Enchancing LDA for Ontology Learning

Ziwei Xu

UNIVERSITE BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE NANTES

THÈSE DE DOCTORAT DE

L'UNIVERSITE DE NANTES

Ecole Doctorale N°601
Mathèmatique et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*
Par

« Ziwei XU »

« Enhancing LDA for Ontology Learning »

Thèse présentée et soutenue à NANTES , le Le 3 juin 2021
Unité de recherche : LS2N - UFR Sciences et techniques

Rapporteurs avant soutenance :

Hedi KARRAY Maitre de conférences-HdR, INP-ENIT, Tarbes, Université de Toulouse
Julien VELCIN Professeur des universités, Université de Lyon 2

Composition du jury :
Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition ne comprend que les membres présents

Président : Giuseppe BERIO

Examinateurs :	Giuseppe BERIO	Professeur des universitPrés, UniversitPré de Bretagne sud, Vannes
	Ryutaro ICHISE	Associate Professor, National Institute of Informatics, Tokyo, Japan
Dir. de thèse :	Fabrice GUILLET	Professeur des universités, Polytech Nantes, Université de Nantes
Co-dir. de thèse :	Mounira HARZALLAH	Maitre de conférences-HdR, IUT QLIO, Université de Nantes

Invité(s)

ACKNOWLEDGEMENT

This thesis was accomplished thanks to the financing of Université de Nantes. Obviously none of this would have been possible without the diligent advising of Fabrice Guillet and Mounira Harzallah. Working with them was very encouraging, and they taught me to learn about new knowledge patiently, analyze the problem logically and broaden my horizons with respect to what constitutes a machine learning problem.

My overseas internship working with Ryutaro Ichise at the National Institute of Informatics of Japan was a great experience where I learned a lot about how research accomplishes its possibilities in various directions.

My parents and family have always been supportive regarding my ambitions of researching, and for that, I am incredibly grateful. Moreover, with the following names cloud, I want to thank all the people, who in one way or another, made this adventure of my doctoral studies a wonderful and enriching experience.

Finally, I would like to express many thanks to myself. The roads you chose are always not easy, during the struggling, fortunately, I had more chances to communicate with you, understand you and take care of you. I wish you could still be an enthusiastic, kind-hearted, and brave person available to your family, to your friends, to your colleagues, and to society. To infinity and beyond !

TABLE OF CONTENTS

Introduction 15

1 Ontology Construction and Learning 23
- 1.1 Ontology . . . 23
  - 1.1.1 Core Ontology . . . 24
  - 1.1.2 Modular Ontology . . . 25
- 1.2 Ontology Construction From Text . . . 26
  - 1.2.1 Ontology Construction Approaches . . . 26
  - 1.2.2 Ontology Construction With/Without Prior Knowledge . . . 29
- 1.3 Ontology Learning From Text . . . 31
  - 1.3.1 Linguistics-Based Techniques . . . 33
  - 1.3.2 Statistics-Based Techniques . . . 36
  - 1.3.3 Summary . . . 38

I Contribution : Framework for Term Clustering as a Task towards Ontology Learning 39

2 Related Works about Term Clustering 40
- 2.1 Term Clustering . . . 41
- 2.2 Term Representation Techniques . . . 41
  - 2.2.1 Term Selection . . . 42
  - 2.2.2 Feature Space Construction . . . 42
- 2.3 The Clustering Algorithms . . . 46
  - 2.3.1 K-Means . . . 46
  - 2.3.2 K-Medoids . . . 47
  - 2.3.3 Affinity Propagation . . . 47
  - 2.3.4 DBscan . . . 48
  - 2.3.5 Co-clustering . . . 49
- 2.4 Evaluation Indices . . . 51

2.5 Clustering based Approaches for Ontology Learning 58
2.6 Summary 59

3 The proposal Framework for Term Clustering towards Ontology Learning and its deployment 61
3.1 The Proposal Framework of Term Clustering 62
3.1.1 Corpus Pre-processing 63
3.1.2 The NPs Representations 64
3.2 The LDA-based Term Clustering Strategy 68
3.2.1 Term Cluster Formation 68
3.2.2 Term Cluster Thinning 71
3.3 Experiment 74
3.3.1 Corpus Preprocessing 74
3.3.2 Gold Standard 75
3.3.3 Experiment Settings 82
3.4 Evaluation of clustering strategies 84
3.4.1 Influence of clusters numbers 85
3.4.2 The optimal cluster numbers 87
3.4.3 The comparison between the classic strategy of clustering framework and the LDA-based clustering strategy 88
3.5 Summary 93

