lines indicate the final parameter choice for this specific representation. In this figure, we select 20 as the optimal k of left experiment and select 14 for right experiment. Besides, the selection procedures of the rest feature representations follow the same rules.

In other words, it seems better to choose a locally optimal k around the number of core concepts, so as to restrict the number of clusters within a suitable range for ontology learning purpose. This assumption takes into consideration of number of core concepts. However, it rejects the possibilities of high-quality clustering along with smaller clusters. Therefore, in replace of the local optimization approach, global optimization of all indices is preferred to choose parameters of k-means and affinity propagation clustering for each feature representation.

5.2 Evaluation of Clustering

Obviously, to complete the experiments, we need to apply the 5 different clustering methods upon the 4 diverse feature representations. Thus around 20 experiment outcomes are presented for each corpus. On the basis of these statistics, we have made multiple comparisons to discover the valuable matching from corpus to clustering algorithms and to feature representations for term clustering. The table 5 and table 6 indicate the evaluation of 5 clustering methods and 4 feature representations. The *co-occurrence representations* are denoted as 'NP-VPC', while their extended embedding techniques *weighted co-occurrence representation* and *NMF representation* are denoted by 'NP-VPC-tfidf' and 'NP-VPC-NMF'. To be short, these 3 representations are called by a joint name 'NP-VPC representation family'. Besides, the *word embedding representations* is said to 'NP-w2v'.

Generally, in the aspect of the corpus, it is evident that Music Corpus (see table 6) reaches a much higher purity and higher Asymmetric Rand Index than that of Ontology Learning Corpus (see table 5). It can be due to that bigger corpus (Music Corpus) provides significant contextual features to cluster terms.

For the difference between 5 clustering methods, first of all, there is no overwhelming clustering approach according to those evaluation indices. The performance of k-medoids is comparable or better than that of k-means, which conforms to our intuition somehow. The k-medoids clustering methods need the knowledge of centroids before calculation, whose results are expected to be more accurate than that of k-means. Moreover, co-clustering has a rather poor performance than others. During experiments, some feature representations are even failed with this algorithm, due to the contingency

Table 5. The evaluation of 5 clustering methods and 4 feature representations (Ontology Learning Corpus).

corpus	clustering	feature representation	# selected k	Purity	Asymm Rand Index	Dunn2 Index	Silhouette Width
Ontology	K-Means	NP-VPC	20	32.3%	-0.1%	32.3%	18.6%
		NP-VPC-tfidf	44	60.3%	-2.6%	22.2%	-2.0%
		NP-VPC-NMF	37	43.3%	-1.0%	24.7%	47.0%
		NP-w2v	37	62.1%	**51.2%**	64.7%	13.9%
	K-Medoids	NP-VPC	25	36.2%	-0.5%	70.2%	24.3%
		NP-VPC-tfidf	18	36.2%	-0.5%	70.2%	24.3%
		NP-VPC-NMF	21	38.7%	1.0%	**99.4%**	25.7%
		NP-w2v	17	51.8%	9.7%	85.5%	7.8%
	Affinity Propagation	NP-VPC	47	48.2%	-0.3%	76.8%	34.7%
		NP-VPC-tfidf	-	-	-	-	-
		NP-VPC-NMF	26	41.0%	3.1%	10.5%	**50.0%**
		NP-w2v	43	**62.2%**	7.2%	87.1%	13.0%
	DBscan	NP-VPC	7	27.2%	-0.6%	-	1.6%
		NP-VPC-tfidf	1	25.6%	-1%	-	17.6%
		NP-VPC-NMF	76	56.6%	3.5%	79.4%	45.5%
		NP-w2v	16	35.8%	0.7%	73.3%	-12.2%
	Co-clustering	NP-VPC	9	26.6%	0.8%	49.0%	-10.7%
		NP-VPC-tfidf	-	-	-	-	-
		NP-VPC-NMF	9	26.6%	-0.4%	**99.4%**	4.2%
		NP-w2v	-	-	-	-	-

Note: All values are statistically averaged

requirement of input data. While the affinity propagation algorithm achieves the relatively best performance of clustering, and the DBscan clustering methods are slight deficient than that of k-means, k-medoids and affinity propagation.

As for the evaluation indices, we notice that it occurs negative values in asymmetric Rand Index and silhouette width. In this situation, the former index reflects a worse elements labeling of clusters comparing to Gold Standard, while the latter index shows that there are overlapping parts between different partitions, which means feature similar terms probably share different labels. That is inevitable in linguistic because the similar context of terms could not straightly infer to the same meaning of them.

