Core-Concept-Seeded LDA for Ontology Learning

Hao Huang, Mounira Harzallah, Fabrice Guillet, Ziwei Xu

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2019) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Core-Concept-Seeded LDA for Ontology Learning

Hao Huang*, Mounira Harzallah, Fabrice Guillet, Ziwei Xu

LS2N, University of Nantes, Rue Christian Pauc, Nantes, 44300, France

Abstract

Ontologies are powerful semantic models applied for various purposes such as improving system interoperability, information retrieval, question answering, etc. However, building domain ontologies remains a challenging task for humans, especially when the concepts and properties are large or evolving, and also when they are built from large-scale textual data. Machine learning allows to automate the building of ontologies from texts. In particular, clustering techniques have a promising ability on the concept formation task by identifying the cluster of semantically closed terms as a concept. However, current works encounter issues in learning relevant domain-specific clusters or in identifying the relevant concept labels for each cluster. To solve these issues, we propose both to use core concepts from a domain ontology as prior knowledge, and to adapt term clustering with seed knowledge-based LDA models in order to take these core concepts into account. First, each topic is associated with a set of seed terms of a single core concept, then the learning is guided by these seeds to gather in the same topic the terms that refer to its core concept. We evaluate our proposal on two textual corpora and compare it to the baselines (LDA, K-means, and SMBM). The results show that our approach performs significantly better than other methods on the class-balanced dataset and works well on the class-imbalanced dataset with a proper number of topics for each core concept.

Keywords: Ontology Learning; Core Ontology; LDA; Term Clustering; Seed Knowledge; Prior Knowledge; Semantic Coherence; Word2vec

1. Introduction

With the increasing availability of textual resources on the web, ontology learning from texts becomes a challenging issue. The traditional process for building an ontology consists of five main tasks: terms extraction, concepts formation, taxonomy extraction, ad-hoc relationships extraction, and axioms extraction [13]. Many machine learning approaches have been proposed for achieving these tasks, generally classified into two main categories: pattern-based approaches and distributional approaches [3]. For the first four tasks, pattern-based approaches show quite high precision whereas their recall is low because of the large variability in natural language for expressing a meaning [27].

* Corresponding author. Tel.: +86 17680325595.
E-mail address: hao.huang@etu.univ-nantes.fr

Distributional approaches, either supervised or unsupervised, are based on co-occurrence context distribution to detect semantic relationships between term pairs [32]. Supervised methods learn a model for predicting the class of terms or semantic relationships between term pairs. Unsupervised approaches are either measure-based approaches or clustering-based approaches. The former computes a score of the semantic closeness or the relatedness of two terms based on symmetric or inclusion measures and the latter gathers semantically closed terms on the same cluster. Generally, the supervised approaches outperform non-supervised approaches [33]. However, unlike unsupervised approaches, it requires additional effort to build a sufficiently large labeled training data set.

We are interested in *term clustering* methods for *concept formation*, and more specifically in clustering terms according to the *core concepts* (CCs) of a domain ontology. CCs are the minimal concepts that allow defining the other concepts of their domain [8]. In our works, we claim an ontology building approach guided by a core ontology (i.e. a model composed of CCs and core relationships between them). To be specific, according to the previous ontology building process, after terms extraction (step 1), we recommend first classify each term in a class associated with a CC (called in the remainder CC-class), then steps 2 and 3 are performed inside each class to form sub-concepts of CCs and hypernym relations between them. A CC-class is a class of terms that refer to this CC, i.e. its synonyms, hyponyms, or semantically closed terms. This approach allows to bound the look for synonym or hypernym relationships and then to reduce the computing time.

We focus on *Latent Dirichlet Allocation* (LDA) [6] for term clustering since it can deal with a huge number of documents and tackle the issue of text sparsity. LDA-based approaches provide a probability distribution of terms for each topic. From each topic, a cluster of the most relevant terms may be extracted according to the probability distribution (often the top-k most probable terms), then a cluster is labeled either by a domain expert or as terms with high probability. However, two key issues remain to be addressed. First, the semantic coherence of a cluster is not ensured, i.e the terms of a cluster do not refer necessarily to the same concept, thus the cluster is not entirely meaningful [12, 25]. Second, these approaches do not deal with the relevance of clusters for the targeted knowledge domain. Indeed, a cluster could be out of the target, i.e. a cluster of terms referring to an irrelevant concept for the ontology domain. To solve these issues, we propose the *Core-concept seeded LDA* that aims to perform automatically the task of classification of terms on CC-classes by resorting to clustering. The approach is an adaptation of seed based LDA models by choosing as seed terms those referring to CCs and labeling a priori each topic by a CC.

In this paper, we consider four seed based LDA models (Z-labels LDA[2], Seeded LDA M1, Seeded LDA M2, Seeded LDA[20]). In these models, a latent topic is about a subject that may be a large notion, and even if a topic is carried with high probability, it does not refer necessarily to a unique concept. For example, in [20], the topic of "earn" has as seed terms "company" and "quarter" that don't refer to the same concept. Further, the seed based LDA models are dedicated to document clustering. To the best of our knowledge, none of them has been used for term clustering for ontology learning. The paper is organized as follows. Section 2 presents term clustering approaches for ontology learning with a focus on LDA based approaches. Section 3 provides background knowledge of LDA and seeded based LDA models, with a focus on the principles of each. In section 4, we present our approach for adapting these models to term clustering over CCs. In section 5, we present several experiments conducted on two corpora. We compare the approach to unsupervised and semi-supervised baselines, we analyse the effectiveness of the proposed seed sets kind and we investigate the performance of the approach on imbalanced corpus before the conclusion.