II Contribution : Semi-supervised Modular Ontology Learning with Topic Modeling Driven by Core Concepts 97

4 Related Works about LDA 98
4.1 Specific Notation and Terminology 98
4.2 The overview of topic models 100
4.3 The typical topic models 101
4.3.1 Simple Statistical Models 101
4.3.2 Simple latent variable models 103
4.3.3 Latent Dirichlet Allocation 104
4.4 The extensive LDA models with seed information 108
4.4.1 z-labels LDA 109

4.4.2 Seeded LDA . . . 112
4.5 LDA based approaches for ontology learning . . . 115
4.6 Evaluation Metrics for Topic Models . . . 120
4.6.1 Model perplexity . . . 120
4.6.2 Topic coherence . . . 121
4.6.3 Metrics for Topical Terms . . . 122

5 LDA driven by core concepts for term clustering 125
5.1 Our proposal : LDA guided by core concepts and their hyponyms . . . 126
5.1.1 Prior knowledge embedding . . . 127
5.1.2 Taxonomy relation discovery . . . 129
5.1.3 Towards modular ontology . . . 134
5.2 The Influencing Factors of our Proposal . . . 135
5.2.1 Composition of the reconstituted Corpus . . . 135
5.2.2 Corpus Filtering . . . 137
5.3 Experiments . . . 140
5.3.1 Parameter setting . . . 140
5.3.2 Random effect . . . 142
5.3.3 Model selection . . . 144
5.4 Evaluation : Results, Analysis and Comparison . . . 145
5.4.1 Results and Analysis . . . 146
5.4.2 Comparison . . . 149
5.4.3 Ontology Visualization . . . 151
5.5 Summary . . . 151

6 Conclusion 155

Conclusion 155
6.1 Contributions . . . 155
6.2 Perspectives . . . 157

7 Appendix 158

Appendix 158
7.1 The Top-N of Partition Size . . . 158
7.2 The Comparison between Two Clustering Strategies on CS corpus . . . 162

7.3 Human Evaluation 165
7.4 Term Partition Evaluation 166
7.5 The results of hyponym acquisition in DBpedia 169
7.6 The example of Wikidata-Taxonomy 170
7.7 The partial results of 'C6A4' term clusters. 171
7.8 The methods of selecting seed words for topics. 172

Bibliography **173**

LIST OF FIGURES

1 The overview of the contributions in this thesis. 18

1.1 a core ontology of the music domain and its core concepts. 25
1.2 An excerpt of a modular music domain ontology. 25
1.3 Ontology Construction Layer Cake. 27
1.4 The practical procedure to build an ontology with seed knowledge, extract from *Medelyan, Witten, Divoli et al. 2013* [29] 29
1.5 An overview of ontology Learning. 32
1.6 A sentence is parsed by the displaCy Dependency Visualizer. 35

2.1 An example of co-occurrence term representation. 43
2.2 An example of term clusters. 55

3.1 The framework of term clustering for ontology building. 62
3.2 The instantiated co-occurrence couples extraction. Extracted from Xu et al. [140] 63
3.3 The merged co-occurrenc matrix. Extracted from Xu et al. [140] 65
3.4 The distinction of the maximum term probability in each topic cluster. . . 70
3.5 The distinction of the normalized maximum term probability in each topic cluster. 70
3.6 The 2D visualization of the clustering for all NPs by T-SNE. *Notes : the core concepts of the Computer Science corpus are positioned as the red dots.* 72
3.7 The 2D visualization of the clustering for the residual NPs by T-SNE. *Notes : the core concepts of the Computer Science corpus are positioned as the red dots.* 72
3.8 The heat map between the practical Gold Standard and the labels of Annotator1. 78
3.9 The performance of clustering strategies with the increasing of cluster numbers (Reuter corpus). 86

