Core-Concept-seeded based LDA towards Ontology Learning

Hao Huang\(^1\), Mounira Harzallah\(^1\), Ziwei Xu\(^2\), and Fabrice Guillet\(^2\)

\(^1\) LS2N, University of Nantes, France
\(^2\) mounira.harzallah@univ-nantes.fr
\(^3\) ziwei.xu@etu.univ-nantes.fr
\(^4\) fabrice.guillet@univ-nantes.fr

Abstract. Ontologies are powerful semantic models applied for various purposes such as improving system interoperability, information retrieval, question answering, etc. However, modeling domain ontology remains a tough task for humans especially when the concepts and properties are large or evolving, and also when the modeling is performed on a large-scale text data. Machine learning provides a valuable help by automatizing ontology learning from texts. In peculiar for concept formation task, clustering techniques are able to deal with a huge number of terms to extract concepts, i.e. clusters of semantically similar terms. However, current works have the issues of learning relevant clusters for specific domain or making relevant labels for clusters. 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 into account these core concepts, by first assigning each LDA topic to a seeding core concept, then guiding LDA to put terms linked to the same core concept in the same topic. We evaluate our proposal on two textual corpora and compare it to 4 other clustering based approaches (two unsupervised methods and two semi-supervised methods). The results show that our approach beats them significantly on clean corpus, and noisy corpus without serious imbalance on core concept classes (considering number of terms). For noisy corpus with prominent imbalance problem, our SMBM-SW is a good alternative.

Keywords: Ontology Learning · LDA · Term Clustering · Seed Knowledge · NLP.

1 Introduction

With the increasing availability of textual resources on the web, ontology learning from texts becomes a challenging process. It includes five main tasks: terms extraction, concepts formation, taxonomy extraction, ad-hoc relation extraction, and axioms extraction \(^{10}\). Many Machine learning approaches have been proposed for achieving these tasks, generally classified into two main categories:
patterns based approaches and distributional approaches. Pattern-based approaches show quite high precision whereas their recall is low because of the large variability in natural language for expressing a meaning. Distributional approaches are based on the Harris hypothesis which suggests that terms sharing the same context tend to have a similar meaning. Thus, either supervised or unsupervised, they use co-occurrence context distribution to detect semantic relations between terms. Supervised methods usually use a training dataset to learn a model for predicting the classification of a term or semantic relation between a pair of terms. Unsupervised approaches are either measure-based approaches or clustering-based approaches. The former computes a score of the semantic closeness or relatedness of a pair of terms based on symmetric or inclusion measures. The latter gathers terms semantically close on the same cluster. Generally, the supervised approaches outperform non-supervised approaches. However, unlike unsupervised approaches, it requires extra effort on building a training dataset. In addition, a recent study shows that their good performance is due to the lexical memorization, and they are sensitive to the distribution of cases (terms or term pairs) in a particular corpus.

We are interested in term clustering methods for concept formation. We focused on Latent Dirichlet Allocation (LDA) since it can deal with a huge number of documents and tackle the issue of text sparsity. Approaches based on LDA for concept formation deliver a term probability distribution for each topic/cluster, then address the problem of cluster labeling either by a domain expert or some top terms with high probability in a topic. However, two other issues are ignored. 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. 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 from another domain. To solve these issues, we propose to use core concepts (CCs) from a domain ontology as prior knowledge, and then to adapt seed based LDA models in order to gather terms semantically close to a core concept into the same topic/cluster. A core ontology of a domain is a basic ontology composed only of the minimal concepts (i.e core concepts) that allow defining the other concepts of the domain. In our approach (called Core-concept-seeded LDA) a cluster label is not to identify, rather it is fixed previously.

In this paper, we consider four seed based LDA models (Z-labels LDA, Seeded LDA model1, Seeded LDA M2, Seeded LDA). Unfortunately, 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 the topic of "earn" has as seed terms "company" and "quarter". Further, the seed based LDA models is dedicated to document clustering. To the best of our knowledge, none of them has been used for term clustering to reach concept formation in 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 on the Latent Dirichlet Allocation (LDA) model and its variants for document clustering, with a focus on the goal and principles of each one. In section 4, we present our proposal for adapting these models to term clustering over core concepts, and we propose some suitable methods for seed term extraction. In section 5, we present several experiments conducted on clean and noisy corpora, a comparison to unsupervised and semi-supervised baselines, and the effectiveness of proposed seed sets. Section 6 discusses the performance of our approach on imbalanced corpus before concluding.

2 Related Works

Several methods based on clustering algorithms as K-means, hierarchical clustering, and agglomerative clustering [14] have been used for ontology learning. The clustering is performed over terms, and each cluster of terms is considered as a concept of the ontology. Then, relations between clusters, generally hypernym relations, are identified to build an ontology [12,15].