2. Related Work

The main idea of distributional approaches is grouping terms by using the distribution of contexts where they appear, based on Harris' distributional hypothesis: terms that occur in similar contexts tend to have similar meanings[18]. Pereira et al.[28] originally proposed the distributional clustering of terms by using Kullback-Leibler (KL) distance to measure the term similarity. Caraballo [10] adapted a bottom-up clustering method for clustering noun terms and build up the hypernym relations among clusters. To improve the cluster coherence, Cimiano [14] recommended a guided agglomerative clustering algorithm for inducing concept hierarchies from the text corpus. Often researchers used as contexts the words occurring with target terms in a fixed-size window [5, 23]. Recently, word embedding techniques are introduced to exploit large contextual information. Several works [1, 4, 11] explored the possibility of using word2vec for concept formation. To do so, first, seed terms are predefined and used for representing the domain concepts. Then the domain corpus is used for learning term representation by word2vec. Finally, candidate terms of

each concept are selected by measuring the similarity between terms and seed terms of concepts. Those having high similarity with a seed term are chosen to enrich the concept.

Some works cast term clustering as a graph partition problem. The idea of graph-based representation of terms has its origin in [36]. A graph is defined as $G = (V, E, W)$, where V is a set of vertices representing terms, E is a set of edges between terms, and W represents the weights of edges. Based on this graph, Matsuo et al.[24] introduced a graph clustering algorithm (Newman clustering) for word clustering. To define the weight of an edge, different measures (point-wise mutual information, Jaccard coefficient, chi-square) are used based on the frequency of each word and that of its word pair. An edge between two words is eliminated if its weight is lower than a threshold. Lee and Luo [22] used the cosine similarity and word2vec for word representation to compute edge weights. Then, they proposed a term clustering method based on Louvain community detection algorithm [7], where each term community acts as a concept. Like the previous works, the author of [29] applied a community detection algorithm on terms graph for concept formation. The edge weights are computed in three ways: term co-occurrence based, word2vec based, and LDA based (i.e. the joint probability distribution of two words is calculated based on LDA theory). Thaiprayoon et al.[35] introduced a hierarchical agglomerative clustering to find similar words according to the criterion of distance range on a graph where each edge is weighted by the inverse of word pair frequency.

A recent study [26] shows that word embeddings are unable to predict some of the characteristics of human similarity judgment. Compared with word embeddings, LDA can capture relevant word associations. With a special focus on the relation between concepts and terms, Rani et al. [31] explored an LSI&SVD based method and a Mr.LDA (a map reduce LDA) based method for concept formation. Experimental results suggest the Mr.LDA based method is better than the LSI&SVD based method on several criteria. However, these methods need extra human effort to label each topic as a meaningful concept. In addition, in this work, they didn't consider the impact of irrelevant terms (terms of other domains) on topic building. To solve this problem, Xu et al. [37] proposed a twice-trained LDA where an LDA model is trained on a corpus the first time to identify irrelevant terms. Terms with a low probability on each topic are removed from the corpus. Then the LDA model is trained a second time with the cleaned corpus to build up topics. However, the author didn't exam the loss of domain-relevant terms. The approach in [16] also used different thresholds to pick up aggregate terms to represent the concepts formed from topics. For labeling topics, they used LDA to compute the conditional probability between term pairs and determine the aggregate roots as terms whose occurrence is not implied by the occurrence of other terms of the corpus. Finally, the aggregate root of each topic is used to label it. However, most approaches based on the LDA model didn't consider the semantic coherence of term clusters. Moreover, extracted concepts (represented by term clusters) may be out of desire. Our approach, Core-Concept-Seeded LDA, aims at improving the cluster semantic coherence and providing human expected concepts.

3. Background

LDA is a probabilistic approach modeling topics for document clustering. A topic is assumed to be a distribution over words (ϕ), and a document is assumed to be a distribution over topics (θ). Documents are clustered based on the similarity of their topic distribution. The generative process is vital to understand the philosophy behind LDA. Blei [6] described it as following:

1. For the k^{th} topic in a set of K topics, draw a word distribution $\phi_k \sim Dirichlet(\beta)$.
2. For the d^{th} document in a corpus of D documents, draw a topic distribution $\theta_d \sim Dirichlet(\alpha)$.
3. For each word w_n of the N words in the d^{th} document:
   (a) Select a topic $z_n \sim Multinomial(\theta_d)$.
   (b) Choose a word $w_n \sim Multinomial(\phi_{z_n})$.

where α and β are respectively the prior parameter of Dirichlet distributions θ_d and ϕ_k. θ is a $D \times K$ matrix, ϕ is a $K \times V$ matrix, V is the number of unique words in the vocabularies of the corpus. The idea behind LDA is exploiting the co-occurrence information of words to extract the latent topic structure by maximizing the probability of the corpus generated from the model. Usually, Variational Inference [6] or Gibbs Sampling [17] is used for model training. The latter samples topic of words from a Markov chain which finally converges to a stable distribution.

However, LDA will pay more attention to these statistically prominent topics with frequent terms. The non-frequent terms referring to different real-world topics are more likely to be mixed in a semantically ambiguous topic. To solve this problem, seed based LDA models have been proposed. The general idea behind them is to use seed information

as prior knowledge to guide LDA and deliver topics more relevant to the user's interests. The seed information is composed of S seed sets, each of which contains seed words related to a topic. Four models are considered in our work: *Z-labels*, *Seeded LDA M1*, *Seeded LDA M2*, and *Seeded LDA*.