3.10 The performance of clustering strategies with the increasing of cluster numbers (CS corpus). 86
3.11 The silhouette width of the two clustering strategies(Reuter corpus) . . . 90
3.12 The dunn score of the two clustering strategies(Reuter corpus) 90
3.13 The macro precision of the two clustering strategies(Reuter corpus) . . 91
3.14 The micro precision of the two clustering strategies(Reuter corpus) . . . 91
3.15 The asymmetric rand score of the two clustering strategies(Reuter corpus) 92

4.1 The matrix resolving of word document distribution. 104
4.2 The directed graphical model of LDA. 106
4.3 An example of the LDA generative process extracted from *Andrzejewski, Craven et Zhu 2010* [158]. 107
4.4 The directed graphical model of LDA. 110
4.5 The graphical model of z-labels LDA. 110
4.6 The graphical model of topic-term distribution part of Seeded LDA. . . . 113
4.7 The graphical model of document-topic distribution part of Seeded LDA. 113

5.1 The modular ontology construction with topic clusters. 134
5.2 The procedures to extract NPs with high topical significance in the modle of twice trained LDA [188]. 136
5.3 The comparison of the top-10 terms of TF and DF regarding to their maximum TF-IDF values. 138
5.4 The number of 'common terms' that exceeds the different ratios of the lower bound. 138
5.5 The Macro, Micro and Pairwise evaluation results for different cases and approaches. 148
5.6 The micro precision value of z-label model. 152
5.7 The micro precision value of the variants of seeded model. 152
5.8 The overview of the resulting ontology. 153
5.9 The zoom-in of the resulting ontology. 153

7.1 The distinction of the **top-10** normalized maximum term probability in each topic partition. 158
7.2 The distinction of the **top-20** normalized maximum term probability in each topic partition. 159

7.3 The distinction of the **top-30** normalized maximum term probability in each topic partition. 159
7.4 The distinction of the **top-40** normalized maximum term probability in each topic partition. 160
7.5 The distinction of the **top-50** normalized maximum term probability in each topic partition. 160
7.6 The number of terms exceeding the threshold on top-50 probability in each topic partition. 161
7.7 The **silhouette width** of the two clustering strategies(CS corpus) 162
7.8 The **dunn score** of the two clustering strategies(CS corpus) 162
7.9 The **macro precision** of the two clustering strategies(CS corpus) 163
7.10 The **micro precision** of the two clustering strategies(CS corpus) 163
7.11 The **asymmetric rand score** of the two clustering strategies(CS corpus) 164
7.12 A task explanation document for volunteers, which describes the annotation tasks detailed in Section 3.3.2. 165
7.13 The partition-oriented evaluation of 10 cases (including twice trained LDA model) in **silhouette width**. 166
7.14 The group-oriented evaluation of 10 cases (including twice trained LDA model) in **purity score**. 167
7.15 The group-oriented evaluation of 10 cases (including twice trained LDA model) in **asymmetric rand index**. 167
7.16 The group-oriented evaluation of 10 cases (including twice trained LDA model) in **Matthew correlation coefficient**. 168
7.17 The group-oriented evaluation of 10 cases (including twice trained LDA model) in **adjusted mutual information**. 168
7.18 The outputs of SPARQL query in DBpedia. 169
7.19 The partial taxonomy of 'Machine Learning' by Wikidata-Taxonomy tool 170
7.20 The partial term clusters of 'C6A4' case. 171
7.21 An example of selecting the seed words regarding different number of topics. 172

LIST OF TABLES

1.1 The snippet example extracted from the BBC Earth. 35

2.1 The co-occurrence techniques of term representation. 44
2.2 The condensed term representation. 44
2.3 The summation of the clustering algorithms on text. 50

3.1 The corpus size and statistics. 75
3.2 The experimental recordings of the annotators. 76
3.3 An example to calculate cohen's kappa. 80
3.4 The agreement degree between human annotators. A1, A2 and A3 is the identity for different annotators. 80
3.5 The Gold Standard and keywords of the whole corpus. 81
3.6 The practical library parameters of the typical clustering algorithms. . . 83
3.7 The size of clusters in different clustering strategies. 84
3.8 The optimal number of clusters regarding to the combined word representations and clustering algorithms 88
3.9 The deviation for different clustering strategies 94