With the prevalence of word embedding, some researchers proposed similarity measure based methods (SMBM) that use similarity measure and word2vec (a term vector space representation) for ontology learning, where terms are partitioned into the cluster of a core concept that has the highest similarity with them [1,7,8].

Usually used for document classification, LDA is also used for ontology learning. It is a probabilistic method that forms latent topics where each term is assigned to a topic with a given probability (i.e. topic-word distribution) and a document is represented as a distribution of topics (i.e. document-topic distribution). We can distinguish three kinds of approaches based on LDA for ontology learning. The first is developed for concept formation without any prior knowledge where often a topic is considered as a concept of an ontology. Then, the semantic closeness among topics is computed to establish a hierarchy [23,29,31].

As topics are defined by word distribution, a term may be assigned to different topics leading to topic overlapping. To remove the overlap, term partition is achieved by associating to each topic a cluster including only terms with higher distribution probability for this topic [30]. Otherwise, for the purpose of ontology learning, clusters need to be labeled to provide them meaningful interpretation. Labeling can be made manually and may allow us to detect the cluster irrelevance for the ontology domain. However, even manual labeling is not easy to perform if the terms of a cluster refer to different things or if their sense is ambiguous because it affects the semantic coherence of the cluster. To tackle the lack of semantic coherence, some approaches consider only the top terms of a topic as elements of the cluster associated with this topic (e.g. only the first 5 terms having the higher distribution probability for the topic) [23,28]. The weakness of these approaches is to ignore terms without higher probability in any topics, terms that may hold a key meaning for the ontology to build.

The second kind of approach for ontology learning does not aim at term clustering for concept formation. Rather, they use LDA to compute the condi-
tional probability between term pairs and determine the roots as terms whose occurrence is not implied by the occurrence of other terms of the corpus (called "aggregate terms for each root"). The root terms are considered as the concepts of the ontology to build. Then hypernym relations between root terms are extracted to learn a terminology \[11,20\]. As noted before, the default of this approach is to ignore all the other aggregate terms that can bring or hold a key meaning for the ontology to build.

Finally, the third kind of approach performs a term clustering based on the topical feature space learned by LDA \[8,28\]. In these works as before, the issues of cluster labeling, semantic coherence, and relevance remain to tackle.

3 Background

LDA is a probabilistic approach modeling topics from a collection of documents for document clustering. The model assumes that a topic is a distribution of words, and a document is a distribution over topics. Documents having the same topic distribution are expected to be similar. The idea behind LDA is exploiting the co-occurrence information of words, and extracting the latent thematic structure (splitting words into topics in a probabilistic way) by maximizing the probability of the corpus generated from the model.

The generative process is vital to understand the philosophy behind the LDA. Blei in \[4\] describes it as following:

1. For each topic \( k = 1, ..., K \):
   - Draw a word distribution \( \phi_k \sim \text{Dirichlet}(\beta) \) for topic \( k \).
2. For each document \( d \) in a corpus of \( D \) documents:
   - Draw a topic distribution \( \theta_d \sim \text{Dirichlet}(\alpha) \) for document \( d \).
3. For each word \( w_n \) in document \( d \):
   - (a) Select a topic \( z_n \sim \text{Multinomial}(\theta_d) \).
   - (b) Choose a word \( w_n \sim \text{Multinomial}(\phi_{z_n}) \).

where \( K \) is the number of topics, \( D \) is the number of documents, \( W \) is the total number of unique words in the corpus. The \( \alpha \) and \( \beta \) are the prior of distribution \( \theta_d \) and \( \phi_k \). \( \phi \) is a \( K \times W \) matrix, \( \phi_k \) is the word distribution of topic \( k \). \( \theta \) is a \( D \times K \) matrix, \( \theta_d \) is the topic distribution of document \( d \).

LDA is a powerful tool for exploring latent topics in a text corpus. However, since the goal of LDA is to maximize the probability of the observed data, it will pay more attention to these statistically prominent topics that contain many frequent terms. However, those non-frequent terms referring to different real-world topics are more likely to be mixed in a topic produced by LDA. Therefore, the discovered topics will not be semantically meaningful and contribute less to the downstream tasks \[9\].

To solve this problem, seed based LDA models have been proposed. The general idea behind them is using some seed information as prior knowledge to guide LDA and deliver topics more relevant to the user’s interests. The seed information is a group of seed sets \( SS \), each of which contains a list of seed
words related to a topic to form. In our work, four models are considered. They are Z-labels, Seeded LDA M1, Seeded LDA M2, and Seeded LDA.