In LDA model, a word is assigned to any topic with a certain probability. In Contrast, *Z-labels* model constrains the topics of seed words to be assigned to, while other words are free to sample any topic. The parameter π specifies the probability of a seed word generated from the constrained topic. The idea of Z-labels is that using partial supervision information over some words, the topic sampling of other words through the Markov chain is impacted. *Seeded LDA* is derived from Z-labels by using seed information differently. It is composed of two models: *Seeded LDA M1* and *Seeded LDA M2*. In the first model, each topic k is represented as a mixture of a "regular topic" distribution ϕ_k^r and a "seed topic" distribution ϕ_k^s. The former can generate any words (including seed words), while the latter can only generate seed words. A parameter π specifies the probability of a seed word generated from a seed topic. The original (LDA) conditional distribution for topic sampling is $q_{i,k} = p(z_i = k| Z_{-i}, \alpha, \beta)$. In Z-labels, the seed word w has the probability of $q_{i,k}$ to be generated by its constrained topic, and the probability of $(1-\pi)q_{i,k}$ by other topics. While in Seeded LDA M1, w has the probability of $\pi\phi_k^s + (1-\pi)\,\phi_k^r$ to be produced by its constrained topics, and the probability of $(1-\pi)\,\phi_k^r$ by under other topics. Looking back to the basic LDA, *a symmetric β is used as the prior distribution of the word distribution for each topic, without bias on generating words*. However, the idea of both Z-labels and Seeded LDA M1 is roughly like using an asymmetric β where seed words have more chance to be generated, thus each topic has its own preference on words. *Seeded LDA M2* uses a symmetric prior parameter for word distribution, but considers an asymmetric prior parameter for topic distribution. In Seeded LDA M2, a document will sample a group to determine its topic preference. A group (also called seed topic), has its own group-topic distribution which is used as the prior for the document-topic distribution. During the group sampling process of a document, the number of times each group occurred in this document (ie. sum of the frequencies of all seed words in each group) is used to calculate a probabilistic distribution where a group id s is sampled for this document. Then, the group-topic distribution ψ_s (an asymmetric prior distribution) is used as the prior of the topic distribution θ_d. By this mechanism, a document will be more likely to choose topics relevant to its seed words.

4. Our Approach: Core-Concept-Seeded LDA

The *Core-Concept-Seeded LDA* is an adaptation of seed based LDA models for term partition over *CC*s. Seed based LDA models are originally designed for document clustering, and the authors suggest using the seed words with a strong ability to distinguish document categories. As a consequence, a better document-topic distribution is obtained for document representation. But in our approach, the topic-word distribution (ϕ) is the key for term clustering. We consider terms instead of words as the vocabulary of LDA models. Then, we propose a different CC-seed sets setting compared with their model. Specifically, a *CC-seed* set of a topic is designed to include hyponyms or synonyms of a *CC*. In addition, a hard constraint that no overlap of any two *CC-seed* sets is added, while this overlap is allowed in the original models (i.e. one seed word can be shared by two or more topics). Finally, unlike seed based LDA models, our approach provides non-overlap clusters.

4.1. Steps of the Core-Concept-Seeded LDA

The *Core-Concept-Seeded LDA* is composed of three main steps: 1) Material Preparing, 2) Model Training, 3) Cluster Formation (Fig 1). In *step 1*, we prepare the term dataset, the *CC-seed* sets, and the document-term matrix. Firstly, the corpus is pre-processed with text segmentation, part of speech tagging, dependency parsing, and noun phrase (NP) lemmatisation. NP lemmas are extracted and filtered to constitute the terms of the vocabulary. For filtering, NP lemmas with corpus frequency less than 3 or included in the "stop words" list are eliminated. Then, the *CC-seed* sets are extracted by either a domain expert or automatically (see the next section). Finally, the $D \times V$ document-term matrix is constructed where each entry represents the frequency of a term in a document. In *step 2*, the *CC-seed* sets and the document-term matrix are the inputs of seed based LDA models. We choose the number of topics the same as the number of CC-seed sets: $K = S$ (one CC-seed set guides one topic). During the training process, a topic is labeled a priori by a *CC* and guided by its *CC-seed* set. Then a $K \times V$ topic-term probability matrix ϕ is generated. Each topic gathers terms by assigning each of them a probability value. A term w has a topic probability vector $\phi_{:,w}$

representing its closeness to all topics, but implying topic overlap. In *step 3*, to deal with topic overlap, a cluster is associated with each topic. Then a term w is assigned to a cluster associated to a topic k if $k = \underset{1\le t\le K}{argmax}\ \phi_{t,w}$. Finally, a cluster associated with topic k is labeled by its CC_k. Thus, K labeled clusters are formed, including the CC that labels it and its seed set.