4.1 The notions of topic models. 99
4.2 The comparison table of topic models, adapted from *Barde et Bainwad 2017* [169] 116
4.3 The summary of LDA based works for ontology learning 117

5.1 The diverse approaches to employ prior knowledge in different cases. . 128
5.2 The discovered hyponyms of head terms 130
5.3 An example for hyponym acquisition in DBpedia. 131
5.4 The extracted hyponyms and its appearance in corpus 133
5.5 The different cases of corpus reconstitution. 137
5.6 The statistics of filtering of computer science corpus. 140

5.7 The silhouette width on topic term clusters with different parameters of LDA in Computer Science corpus. 143
5.8 The random effect of LDA model. 144
5.9 The evaluation of five different metrics on the re-constituted corpus. . . 146

LIST OF ABBREVIATIONS AND NOTATIONS

Categories	Abbreviations	Meaning
Machine Learning Techniques	**eps**	the radius of a neighborhood concerning some points
	minPts	the minimum number of points required to form a dense region
	AP	Affinity Propagation
	DBscan	density-Based spatial clustering of applications with noise
	LDA as clustering	the novel strategy of term clustering based on LDA
	MCMC	Markov Chain Monte Carlo
	NMF	non-Negative Matrix Factorization
	SVD	Singular Value Decomposition
Evaluation Metrics	**AMI**	Adjusted Mutual Information score
	ARI	Adjusted Rand Index
	MCC	Matthews Correlation Coefficient
	PMI	Pointwise Mutual Information
	RI	Rand Index
Topic Models	**BigARTM**	Additive Regularization of Topic Models
	DTM	Dynamic Topic Models
	LDA	Latent Dirichlet Allocation
	LSA	Latent Semantic Analysis
	LSI	Latent Semantic Indexing
	pLSI	probabilistic Latent Semantic Indexing

Continued on next page

TABLE 1 – continued from previous page

Categories	Abbreviations	Meaning
	TF-IDF	Term Frequency-Inverse Document Frequency
	TMT	Stanford topic modeling toolbox
	TOT	Topic Over Time
Tools	**GATE**	General Architecture for Text Engineering
	MALLET	The MAchine Learning for LanguagE Toolkit
	NELL	Never-Ending Language Learning
	NLTK	Natural Language Toolkit
	YAGO	Yet Another Great Ontology
Concepts (terms)	**hypernym**	a superordinate, whose semantic field is broader than that of a hyponym
	hyponym	a term whose semantic field is more specific than its hypernym
	core concept	a term that represents the concept of a key sub-domain
	keyword	a term given by the corpus that associates to documents
	seed term	a term that we know its label information and use it in the starting of training
Concepts (general)	**BOW**	Bag-Of-Word
	RDF	Resource Description Framework
	GS	Gold Standard
	KB	Knowledge Base
	KG	Knowledge Graph
Concepts (NLP language)	**dobj**	direct object
	NLP	Natural Language Processing
	NP	Noun Phrase
	nsubj	subject

Continued on next page

TABLE 1 – continued from previous page

Categories	Abbreviations	Meaning
	nsubjpass	passive subject
	POS	Part-Of-Speech
	VPC	Verb Preposition Composition
Concept (self-defined)	**AllSum**	calculated by dividing the sum of all values
	CCpartition	once a core concept is contained within a thematic partition, it is convinced to label this partition by the involved core concept directly
	CS corpus	Computer Science corpus
	DiagnalSum	calculated by dividing the sum of diagonal values
	non-CCpartitions	all the other partitions without prior knowledge
	subCCpartition	if we have identified the hyponyms of core concepts inside a thematic partition, this thematic partition could be labeled by such a related core concept
Feature Representations	NPs_lda	the word topic representations
	NPs_VPCs_nmf	the NMF co-occurrence representations
	NPs_VPCs_tfidf	the weighted co-occurrence representations
	NPs_VPCs	the basic co-occurrence representations
	NPs_w2v	the word embedding representations

INTRODUCTION

> Un voyage de mille lieues commence toujours par un premier pas ; A journey of a thousand miles begins with a single step ; 千里之行，始於足下。
> — Laozi, *Tao Te Ching*, Verse 64, the 4th century BC

Research Context

With the rapid development of technologies, a mass of distributed and dynamic information result in difficulties of information interpretation. Knowledge representation turns to be crucial in managing the information into knowledge, which helps decrease the cost of searching and increase the possibilities of knowledge usage.