In the basic LDA model, a word is assigned to any topic with a certain probability. In Z-labels, however, those words from a seed set are constrained to be assigned to either a specific topic or a limited list of topics. The parameter \( \pi \) specifies the probability of a seed word generated from the constrained topics. The idea of Z-labels is using partial supervision information over some terms (based on expert’s knowledge) into LDA and impacting the topic sampling of other terms through the MCMC chain.

Getting inspiration from Z-labels, the authors of Seeded LDA used the seed information in a different way, and decline it in 2 models: Seeded LDA M1 and Seeded LDA M2. In Seeded LDA M1, a document is a mixture of \( K \) topics, where each topic \( k \) is represented as a mixture of a ”regular topic” distribution \( \phi^r_k \) and a ”seed topic” distribution \( \phi^s_k \). The former can generate any words (includes seed words), while the latter can only generate seed words (from seed set of topic \( k \)). A parameter \( \pi \) represents the probability of a seed word generated from the seed topic. It seems that Seeded LDA M1 uses a totally different mechanism. However, the idea of using seed topics here is similar to that of using constrained topics in Z-labels. In Z-labels, topics from the constrained topic set share the same probability \( \pi \) to be selected by a seed word \( w \). While in Seeded LDA M1, \( w \) has different probabilities on choosing all seed topics. The probability of seed word \( w \) generated by seed topic \( k \) is \( \pi \times \phi^s_{k,w} \). Looking back to the basic LDA, where the symmetric \( \beta \) is used as the prior distribution of the word distribution for each topic (has no bias when generating words). However, the idea of both Z-labels and Seeded LDA M1 is roughly like using an asymmetric \( \beta \), thus each topic has its own preference on words.

Seeded LDA M2 uses the symmetric prior for word distribution, but considers an asymmetric prior distribution for topic distribution. In Seeded LDA M2, a group, also called a seed topic, has its own group-topic distribution which is used as the prior for the document-topic distribution. During the training process, the seed words information (the frequency of each seed words in a document) of a document \( d \) is used to determine a group id \( s \) of this document. Then, the group-topic distribution \( \psi_s \) (an asymmetric prior distribution) is used as the prior of the topic distribution \( \theta_d \). This mechanism will lead a document to choose topics relevant to the seed words within the document. In other models, however, each document uses the symmetric prior \( \alpha \) to generate their topic distribution and have no bias on selecting topics.

4 Our approach : core-concept-seeded LDA

Core-concept-seeded LDA is an adaptation of seed based LDA models for concept formation. For that purpose, at first, a core concept represented by a set of seed words is associated with each topic, then LDA is guided by the core concept of each topic to assign more relevant terms with a high probability to the topics. By more relevant terms, we mean the terms that are better related to
a single core concept, such as synonyms, hyponyms, and more generally terms semantically closed to the core concept. Then, the cluster generated from each topic is implicitly labeled with its associated core concept.

Seed based LDA models are originally designed for document clustering, they 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, as we focus on the topic-word distribution for term clustering, the seed term setting differs from the existing seed based LDA. For that, the seed term set of each topic is chosen as hyponyms or synonyms of its associated core concept. In addition, we have a hard constraint that there is no overlap between any two seed sets, while this overlap is allowed in the original models (i.e. one seed word can be shared by two or more topics).

**Steps of the Core-concept-seeded LDA** Core-concept-seeded LDA is composed of three main steps: 1) pre-processing, 2) seed based LDA training, and (3) cluster formation (Figure 1).

![Fig. 1. Core concept seeded based LDA steps for term clustering over core concepts](image)

The inputs consist a selected corpus of a domain and the seed sets of the core concepts from this domain. In the first step, we process the domain corpus (i.e. text segmentation, part of speech tagging,..., term identification and lemmatisation). Noting that the rare terms (with document frequency less than 3) or terms recognized as stop words are eliminated. The seed sets are extracted by either a domain expert or a seed sets extraction method.

In step two, seed based LDA training uses seed sets and the reconstituted corpus as inputs. During the training process, each topic is under the guidance of the seed set of a core concept. Here we explicitly define the map of relations “core concept − seed set − topic”. This means one core concept has only one seed set, and one seed set is designed to guide a specific topic in the LDA model. This relation map latter will be used for identifying the core concept of topics. Going through multiple Gibbs Sampling training iteration, the model is well-trained.
In the last step, we generate labeled clusters based on the topic-word distribution of the model and the relation map (CC-SS-topic map). Indeed, from the trained LDA model, we obtain the topic-word distribution \( \phi \) (a \( K \times W \) matrix), where each topic gathers terms by assigning each of them a probability value. This means each term \( w \) has probabilities vector \( \phi_{:,w} \) representing its closeness to each topic. The overlap is removed by allocating each term \( w \) into cluster \( C_k \) (the cluster generated from topic \( k \) and \( k = \arg \max_{1 \leq t \leq K} \phi_{t,w} \)). Therefore, a group of clusters is generated under topics. Finally, the core concept label of each cluster \( C_k \) is identified by checking the core concept of topic \( k \) from the predefined CC-SS-topic map. Each labeled cluster contains the associated core concept, seed words from the seed set, and terms semantically closed to the core concept.