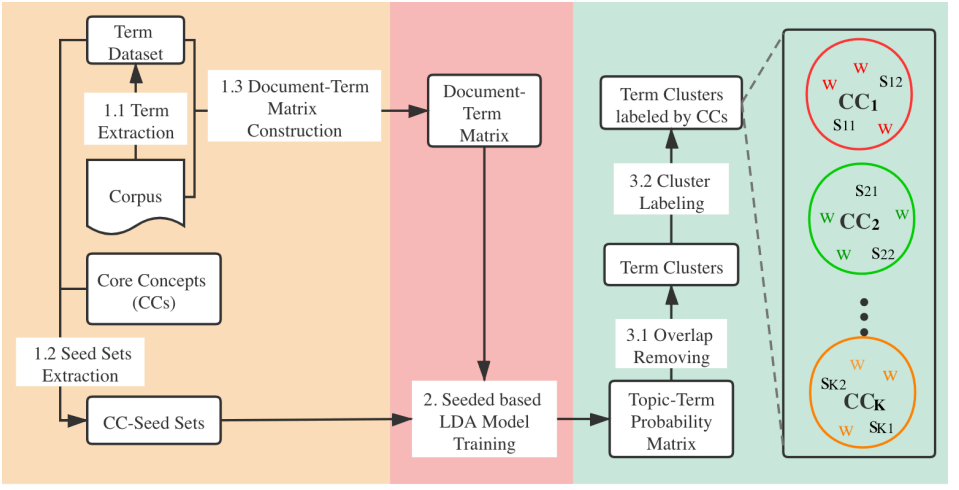


Fig. 1. Core-Concept-Seeded LDA steps for term clustering over core concepts.

4.2. CC-seed Sets

The CC-seed sets, as one of the inputs of seed based LDA models, are keystones in our approach and directly impact the quality of topics. The authors of seeded LDA [20] use CC-seed sets that have a strong ability for discriminating categories of documents. Therefore, they suggest the information gain based method that requires labeling each document by a CC. Unlike the idea of the previous method, we suggest using the frequent synonyms and hyponyms of a CC as seed terms of a topic labeled by this CC. We assume that a term occurs frequently with its hypernym. If we use as seed terms the synonyms and hyponyms of a CC, its labeled topic can cover more other terms referring to it.

Indeed, in the Gibbs Sampling process, the conditional probability of term w in document d under topic k is calculated by $\frac{n^{(d)}_{-i,k}+\alpha}{\sum_u^K (n^{(d)}_{-i,u}+\alpha)} \times \frac{n^{(w)}_{-i,k}+\beta}{\sum_{w'}^W (n^{(w')}_{-i,k}+\beta)}$, where α and β are the hyper-parameters of the model, $n^{(d)}_{-i,k}$ is the number of terms (except w) in document d assigned to topic k, and $n^{(w)}_{-i,k}$ is the number of terms (except w) in the whole corpus assigned to topic k. The idea of seed based LDA models is to force seed words of a topic k to be sampled for it and utilize the topic information to impact the topic sampling of other terms. If w refers to CC_k, then $n^{(d)}_{-i,k}$ would be bigger than $n^{(d)}_{-i,u}, u \neq k$. Based on our assumption, the seed terms of topic k will largely contribute to $n^{(d)}_{-i,k}$, therefore lead w to sample k as its topic. Then, the topic information of w will be added to guide the topic sampling of its hyponyms. We use as seed terms synonyms and hyponyms that covers the widest semantic space of their CC, so that all terms referring to it will get the impact of its hypernyms and synonyms. In this paper, to form *CC-seed* terms, we query hyponyms and synonyms of CCs from DBpedia, then a domain expert is involved to validate them.

Otherwise, the hybrid score based method proposed in [30] may be adapted to extract CC-seed sets. Basically, the hybrid score measures the importance of a term to a target sub-domain. Like the method used by the authors of Seeded LDA, it requires the prior labeling of each document by a CC. However, it considers not only the discriminating ability but also the domain relevance and consensus of a term.

5. Experiments

We performed 2 kinds of experiments on 2 corpora. The first kind aims to evaluate our approach and compare it to some baselines for term clustering. We have conducted experiments on both relevant term dataset (i.e. dataset

including only terms relevant to the ontology domain) and whole term dataset (i.e. dataset including all extracted terms, as explained in section 4.1). Experiments on the latter aim to analyse the impact of irrelevant terms on the performance of our approach. The second kind of experiments conducted also on the whole term dataset aims to compare the impact of seed set extraction methods, to analyse the effect of seed set size, and to investigate the adaptation of our approach for imbalanced dataset.

5.1. Experiment Settings

Corpora. To our knowledge, there is no benchmark for term clustering as a task of ontology building. Therefore, we considered two domain corpora and built manually a gold standard of each. The first corpus (*CS*) is about the computer science domain, which is a part of the dataset Web of Science offered by [21], including 5747 documents. Each of them is an academic paper labeled by a sub-domain (used as a *CC*) of computer science, consisting of several keywords and the abstract. The second corpus (*Music*) is about the music domain. It contains 10000 unlabeled documents, which are randomly sampled from the original corpus provided by [9].

Gold Standard Building. All extracted terms (see section 4.1) are labeled manually to form a domain gold standard (GS), i.e. a term that refers to a *CC* is labeled by this *CC*, otherwise labeled by "Others". The labeling task was done by two annotating groups of computer science students. The Cohen's Kappa coefficient [15] is used to measure inter-annotator agreement. We obtained the coefficient value around 0.885 for *CS* corpus, and around 0.813 for *Music* corpus. For those terms with contradiction in annotation, we discussed and decided their final labels. 10 *CC*s for *CS* corpus (subdomains of the original corpus) and 5 *CC*s for *Music* corpus (general concepts of the music domain from DBpedia) are used for labelling terms. The final gold standards for two corpora are depicted in Table 1.

Table 1. The number of terms labeled by each *CC* class or by "Others" in the 2 corpora.