In terms of the encoding representations, Tf-Idf representations provide unevenly lower accuracy in clusters. While the NMF representations and the word embedding representations have a good clustering quality overall. On the other hand, the performance of co-occurrence representations varies along with different corpus. From the results of evaluation, we observe a rather better performance in Music corpus than that in Ontology Learning corpus. Due to that the bigger corpus(the Music corpus) contains more frequent NP-VPC pairs, the co-occurrence matrix can present more distinguishing values to accentuate their features for term clustering purpose.

In the aspect of the combination of clustering and feature representations, it is preferable to list the most outstanding feature embedding technique for each clustering method. In table 5 and table 6, we select the required feature representations for each clustering approach only if the amount of the underlined indices for that feature is as many as possible (the underlines are used to mark the highest value for each clustering method). According to the bold values in table 5 and table 6, we are able to choose the best combination for each corpus. The selected results are presented in table 7.

Table 6. The evaluation of 5 clustering methods and 4 feature representations (Music Corpus).

corpus	clustering	feature representation	# selected k	Purity	Asymm Rand Index	Dunn2 Index	Silhouette Width
Music	K-Means	NP-VPC	14	75.1%	63.3%	22.1%	20.8%
		NP-VPC-tfidf	10	48.2%	0.1%	28.0%	13.3%
		NP-VPC-NMF	14	70.4%	24.4%	64.6%	19.0%
		NP-w2v	13	81.1%	59.9%	79.7%	16.1%
	K-Medoids	NP-VPC	23	78.9%	67.9%	72.3%	16.0%
		NP-VPC-tfidf	17	53.1%	5.2%	-	22.2%
		NP-VPC-NMF	25	75.1%	**74.5%**	**98.7%**	16.7%
		NP-w2v	27	80.9%	59.5%	87.1%	5.8%
	Affinity Propagation	NP-VPC	33	85.3%	74.3%	86.0%	**39.3%**
		NP-VPC-tfidf	37	69.9%	8.9%	11.1%	30.9%
		NP-VPC-NMF	19	75.2%	59.1%	98.2%	26.9%
		NP-w2v	37	**89.6%**	73.8%	91.4%	23.3%
	DBscan	NP-VPC	14	65.3%	19.1%	83.8%	2.6%
		NP-VPC-tfidf	12	48.2%	2.2%	-	20.7%
		NP-VPC-NMF	60	78.2%	22.1%	78.2%	14.2%
		NP-w2v	26	56.6%	7.0%	75.1%	-12.0%
	Co-clustering	NP-VPC	9	49.5%	-3.4%	59.9%	-8.7%
		NP-VPC-tfidf	-	-	-	-	-
		NP-VPC-NMF	9	48.5%	-0.5%	96.7%	1.7%
		NP-w2v	-	-	-	-	-

Note: All values are statistically averaged

Table 7. The resulted combination of clustering and feature representation

clustering	feature representations	
	Ontology Learning Corpus	Music Corpus
K-Means	NP-w2v	NP-w2v
K-Medoids	NP-w2v	NP-VPC-NMF
Affinity Propagation	NP-w2v	NP-VPC
DBscan	NP-VPC-NMF	NP-VPC-NMF
Co-clustering	NP-VPC-NMF	NP-VPC-NMF

From these voted combinations, we aware that the majority of feature representation lies in NMF embedding technique. Except for k-means and affinity propagation, the other clustering methods are prone to fit well with NMF in at least one corpus. From table 7, we notice several outperforming combinations of clustering and feature representation, including k-means with word embedding representation and DBscan or co-clustering with NMF embedding technique. However, for the counterpart of k-medoids and affinity propagation clustering algorithm, it does not exist a dominant feature representation for the different corpus, however the word embedding representation can achieve a rather good performance in small size corpus.

In general, NMF embedding technique and word embedding representations are prominent in most clustering situations. The word embedding representations show an enhanced quality of clustering with K-Means. The DBscan clustering algorithm accompanying with NMF encoding technique achieves a rather good performance in both corpus. On the other hand, the co-occurrence representations reach comparatively good

performance with affinity propagation clustering, which shows the affinity propagation's feasibility over co-occurrence pairs.