**Seed sets extraction** The seed sets, as one of the inputs of seed based LDAs, are keystones in our approach. The choice of seed sets can directly impact the quality of topics. In this work, we choose to extract automatically the seed sets associated with the core concepts; alternatively, it could be performed manually. We compare 4 methods for automatic extraction of seed set, and the effectiveness of them will be discussed in the experiment section.

1. **Information gain.** The authors of seeded LDA \[13\] suggest using the seed terms that have a strong ability for discriminating categories of documents. They proposed an information gain method to extract seed terms, especially for document clustering. However, there exists a constrain that the documents should be labeled with the core concepts. Therefore, the information gain can be used to pick up the discriminant seed words of a core concept.

2. **Hybrid score.** With the same constrain (using core concepts as document label), we recommend the hybrid score (idea from \[22\]) to identify the seed words of each core concept. The hybrid score \( \text{hybrid\_score}(w, c) \) is used to measure the importance of a term \( w \) to a target sub-domain (or core concept in our case) \( c \). After the computation of hybrid scores between terms and core concepts, terms are divided into \( S \) (the number of core concepts) sets, each tied with a core concept, and a term is allocated into the set with the highest hybrid score. Thereafter, for each core concept, we rank the terms of its set by decreasing hybrid score for this core concept, then select the top \( L \) terms as the seed words of the core concept. Comparing with the information gain, the hybrid score considers not only the discriminating ability but also the domain relevance and domain consensus of a term.

3. **Synonyms and hyponyms.** With the same constrain (using core concepts as document label), we recommend the hybrid score (idea from \[22\]) to identify the seed words of each core concept. The hybrid score \( \text{hybrid\_score}(w, c) \) is used to measure the importance of a term \( w \) to a target sub-domain (or core concept in our case) \( c \). After the computation of hybrid scores between terms and core concepts, terms are divided into \( S \) (the number of core concepts) sets, each tied with a core concept, and a term is allocated into the set with the highest hybrid score. Thereafter, for each core concept, we rank the terms of its set by decreasing the hybrid score for this core concept, then select the top \( L \) terms as
the seed words of the core concept. Comparing with the information gain, the hybrid score considers not only the discriminating ability but also the domain relevance and domain consensus of a term.

4. Context related terms. Usually, we use the context of terms to classify or cluster terms. Terms that frequently occur in similar contexts should be gathered in the same cluster. Thus, we propose a window-based method (with a parameter window size) to extract the terms that frequently co-occur with the core concept. These terms with a high frequency of co-occurrence would be the seed words of the core concept.

5 Experiments

We performed 3 groups of experiments on 2 corpora. The first one, conducted on clean corpora, aims to evaluate the performance of our proposal for cluster semantic coherence and compare it to some baselines. We considered baselines either with prior cluster labels or without. The second group, conducted on noisy corpora, aims to analyse the impact on cluster semantic coherence of noisy terms. The last group, conducted on noisy corpora, targets for comparing different groups of seed sets and analysing the effect of seed set size.

5.1 Experiment Settings

Corpus and dataset. To our knowledge, there is no benchmark for term clustering task towards ontology building. Therefore, we considered two text corpora and built manually a gold standard of each (i.e. we classified the extracted terms into different core concept classes). The first corpus (CS) is about the computer science domain, which is a part of the dataset Web of Science offered by [16], including 5747 documents. Each of them is an academic paper labeled by a sub-domain (used as our core concept) 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 [6].