CS corpus		*Music* corpus	
Data Structures(DS): 323 Cryptography(C): 230 Software Engineering(SE): 248 Computer Graphics(CG): 369 Network Security(NS): 247 Computer Programming(CP): 157 Algorithm Design(AD): 118 Operating Systems(OS): 170 Distributed Computing(DC): 167 Machine Learning(ML): 298 Others(O): 5934	relevant terms: 2327 all terms: 8261	Musicians(M): 1297 Albums(A): 484 Genres(G): 395 Instruments(I): 211 Performances(P): 485 Others(O): 9457	relevant terms: 2872 all terms: 12329

Baselines. We compare our proposal to 2 kinds of baselines: 1) *unsupervised clustering* baselines and 2) *semi-supervised clustering* baselines. For the first one, we consider LDA and K-means with word2vec for term representation. For the second one, we consider the similarity measure based method (SMBM)[1] with word2vec for term representation and the cosine similarity between a term and each *CC* to add it to the cluster of its closest *CC*.

Evaluation Metrics. We use micro precision (P) and micro recall (R) [34] as metrics to evaluate the performance of our approach and compare it with baselines. Let's C and GS be 2 partitions of sets of terms, $C = [C_1, C_2, ..., C_K]$ are the K formed clusters and $GS = [GS_1, GS_2, ..., GS_S]$ are the S CC-classes of the gold standard (class "Others" is excluded) associated respectively to $CC_1, CC_2, ..., CC_S$. We don't take into account the False Positives caused by terms from the class "Others" since in this paper we don't deal with non-domain terms. Therefore in our experiments, P equals to R defined as: $P = R = \frac{\sum_k |C_k \cap GS_{l_k}|}{\sum_s |GS_s|}$, where C_k has the label of GS_{l_k} (i.e. CC_{l_k}).

For a semi-supervised method, the label CC_{l_k} $(1 \leq l_k \leq S)$ of a cluster is assigned a priori (called "prior labels"), while it is not the case for unsupervised methods. However, to evaluate the performance of the latter, we have to label clusters a posteriori. For that, we use the majority labeling method as in [37]. The label of a cluster C_k is CC_{l_k} where $l_k = \underset{s}{argmax}|C_k \cap GS_s|$ that means CC_{l_k} is the label CC_s of the class GS_s that has the largest overlap with

C_k. Metrics using the majority labeling measure the potential largest number of true positives (TP) that unsupervised clustering-based methods can achieve.

Parameters Setting. For all *LDA models*, we follow the parameter setting of α and β from [19] where $\alpha = 1/K$, $\beta = 0.01$, and keep other parameters as their default setting. While for *word2vec*, we use the skip-gram model as it can achieve better performance [11]. We also follow its parameter setting where vector dimension is 300, window size is 10, sub-sampling threshold is 1e-5, minimum count is 5, and learning rate is 0.025.

5.2. Experiment Results

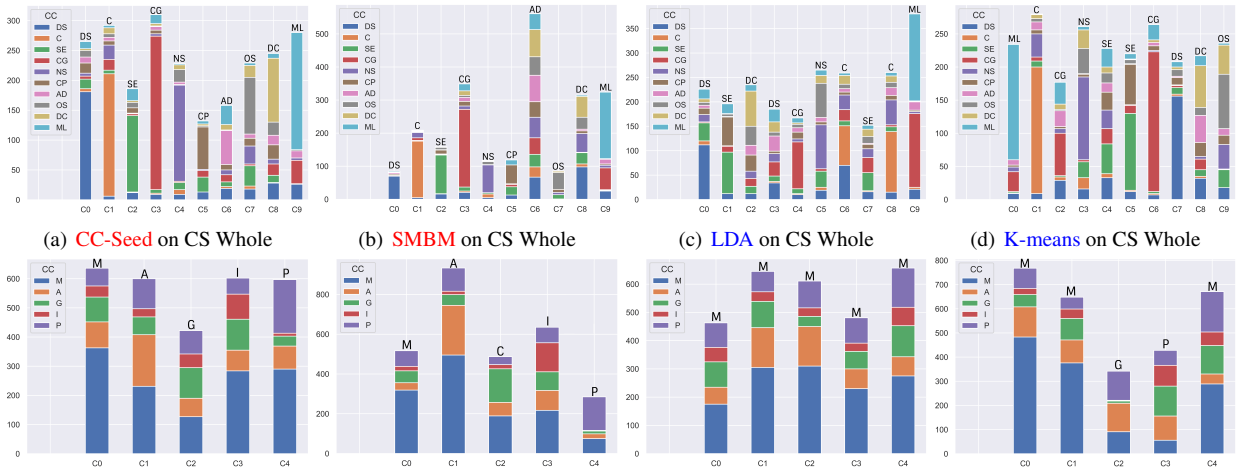
We performed two kinds of experiments. In the first one, we compared our approach (the adaptation of the 4 models denoted *CC-Seed*, *CC-SeedM1*, *CC-SeedM2*, *CC-Z-labels*) with baselines (LDA, K-means, SMBM) over two corpora and their two datasets and with *CC-seed* sets. In the second one, only *CS* corpus and Seeded LDA model are used. *Music* corpus is not considered because its documents are unlabeled whereas labels are needed for CC-seed sets extracted by information gain and hybrid score based methods. For each metric, the experiments are repeated 20 times for each model.