6 Conclusions

Many works suggest making use of core ontology to build modular ontology. However, most of these efforts are manually constructed and seldom in the automatic approach. Term clustering according to a core ontology supports modular ontology construction without artificial demands. Taxonomic relations are constructed by gathering of NPs appearing with prominent syntactic roles after VPCs respecting to core concepts. Successfully we constructed feature space with these characteristics from two specialized corpora. To tackle the problem of sparsity, we benefit from feature selection and feature extraction techniques, such as adjusted Tf-Idf algorithm and NMF technique. Apart from that, word2vec is also compared as a benchmark. Along with all the extended representations, terms are clustered by 5 different clustering algorithms, which contains k-means, k-medoids, affinity propagation, DBscan and co-clustering algorithm. We found that the original co-occurrence feature space appearing with syntactic roles is not the most outstanding feature representation, while the usage of Affinity propagation clustering based on this original representation could prominently improve clustering performance. It is proved that the word embedding representations show an enhanced quality with K-Means and NMF encoding technique achieves a rather good performance with DBscan clustering algorithm. While the k-medoids and affinity propagation clustering algorithm have their preference to feature representations depending on the size of corpus.

From the comparison of term clustering frameworks, we recommend to start with a bigger domain-specific corpora. The syntactic relations between noun phrases and verbs are sufficient as features representation, with the assistance of encoding techniques, it gives rather convincing results in term clustering, which provides us a guideline for modular ontology building.

In the future work, we would like to explore the relations between terms in each module of ontology, so as to construct the concept hierarchy in modules. On the other hand, the relation between modules is still under our interests, in order to form a complete domain modular ontology.

Bibliography

[1] Aggarwal, C.C., Zhai, C.: A survey of text clustering algorithms. In: Mining text data, pp. 77–128. Springer (2012)

[2] Arnold, T.: A tidy data model for natural language processing using cleannlp. The R Journal **9**(2), 1–20 (2017), https://journal.r-project.org/archive/2017/RJ-2017-035/index.html

[3] Buitelaar, P., Cimiano, P., Magnini, B.: Ontology learning from text: An overview. Ontology learning from text: Methods, evaluation and applications **123**, 3–12 (2005)

[4] Buitelaar, P., Olejnik, D., Sintek, M.: A protégé plug-in for ontology extraction from text based on linguistic analysis. In: European Semantic Web Symposium. pp. 31–44. Springer (2004)

[5] Burita, L., Gardavsky, P., Vejlupek, T.: K-gate ontology driven knowledge based system for decision support. Journal of Systems Integration **3**(1), 19–31 (2012)

[6] Camacho-Collados, J., Delli Bovi, C., Espinosa-Anke, L., Oramas, S., Pasini, T., Santus, E., Shwartz, V., Navigli, R., Saggion, H.: SemEval-2018 Task 9: Hypernym Discovery. In: Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018). Association for Computational Linguistics, New Orleans, LA, United States (2018)

[7] Chulyadyo, R., Harzallah, M., Berio, G.: Core ontology based approach for treating the flatness of automatically built ontology. In: KEOD. p. 316:323. Portugal (Sep 2013)

[8] Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Computational linguistics **16**(1), 22–29 (1990)

[9] Cimiano P., de Mantaras, R.L., Saitia, L.: Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text. In: 16th european conference on artificial intelligence conference proceedings. vol. 110, p. 435 (2004)

[10] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological) **39**(1), 1–22 (1977)

[11] Despress, S., Szulman, S.: Merging of legal micro-ontologies from europen directives. Artif. Intell. Law **15**(2), 187–200 (Jun 2007)

[12] Dunn, J.C.: Well-separated clusters and optimal fuzzy partitions. Journal of cybernetics **4**(1), 95–104 (1974)

[13] El Ghosh, M., Naja, H., Abdulrab, H., Khalil, M.: Application of ontology modularization for building a criminal domain ontology. In: AI Approaches to the Complexity of Legal Systems, pp. 394–409. Springer (2015)

[14] Esposito, F., Fanizzi, N., dAmato, C.: Partitional conceptual clustering of web resources annotated with ontology languages. In: Knowledge Discovery Enhanced with Semantic and Social Information, pp. 53–70. Springer (2009)

[15] Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD. vol. 96, pp. 226–231 (1996)