Gold standard building. After obtaining the extracted terms, we labeled the extracted terms to form the domain gold standard (GS). Term referred to any core concept will be labeled by the core concept, otherwise labeled by "Others". The labeling task was done by a PhD and four master students from computer science. We used 10 core concepts (the sub-domain labels from original corpus) for CS corpus and 5 core concepts (reference on the general concepts of music domain from DBpedia) for Music corpus. The number of terms of each core concept and each corpus is listed in Table 1. Noting that the noisy corpus contains all terms, while the clean corpus excludes the terms of "Others".
Table 1. The number of terms for each core concept CC of the 2 corpora

<table>
<thead>
<tr>
<th>CS corpus</th>
<th># terms in CC</th>
<th># all terms</th>
<th>Music corpus</th>
<th># terms in CC</th>
<th># all terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structures(DS):</td>
<td>323</td>
<td></td>
<td>Musicians(M):</td>
<td>1297</td>
<td></td>
</tr>
<tr>
<td>Cryptography(C):</td>
<td>230</td>
<td></td>
<td>Albums(A):</td>
<td>484</td>
<td></td>
</tr>
<tr>
<td>Software Engineering(SE):</td>
<td>248</td>
<td></td>
<td>Genres(G):</td>
<td>395</td>
<td></td>
</tr>
<tr>
<td>Network Security(NS):</td>
<td>247</td>
<td></td>
<td>Performances(P):</td>
<td>485</td>
<td></td>
</tr>
<tr>
<td>Computer Programming(CP):</td>
<td>157</td>
<td></td>
<td>Others(O):</td>
<td>9457</td>
<td></td>
</tr>
<tr>
<td>Algorithm Design(AD):</td>
<td>118</td>
<td></td>
<td>clean: 2327</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating Systems(OS):</td>
<td>170</td>
<td></td>
<td>noisy: 8261</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed Computing(DC):</td>
<td>167</td>
<td></td>
<td>clean: 2872</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Machine Learning(ML):</td>
<td>298</td>
<td></td>
<td>noisy: 12329</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others(O):</td>
<td>5934</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Baselines. We compare our proposal to 2 kinds of baselines: 1) unsupervised clustering baselines without prior labels of clusters, and 2) semi-supervised clustering baselines with prior labels of clusters. For the first one, we consider as baselines LDA and K-means (we use word2vec for term vector space representation and vector of core concepts to initialize the centroids of K-means). We compare our proposal to 2 kinds of baselines: 1) unsupervised clustering baselines without prior labels of clusters, and 2) semi-supervised clustering baselines with prior labels of clusters. For the first one, we consider as baselines LDA and K-means (we use word2vec for term vector space representation and vector of core concepts to initialize the centroids of K-means). For the second one, we consider two similarity measure based methods (SMBM) \[1\] as baselines, using cosine similarity between terms vectors in a word2vec term representation. The first (SMBM-CC) using core concepts, computes the similarity between a term and each core concept and adds the term to the cluster of the closest core concept. The second (SMBM-SW) using seed words is our proposal. For each term t, we select the seed word sw which has the best cosine similarities \(\text{cosine}(t, sw)\). Then t is assigned to the cluster of a core concept using sw as its seed word. We propose this baseline based on 2 observations from \[18\]: (1) Terms under a semantic hierarchy are usually clustered together in terms of their word embeddings; (2) Terms that are taxonomically related are usually located near the centroids of the clusters formed by their word embedding offsets.

Evaluation Metrics. We use precision (P) and recall (R) as metrics to evaluate the performance of our approach and compare it to 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 \(GS = [GS_{CC_1}, GS_{CC_2}, ..., GS_{CC_S}]\) are the \(S\) gold standard classes (class ”Others” is excluded), where the number of clusters (or topics) \(K\) equals to the number of core concepts (GS classes) \(S\), however we use different symbols to distinguish them. Here we give the definition of precision and recall:

\[
P = \frac{\sum_k |C_k \cap GS_l|}{\sum_k |C_k|}; \quad R = \frac{\sum_k |C_k \cap GS_l|}{\sum_s |GS_s|}
\]
where the $l_k \in CC$ is the core concept label of $C_k$. For a semi-supervised method, the label $l_k$ of a cluster is pre-designed (the label is called "prior labels"). However, for an unsupervised method, it is not the case and we have to label them to evaluate their performance. The label of cluster $C_k$ is $l_k = \arg\max_{C \in CC} |C_k \cap GSCC|$, by checking the gold standard and choosing the core concept whose class makes the largest overlap with $C_k$. We call this type of label "majority labels", since the class of majority terms determined the label of a cluster. Criteria using these labels measure the potential best performance that an unsupervised clustering-based method can achieve.

5.2 Experiment Results

We compare our 4 approaches, the adaptations of Seeded LDA, Seeded LDA M1, Seeded LDA M2, and Z-labels (denoted respectively: Seed, Seed M1, Seed M2, Z-labels), to the baselines basic LDA and K-means (denoted LDA and K-Mean respectively) where the precision is calculated based on the majority labels; and compare also to SMBM-CC and SMBM-SW where the precision is calculated with prior labels. Experiments are conducted on both clean and noisy corpora. Each evaluation metrics is averaged over 20 repetitions of experiments.