Comparison with Baselines. Table 2 shows that *CC-seed* and *CC-seedM1* outperform all the methods either on relevant or whole dataset of *CS* corpus. For both corpora, CC-seeded LDA models don't benefit a lot from the whole dataset where the *CC-SeedM2* and *CC-Z-labels* even get worse results, which suggests that CC-seeded LDA models don't take advantages of irrelevant terms as context information. CC-seeded LDA models perform worse on *Music* corpus than on *CS* corpus. In addition, for *Music* corpus unsupervised approaches outperform semi-supervised approaches and CC-Seeded LDA models achieve worse than SMBM. To better understand the behavior of these methods, we analyse the composition of clusters learned from the whole dataset. Fig 2 shows the composition of clusters learned by *CC-Seed*, SMBM, LDA and K-means. For *CS* corpus, *CC-Seed* achieved the highest value of TP (1458), and each cluster includes a high number of terms referring to its *CC*. It learned clusters with good semantic coherence. However, other methods produced clusters with low semantic coherence, for instance, the clusters "C6" and "C8" formed by SMBM (Fig2(b)), the clusters "C6" and "C9" formed by LDA (Fig2(c)), the cluster "C4" learnt by K-means (Fig2(d)). In addition, the semantic separateness of these clusters is very bad: their composition is a mix of similar proportions of terms referring to several *CC*s. For *Music* corpus, the partitions of all the methods are bad. Semantic coherence and separateness of clusters are poor: the majority of clusters are dominated by the terms of the Musician class. For example, with majority voting the clusters produced by LDA are all labeled as "M" (Fig 2(g)); for Kmeans 3 clusters labeled as "M" and no cluster labeled as "A" or "I" (Fig 2(h)). The similar problem occurs for CS corpus: no cluster labelled as "CP", "AD", or "OS" for LDA (Fig 2(c)), no cluster labelled as "CP" or "AD" for Kmeans (Fig 2(d)). This situation leads to the omission of concepts when using these unsupervised approaches for domain ontology building. In addition, the fact that the performance of unsupervised methods is better than that of CC-seeded models (see above), does not mean that the quality of their partition is better than that of CC-seeded models. SMBM somehow produces a little better partition than others ("C0" and "C4" in Fig 2(f)).

Table 2. The performance (P or R) of each method on different datasets. Semi-supervised methods are in red, unsupervised methods are in blue.

Corpus	Dataset	CC-Seed	CC-SeedM1	CC-SeedM2	CC-Z-labels	SMBM	LDA	K-means
CS	CS Relevant	**0.5804**	0.5605	0.5503	0.5160	0.5071	0.3789	0.5345
	CS Whole	**0.6323**	0.6159	0.4947	0.4010	0.5071	0.4566	0.5214
Music	Music Relevant	0.3294	0.3171	0.2749	0.2940	0.3670	0.4548	**0.4701**
	Music Whole	0.3326	0.3151	0.2389	0.2381	0.3670	0.4516	**0.4817**

Efficiency of Seed Set Kinds. We have conducted experiments on the CS whole dataset with CC-Seed to check the pertinence of CC-seed sets kinds and to compare then our approach to the original seeded LDA. It is important to remind that the latter uses *information gain* score based method for seed set extraction. *IG* and *Hybrid* are the CC-seed sets extracted respectively by *information gain* score based method and *hybrid score* based method. As mentioned before, in our approach, a CC-seed set includes hyponyms and synonyms of a CC. Each CC-seed sets kind comprises 100 terms with 10 terms/set. The seed terms of *IG* and *Hybrid* are more or less semantically close to their CC, some of them are hyponyms of the CC. *IG* has 27 common terms with the GS, and *Hybrid* has 24. The results

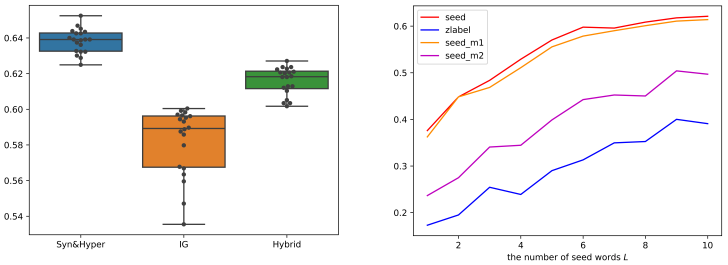


(a) CC-Seed on CS Whole (b) SMBM on CS Whole (c) LDA on CS Whole (d) K-means on CS Whole

(e) CC-Seed on Musics Whole (f) SMBM on Music Whole (g) LDA on Music Whole (h) K-means on Music Whole

Fig. 2. Cluster composition of CC-Seed, SMBM, LDA and K-means. X-axis represents the clusters. Y-axis represents the number of terms (terms of the class "Others" are not counted) in each cluster. The CC label of each cluster (prior labels for semi-supervised methods, majority labels for unsupervised methods) is mentioned above its bar. The number of TP is mentioned above each sub-graph.

show that CC-seed sets are the best kind for term clustering over CCs (Fig 3(a)). *IG* and *Hybrid* can be also used, the performance with *Hybrid* is better. This result proves that our approach outperforms the original seeded LDA.



(a) Efficiency of seed sets kinds (b) Impact of CC-seed set size

Fig. 3. Experiments on seed sets. P or R is used as criteria.