[16] Faure, D., Ndellec, C., Rouveirol, C.: Acquisition of semantic knowledge using machine learning methods: The system ”asium”. Tech. rep., Universite Paris Sud (1998)
[17] Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: From ontological art towards ontological engineering. In: AAAI 1997 (1997)
[18] Frey, B.J., Dueck, D.: Clustering by passing messages between data points. science **315**(5814), 972–976 (2007)
[19] Gábor, K., Zargayouna, H., Tellier, I., Buscaldi, D., Charnois, T.: Unsupervised relation extraction in specialized corpora using sequence mining. In: International Symposium on Intelligent Data Analysis. pp. 237–248. Springer (2016)
[20] Gamallo, P., Bordag, S.: Is singular value decomposition useful for word similarity extraction? Language resources and evaluation **45**(2), 95–119 (2011)
[21] Gangemi, A., Catenacci, C., Battaglia, M.: Inflammation ontology design pattern: an exercise in building a core biomedical ontology with descriptions and situations. Studies in health technology and informatics **102**, 64–80 (2004)
[22] Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Modelling ontology evaluation and validation. In: European Semantic Web Conference. pp. 140–154. Springer (2006)
[23] Govaert, G., Nadif, M.: Latent block model for contingency table. Communications in Statistics - Theory and Methods **39**(3), 416–425 (2010)
[24] Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: A logical framework for modularity of ontologies. In: IJCAI. vol. 2007, pp. 298–303 (2007)
[25] Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge acquisition **5**(2), 199–220 (1993)
[26] Hao, J., Zhang, C., Wang, H.: Using keywords clustering to construct ontological hierarchies. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 03. pp. 247–250. IEEE Computer Society (2009)
[27] Harris, Z.: Distributional structure.(j. katz, ed.) word journal of the international linguistic association, 10 (23), 146-162 (1954)
[28] Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics) **28**(1), 100–108 (1979)
[29] Hois, J., Bhatt, M., Kutz, O.: Modular ontologies for architectural design. In: FOMI. pp. 66–77 (2009)
[30] Hubert, L., Arabie, P.: Comparing partitions. Journal of classification **2**(1), 193–218 (1985)
[31] Jiang, X., Tan, A.H.: Mining ontological knowledge from domain-specific text documents. In: Fifth IEEE International Conference on Data Mining. pp. 665–668. IEEE (2005)
[32] Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis, vol. 344. John Wiley & Sons (2009)
[33] Kutz, O., Hois, J.: Modularity in ontologies. Applied Ontology **7**, 109–112 (04 2012). https://doi.org/10.3233/AO-2012-0109
[34] Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788 (1999)

[35] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations. pp. 55–60 (2014)
[36] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
[37] Nancy, P., Ramani, R.G.: Discovery of patterns and evaluation of clustering algorithms in socialnetwork data (face book 100 universities) through data mining techniques and methods. International Journal of Data Mining & Knowledge Management Process **2**(5), 71 (2012)
[38] Oberle, D., Lamparter, S., Grimm, S., Vrandečić, D., Staab, S., Gangemi, A.: Towards ontologies for formalizing modularization and communication in large software systems. Appl. Ontol. **1**(2), 163–202 (Apr 2006)
[39] Opdahl, A., Berio, G., Harzallah, M., Matuleviius, R.: Ontology for enterprise and information systems modelling. Applied ontology **7**, 49–92 (12 2011)
[40] O Connor, L., Feizi, S.: Biclustering using message passing. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 3617–3625. Curran Associates, Inc. (2014), http://papers.nips.cc/paper/5603-biclustering-using-message-passing.pdf
[41] Pal, N.R., Biswas, J.: Cluster validation using graph theoretic concepts. Pattern Recognition **30**(6), 847–857 (1997)
[42] Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association **66**(336), 846–850 (1971)
[43] Rani, M., Dhar, A.K., Vyas, O.: Semi-automatic terminology ontology learning based on topic modeling. Engineering Applications of Artificial Intelligence **63**, 108–125 (2017)
[44] Rdrr.io: Silhouette: Compute or extract silhouette information from clustering. https://rdrr.io/cran/cluster/man/silhouette.html (2019), accessed: 2019-5-10
[45] Rios-Alvarado, A.B., Lopez-Arevalo, I., Sosa-Sosa, V.J.: Learning concept hierarchies from textual resources for ontologies construction. Expert Systems with Applications **40**(15), 5907–5915 (2013)
[46] Scherpa, A., Saathoffa, C., Franza, T., Staaba, S.: Designing core ontologies. Applied Ontology **3**, 1–3 (2009)
[47] Song, Q., Liu, J., Wang, X., Wang, J.: A novel automatic ontology construction method based on web data. In: 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. pp. 762–765. IEEE (2014)
[48] spaCy: Spacy:industrial-strength natural language processing (nlp) with python and cython, explosion ai. https://github.com/explosion/spaCy (2019), accessed: 2019-5-10
[49] Wagner, S., Wagner, D.: Comparing clusterings: an overview. Universität Karlsruhe, Fakultät für Informatik Karlsruhe (2007)
[50] Wang, W., Barnaghi, P.M., Bargiela, A.: Learning skos relations for terminological ontologies from text. In: Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances, pp. 129–152. IGI Global (2011)
[51] Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2016)

[52] XU, Z., Harzallah, M., Guillet, F.: Comparing of term clustering frameworks for modular ontology learning. pp. 128–135. Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 2: KEOD, SCITEPRESS - Science and Technology Publications, Seville, Spain (Sep 2018)