Results on clean corpora Comparing with the unsupervised baselines, our proposals achieved better performances. The Seeded LDA and the Seeded LDA M1 got compatible performances and exceeded other methods. The baseline methods in Table 2 gave unpromising results. An interesting observation is that the results on Music Corpus of all methods seem identical, little differences can be detected among all methods. In Table 3, the Seeded LDA dominates other methods with a moderate priority, followed by the Seeded LDA M1 with similar performance. Considering the semi-supervised baselines, our proposed alternative (SMBM-SW) works better than SMBM-CC which is used in 1.
Results on noisy corpora The precision decreases dramatically on noisy corpora, understandably, because we do not deal with the noisy terms (means classifying noisy terms). Therefore, the precision and recall indicate the same performance (the same number of correctly classified terms).

Table 2. The best performance of each method with majority labels

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Seed</th>
<th>Seed M1</th>
<th>Seed M2</th>
<th>Z-labels</th>
<th>LDA</th>
<th>K-means</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS (P/R)</td>
<td>0.5903</td>
<td>0.5865</td>
<td>0.5722</td>
<td>0.5802</td>
<td>0.4737</td>
<td>0.3095</td>
</tr>
<tr>
<td>Music (P/R)</td>
<td>0.4631</td>
<td>0.4649</td>
<td>0.4596</td>
<td>0.4609</td>
<td>0.4562</td>
<td>0.4592</td>
</tr>
</tbody>
</table>

Results show that the recalls of best competitors have been improved, compared with the one from clean corpora. Performance of the Seeded LDA (denoted Seed) on CS gets 8% of improvement (from 59% to 64%). Meanwhile, the SMBM-SW can even gain 32% of increment on Music corpus. This suggests that noisy terms are useful in term clustering. The better results may benefit from the richer co-occurrence information. For example, "Musician1" and "Musician2" do not co-occur in the same documents if we clear out all noisy terms. However, by adding the noisy term "Music Company", an implicit co-occurrence "Musician1-Music Company-Musician2" helps to model a better topic "Musician".

Table 3. The best performance of each method with prior labels

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Seed</th>
<th>Seed M1</th>
<th>Seed M2</th>
<th>Z-labels</th>
<th>SMBM-CC</th>
<th>SMBM-SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS (P/R)</td>
<td>0.5910</td>
<td>0.5871</td>
<td>0.5756</td>
<td>0.5778</td>
<td>0.2505</td>
<td>0.4271</td>
</tr>
<tr>
<td>Music (P/R)</td>
<td>0.3179</td>
<td>0.2963</td>
<td>0.2820</td>
<td>0.3127</td>
<td>0.2012</td>
<td>0.2852</td>
</tr>
</tbody>
</table>

As shown in Table 4, the 4 seed based LDA variants significantly outperforms the unsupervised baselines on CS corpus, and the Seeded LDA (Seed) achieves the best precision and recall. While the K-means clustering approach achieves its peak in the Music corpus. It seems that the seed based LDAs and the K-means show different advantages when dealing with different corpora.

Table 4. Comparing with unsupervised baseline with majority labels on noisy corpora

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Seed</th>
<th>Seed M1</th>
<th>Seed M2</th>
<th>Z-labels</th>
<th>LDA</th>
<th>K-means</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS (P)</td>
<td>0.1805</td>
<td>0.1786</td>
<td>0.1706</td>
<td>0.1727</td>
<td>0.1513</td>
<td>0.1618</td>
</tr>
<tr>
<td>CS (R)</td>
<td>0.6411</td>
<td>0.6341</td>
<td>0.6058</td>
<td>0.6133</td>
<td>0.5373</td>
<td>0.5745</td>
</tr>
<tr>
<td>Music (P)</td>
<td>0.1288</td>
<td>0.1282</td>
<td>0.1281</td>
<td>0.1283</td>
<td>0.128</td>
<td>0.1405</td>
</tr>
<tr>
<td>Music (R)</td>
<td>0.4575</td>
<td>0.4554</td>
<td>0.4547</td>
<td>0.4557</td>
<td>0.4544</td>
<td>0.4988</td>
</tr>
</tbody>
</table>

Comparing with the semi-supervised baselines (Table 5), again, the seed based LDAs show their strength on CS corpus but weakness on Music corpus. The performances are almost the same as the one in Table 4, which means the seed based LDAs are capable to allocate the best labels and reach their best performances (result is the same as the majority labels, a little difference is due to the variance over 20 times of evaluations). Results of SMBM, however, presents an opposite trend, performing better than SMBM-CC.