Impact of the Size of CC-seed sets. We also experiment on all seed based LDA models with various sizes of the seed set L ($L \in [1, 2, ..., 10]$). Each value of L will go through 20 experiments and the average performance is used. Each time, we randomly choose $L \times S$ seed terms from *CC-seed* sets as seed terms (L terms for each *CC*), while *CC*s are always used as seed terms. The final result (Fig 3(b)) shows that the performance is boosting with the growth of L. However, a small increment can be expected when L is higher than 6. Among all these models, *CC-Seed* and *CC-SeedM1* can make the most use of *CC-seed* sets even just with one seed term (the *CC*) for each topic.

Unbalanced Corpus and Core-concept CC-seeded LDA. Fig 2 shows that none of those methods provide relevant partition on the Music corpus. The analysis of this corpus and more specifically its gold standard (Table 1) allows highlighting a big variance between the size of its CC-classes that can cause the bad partition. Indeed, all LDA models produce the roughly equal size of clusters (see the online supplementary materials[1] for the results of other seed based LDAs). However, the size of the CC-class of "M" (Musician) is six times the size of the CC-class of "I" (Instrument). Assigning to clusters associated with "M" or "I" the same number of terms will produce a big false positive rate.

But why the cluster of a dominant *CC* (i.e. *CC* of a *CC-class* with a large number of terms w.r.t the others *CC-class*es) can not be big and the cluster of non-dominant one can not be small? The reason is related to the LDA approach. LDA generates topics by distributing the relevant terms to a topic with the highest probability. However, the total probability mass (i.e. 1) is shared by all terms in a topic. If a topic refers to a dominant CC, each term under this

[1] Details of the datasets and the cluster information of this paper https://github.com/jason-huanghao/Core-Concept-Seeded-LDA.

topic can get merely a small probability. Therefore, some terms are easily absorbed by other topics associated with non-dominant *CC*s (these terms may have a bigger probability under these topics). We conducted some experiments on the Music corpus to verify our guess. We assumed that allocating more topics for a *CC* will increase the shared probability mass and will improve the partition quality. One way is to increase the number of topics associated with each CC in the same way (e.g. each of the five *CC*s gets 2 topics then $K = 10$). Another solution is to use the number of topics proportional to the number of terms of each CC-class).

We checked these two strategies with some experiments: exp1, exp2, exp3, and exp4 for the first one, and experiments exp5, exp6, and exp7 for the second one. Table 3 shows that the first strategy is not helpful and the performance goes lower as the K grows. However, the results of exp5, exp6, and exp7 verify our guess and support our second strategy. The CC-seed achieved better performance than the SMBM (0.4089 vs 0.3670 on relevant dataset, and 0.3888 vs 0.3670 on whole dataset). But the total number of topics K should be reasonably small since the performance is not ideal with a large K.

Table 3. Performance (P or R) of CC-seeded LDA on Music corpus with different combinations of topic number for CCs. ↑ means the performance is better than the original group, and ↓ means the worse result.

EXP	M(1297)	A(484)	G(395)	I(211)	P(485)	Relevant	Whole
original	1	1	1	1	1	0.3294 -	0.3326 -
exp1	2	2	2	2	2	0.3041 ↓	0.3242 ↓
exp2	3	3	3	3	3	0.2942 ↓	0.3089 ↓
exp3	4	4	4	4	4	0.2997 ↓	0.3011 ↓
exp4	5	5	5	5	5	0.2947 ↓	0.2920 ↓
exp5	6	2	1	1	2	**0.4089** ↑	**0.3888** ↑
exp6	13	5	4	2	5	0.3842 ↑	0.3761 ↑
exp7	26	10	8	4	10	0.3767 ↑	0.3631 ↑

6. Conclusion

In this paper, we propose the Core-concept seeded LDA, a new semi-supervised approach for concept formation, and more specifically for the formation of clusters where each one includes terms referring to a single CC of an ontology domain. This approach is an adaptation of seeded LDA models. We tackled two problems of existing clustering-based methods, i.e., the difficulty of labeling clusters and the low semantic coherence of clusters. The experimental results indicate that our proposal works better than baselines on a corpus with balanced classes or a corpus with imbalanced classes and a suitable number of topics for each *CC*. Among all these seed based LDA models, we recommend using the seeded LDA for its higher performance and stability. Concerning the seed sets, we suggest using the synonyms and hyponyms of the *CC*s and we have shown that *CC-seeded LDA* outperforms the original *seeded LDA*. For future works, irrelevant term elimination would be an important direction because it impacts directly the behavior of LDA models and contributes to decreasing the number of False Positives. In addition, applying our approach to an imbalanced corpus without knowing a priori the size of CC-classes is a challenging task. For that, one can use some optimization algorithms to determine the number of topics for each *CC*.

References

[1] Albukhitan, S., H.T.A.A., 2017. Arabic ontology learning using deep learning, in: Proceedings of the International Conference on Web Intelligence, pp. 1138–1142.

[2] Andrzejewski, D., Zhu, X., 2009. Latent dirichlet allocation with topic-in-set knowledge, in: Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing, pp. 43–48.

[3] Asim, M.N., Wasim, M., Khan, M.U.G., Mahmood, W., Abbasi, H.M., 2018. A survey of ontology learning techniques and applications. Database 2018.

[4] Ayadi, A., Samet, A., de Beuvron, F.d.B., Zanni-Merk, C., 2019. Ontology population with deep learning-based nlp: a case study on the biomolecular network ontology. Procedia Computer Science 159, 572–581.

[5] Biemann, C., Bordag, S., Quasthoff, U., 2004. Automatic acquisition of paradigmatic relations using iterated co-occurrences., in: LREC.

[6] Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. Journal of machine Learning research 3, 993–1022.

[7] Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E., 2008. Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment 2008, P10008.