Results on Seed Sets. Six groups of seed sets are compared as prior knowledge for Seeded LDA. We conducted experiments only on CS noisy corpus, since
Seeded LDA can not deal with the Music corpus (Discussion part). In G1, each seed set includes the synonyms and hyponyms of a core concept. Seed sets of G2, G3, G4, G5 and G6 are extracted respectively by information gain score based method, hybrid score based method, frequent context methods with window size of 3, frequent context methods with window size of 5, frequent context methods with window size of 7. Results are presented in Figure 2(a).

We checked seed words from G2 to G6. Those of G2 and G3 are more semantically close to their core concepts. Both of them contain core concept terms and some hyponyms of the core concepts. Comparing them with the gold standard (GS), we remark that G2 has 27 common terms with the GS, and G3 has 24 common terms. The possible reason that G3 can beat G2 is that the information gain only considers discriminating terms that can help distinguish the category of documents. However, the hybrid score considers the discriminating ability, domain relevance, and domain consensus. Therefore, G3 includes more relevant terms to the domain, not just to distinguish the class of documents. Seed sets from G4 to G6 have low quality, sharing only 7 common terms with the GS.

![Figure 2](image)

Fig. 2. The effectiveness of seed sets. Recall is used for measurement.

The results prove that the synonyms and hyponyms of core concepts are the best prior knowledge for our proposed method. G2 and G3 can be the ideal candidates, but G3 is better. The results of G4, G5, and G6 show that a window of small size can increase the performance. But none of them can be a good choice as the seed information.

We experiment on all LDA models with various sizes of the seed set $L$ ($L \in \{1, 2, ..., 10\}$) to analyze its impact. 20 times of experiments used for each value of $L$. Each time we randomly choose $L$ seed words from G1 for each core concept, while core concept terms are always used as seed words. The final result (Figure 2(b)) shows that the performance is boosting with the growth of $L$. However, a small increment can be expected for those $L$ higher than 6. We found an interesting phenomenon that the seeded LDA M1 can take the advantage of the
small size of seed information, even in an extreme situation where only the core concept terms are used.

6 Discussions

By checking the cluster information of semi-supervised methods, we analyze why all models performed poorly on Music corpus. The cluster information of each method is organized in Table 6 (clean and noisy corpus). The second column is the size of each core concept class. Other columns are cluster information (#true positives / size of cluster) of all methods. Clusters in each line use the leftmost core concept of this line as their labels. The total number of true positives vs the number of terms in a corpus are showed in the last row of each corpus.

Table 6. The cluster information of different semi-supervised methods on Music corpus

<table>
<thead>
<tr>
<th>corpus</th>
<th>CC</th>
<th>Seed</th>
<th>Seed M1</th>
<th>Seed M2</th>
<th>Z-labels</th>
<th>SMBM-SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>clean</td>
<td>M: 1297</td>
<td>376/633</td>
<td>335/597</td>
<td>332/584</td>
<td>376/643</td>
<td>M 261/435</td>
</tr>
<tr>
<td></td>
<td>A: 484</td>
<td>153/678</td>
<td>142/624</td>
<td>157/710</td>
<td>152/660</td>
<td>A 265/1163</td>
</tr>
<tr>
<td></td>
<td>G: 395</td>
<td>106/457</td>
<td>120/560</td>
<td>92/441</td>
<td>106/498</td>
<td>G 72/247</td>
</tr>
<tr>
<td></td>
<td>I: 211</td>
<td>77/563</td>
<td>74/625</td>
<td>52/626</td>
<td>81/579</td>
<td>I 126/830</td>
</tr>
<tr>
<td></td>
<td>P: 485</td>
<td>202/540</td>
<td>180/466</td>
<td>177/511</td>
<td>183/491</td>
<td>P 95/197</td>
</tr>
<tr>
<td></td>
<td>sum: 2872/2872</td>
<td>913/2872</td>
<td>851/2872</td>
<td>810/2872</td>
<td>898/2872</td>
<td>819/2872</td>
</tr>
<tr>
<td>noisy</td>
<td>M: 1297</td>
<td>473/2520</td>
<td>445/2570</td>
<td>327/2469</td>
<td>301/2576</td>
<td>483/2198</td>
</tr>
<tr>
<td></td>
<td>A: 484</td>
<td>156/2423</td>
<td>145/2244</td>
<td>122/2062</td>
<td>113/2146</td>
<td>219/3328</td>
</tr>
<tr>
<td></td>
<td>G: 395</td>
<td>126/2459</td>
<td>125/2354</td>
<td>100/2450</td>
<td>86/2385</td>
<td>124/1755</td>
</tr>
<tr>
<td></td>
<td>I: 211</td>
<td>54/2527</td>
<td>48/2435</td>
<td>47/2466</td>
<td>57/2369</td>
<td>149/2903</td>
</tr>
<tr>
<td></td>
<td>P: 485</td>
<td>179/2400</td>
<td>185/2527</td>
<td>132/2342</td>
<td>129/2524</td>
<td>200/2145</td>
</tr>
<tr>
<td></td>
<td>sum: 2872/12329</td>
<td>989/12329</td>
<td>948/12329</td>
<td>728/12329</td>
<td>687/12329</td>
<td>1175/12329</td>
</tr>
</tbody>
</table>