[8] Burita, L., Gardavsky, P., Vejlupek, T., 2012. K-gate ontology driven knowledge based system for decision support .

[9] Camacho-Collados, J., Delli Bovi, C., Espinosa-Anke, L., Oramas, S., Pasini, T., Santus, E., Shwartz, V., Navigli, R., Saggion, H., 2018. Semeval-2018 task 9: Hypernym discovery, in: Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018); 2018 Jun 5-6; New Orleans, LA. Stroudsburg (PA): ACL; 2018. p. 712–24., ACL (Association for Computational Linguistics).

[10] Caraballo, S.A., 1999. Automatic construction of a hypernym-labeled noun hierarchy from text, in: Proceedings of the 37th annual meeting of the Association for Computational Linguistics, pp. 120–126.

[11] Casteleiro, M.A., D.G., Read, W., Prieto, M.J.F., Maroto, N., F., D.M., Nenadic, G., K., J., Keane, J., Stevens, R., 2018. Deep learning meets ontologies: experiments to anchor the cardiovascular disease ontology in the biomedical literature. Journal of biomedical semantics 9, 13.

[12] Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J., Blei, D., 2009. Reading tea leaves: How humans interpret topic models. Advances in neural information processing systems 22, 288–296.

[13] Cimiano, P., 2006. Ontologies. Springer.

[14] Cimiano, P., Staab, S., 2005. Learning concept hierarchies from text with a guided agglomerative clustering algorithm, in: Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods.

[15] Cohen, J., 1960. A coefficient of agreement for nominal scales. Educational and psychological measurement 20, 37–46.

[16] Colace, F., De Santo, M., Greco, L., Amato, F., Moscato, V., Picariello, A., 2014. Terminological ontology learning and population using latent dirichlet allocation. Journal of Visual Languages & Computing 25, 818–826.

[17] Griffiths, T.L., Steyvers, M., 2004. Finding scientific topics. Proceedings of the National academy of Sciences 101, 5228–5235.

[18] Harris, Z., 1968. Mathematical structures of language, in: Interscience tracts in pure and applied mathematics.

[19] Hoffman, M., Bach, F.R., Blei, D.M., 2010. Online learning for latent dirichlet allocation, in: advances in neural information processing systems, Citeseer. pp. 856–864.

[20] Jagarlamudi, J., Daumé III, H., Udupa, R., 2012. Incorporating lexical priors into topic models, in: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 204–213.

[21] Kowsari, K., Brown, D.E., Heidarysafa, M., Meimandi, K.J., Gerber, M.S., Barnes, L.E., 2017. Hdltex: Hierarchical deep learning for text classification, in: 2017 16th IEEE international conference on machine learning and applications (ICMLA), IEEE. pp. 364–371.

[22] Lee, J., Luo, M., 2016. Word clustering for parallelism in classical chinese poems, in: 2016 International Conference on Asian Language Processing (IALP), IEEE. pp. 49–52.

[23] Mahn, M., Biemann, C., 2005. Tuning co-occurrences of higher orders for generating ontology extension candidates. Learning and Extending Lexical Ontologies by using Machine Learning Methods 28, 40–43.

[24] Matsuo, Y., Sakaki, T., Uchiyama, K., Ishizuka, M., 2006. Graph-based word clustering using a web search engine, in: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 542–550.

[25] Mimno, D., Wallach, H., Talley, E., Leenders, M., McCallum, A., 2011. Optimizing semantic coherence in topic models, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 262–272.

[26] Nematzadeh, A., Meylan, S.C., Griffiths, T.L., 2017. Evaluating vector-space models of word representation, or, the unreasonable effectiveness of counting words near other words., in: CogSci.

[27] Ortega-Mendoza, R.M., Villasenor-Pineda, L., Montes-y Gomez, M., 2007. Using lexical patterns for extracting hyponyms from the web, in: Mexican International Conference on Artificial Intelligence, Springer. pp. 904–911.

[28] Pereira, F., Tishby, N., Lee, L., 1994. Distributional clustering of english words. arXiv preprint cmp-lg/9408011 .

[29] Qiu, J., Chai, Y., Tian, Z., Du, X., Guizani, M., 2019. Automatic concept extraction based on semantic graphs from big data in smart city. IEEE Transactions on Computational Social Systems 7, 225–233.

[30] Qiu, J., Qi, L., Wang, J., Zhang, G., 2018. A hybrid-based method for chinese domain lightweight ontology construction. International Journal of Machine Learning and Cybernetics 9, 1519–1531.

[31] Rani, M., Dhar, A.K., Vyas, O., 2017. Semi-automatic terminology ontology learning based on topic modeling. Engineering Applications of Artificial Intelligence 63, 108–125.

[32] Sahlgren, M., 2008. The distributional hypothesis. Italian Journal of Disability Studies 20, 33–53.

[33] Shwartz, V., S.E., S., D., 2016. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. arXiv preprint arXiv:1612.04460 .

[34] Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Information processing & management 45, 427–437.

[35] Thaiprayoon, S., Unger, H., Kubek, M., 2020. Graph and centroid-based word clustering, in: Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval, pp. 163–168.

[36] Widdows, D., Dorow, B., 2002. A graph model for unsupervised lexical acquisition, in: COLING 2002: The 19th International Conference on Computational Linguistics.

[37] Xu, Z., Harzallah, M., Guillet, F., Ichise, R., 2019. Modular ontology learning with topic modelling over core ontology. Procedia Computer Science 159, 562–571.