The possible reason for the bad results on Music corpus is the extreme imbalance among classes, that is the class "Musician" takes up nearly half of terms. We found that SMBM-SW generates term clusters of different sizes. There are many true positives in the noisy corpus, but few in the clean corpus. SMBM-SW assumes that terms sharing the same context are more semantically related. It performs poorly on the clean corpus may due to the limited context in documents. When it comes to the noisy term, SMBM-SW can take the advantage of rich context information and achieve high performance.

Interestingly, those seeded LDA models produce the roughly equal size of clusters. However, the sizes of core concept classes are imbalanced in Music corpus, where the size of class "M" (Musician) is six times the size of class "I" (Instrument). If assigning these two core concepts ("M" and "I") with the same number of terms, it will produce a big false positive rate. Then, why the dominant core concept (which has a large number of terms) can not obtain a big cluster and the non-dominant one can not obtain a small cluster? We can deduce the possible reason from the nature of LDA. The LDA generates topics by distributing the relevant terms to a topic with the highest probability. However, the sum of the term probabilities of a topic should be equal to 1. Considering the topic of the dominant core concept, terms that belong to this topic should share
the total probability mass (i.e. 1). A term of this core concept gets merely a small probability under this topic, therefore, it is easily absorbed by other topics associated with the non-dominate core concept.

We experiment on the Music corpus to verify our guess. We assumed that allocating more topics for each core concept to increase the shared probability mass would be working. One way is increasing the topic number of each core concept with size (e.g. each of the five core concepts gets 2 topics when $K = 10$). However, the results on two versions of Music corpus (Figure 3) show that this strategy is not helpful and the performance goes lower as the $K$ grows. Another solution is using the number of topics in proportion to the number of terms of each core concept based on the domain expert’s knowledge. We checked only on the seeded LDA with different combinations of topic number. The result in Table 7 verifies our guess and the effectiveness of our second solution. The Seeded LDA gets similar recall (0.4051) in noisy Music corpus comparing with the best competitor SMBM-SW (0.4068). But the total number of topics $K$ should be reasonably small since the performance is not ideal with a large $K$. The datasets and all extra experimental results of parameter tuning are available at 5.

![Fig. 3. The performance (recall) of all seed based LDAs by increasing number of topics with equal number of topics for each core concept.](image)

![Table 7. Performance of Seeded LDA on music corpus with different combinations of topic number for each core concept](table)

<table>
<thead>
<tr>
<th>EXP</th>
<th>M(1297)</th>
<th>A(484)</th>
<th>G(395)</th>
<th>I(211)</th>
<th>P(485)</th>
<th>recall(clean)</th>
<th>recall(noisy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.3212</td>
<td>0.3461</td>
</tr>
<tr>
<td>exp1</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td><strong>0.4143</strong></td>
<td><strong>0.4051</strong></td>
</tr>
<tr>
<td>exp2</td>
<td>13</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>0.3824</td>
<td>0.3903</td>
</tr>
<tr>
<td>exp3</td>
<td>26</td>
<td>10</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>0.3901</td>
<td>0.3572</td>
</tr>
</tbody>
</table>

7 Conclusion

To summarize, we have proposed a new semi-supervised clustering method based on LDA variants toward ontology learning. We tackle two problems of existing

5 Details of parameter tuning and datasets of this paper [https://github.com/jason-huanghao/Seed-Knowledge-based-LDA-towards-Ontology-Learning](https://github.com/jason-huanghao/Seed-Knowledge-based-LDA-towards-Ontology-Learning)
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. Among all these LDA variants, we recommend using the seeded LDA for its higher performance and higher stability. Consider the seed information, we suggest using the synonyms and hyponyms of the core concepts. To achieve an ideal performance, we propose to use at least six seed words for each core concept. For future works, noisy terms (non-domain terms) elimination would be an important direction that is not considered in this work. Another problem is the term imbalance of core concepts and it commonly exists in many other domains. In this situation, we would recommend using our SMBM-SW. However, one can implement the Seeded LDA with a flexible number of topics for each core concept when dealing with different datasets.

References