To organize the knowledge, *Cimiano 2006* [1] mentioned that, "the foremost procedure is to represent the knowledge of massive information by giving information a well-defined meaning". So that the knowledge can be processed by machines and exchanged between different parties in a semantically well-founded way, i.e. an ontology. *Guarino, Oberle et Staab 2009* [2] specified that "an ontology is an advanced knowledge representation technique that provides shared concept formations of a domain and connects them by their relations with the corresponding commitments to the logical theory". If an ontology is developed as a set of small modules and later composed to form a complete ontology, this ontology turns to be a modular ontology, which is easy to understand, extend and reuse. Each module could be derived by a predefined core concept of a domain.

In computer science, learning an ontology is inherently multidisciplinary with regard to artificial intelligence, i.e., machine learning, natural language processing (NLP), semantic web, data mining, knowledge representation, philosophy, etc. *Buitelaar, Cimiano et Magnini 2005* [3] indicated that "ontology learning usually refers to the processes of defining and instantiating a knowledge base with the (semi-)automatic support in ontology development". Especially, learning ontology from textual collections could be simplified into learning concepts and relations from text. In the text documents, the context

could support terms to learn their linguistic information. According to Harris' distribution theory [4], "Terms sharing the same context would have a higher probability of holding a similar semantic meaning, even further to become synonyms of each other". The synonyms or similar terms, who represent the same concept, could be gathered and assigned a label to represent a concrete concept (namely concept formation). In addition, the relation between concepts could be extracted or inferred from the originated text, or even acquired from the external knowledge bases.

In the state of the art of ontology learning, the synonym identification task and concept formation task could be performed by the use of statistical and/or probabilistic based methods, e.g. clustering [5] and topic models [6], while the relations discovery tasks could be obtained by the use of linguistic-based methods, e.g. POS(Part of Speech) tagging, noun modifier relationships [7] and seed words [8]. However, any single type of method falls short of exploring the sufficient semantic features for ontology learning. The hybrid methods were proposed in most cases but applied in a subjective and case-dependent scenario. In short, the proposed hybrid methods lack a pivotal role to connect the sub-tasks throughout the learning of an ontology in a general way.

Goal

To pervade large and unstructured collections of text documents, topic modeling algorithms play significant roles in helping machines interpret text documents. The Latent Dirichlet Allocation (LDA) [9] is one of the typical topic model techniques. We anticipate enhancing the LDA model to provide a reliable estimate about terms' semantic identity so that the terms could be clustered for the purpose of synonym identification. Moreover, we would like to explore LDA's utilities in benefiting from the prior knowledge embedding techniques, which could guarantee that term clusters are close to the predefined core concepts of an ontology. This procedure aims to achieve the concept formation task in the process of ontology learning. Besides, the taxonomic relations between terms would be discovered to construct the main structure of the ontology, internally from the linguistic features and externally from the knowledge bases.

In this manner, the topic model plays a pivotal role to tackle text information for the clustering purpose, which effectively associates the sub-tasks towards ontology learning.

Challenges

Topic modeling is a technique that comes with many algorithms that reveal, discover, and annotate the thematic structure of the collection of documents and of the occurred terms [6]. The topic model could act as a bridge to connect the plain corpus to knowledge representation (semi-)automatically, by clustering together similar terms with topical identity. However, making use of the topic model for clustering purpose is a challenging work for three reasons :

I Although a cluster of similar terms prone to convey the same concept. It is difficult to determine which kind of clustering strategy conforms to the anticipated concepts. Since the different clustering strategies take advantage of the different linguistic features respectively, e.g., statistic features, syntactic features, or semantic features.

II The raw documents lack semantic annotations, external links, and structural information, but only contain enormous plain text depicting a certain domain. It becomes rather complicated to resolve the text corpus into expressive elements. Accordingly, we divide the issue into three sub-tasks :

— How to use topic models for the synonym identification and the concept formation task ? The term clustering strategy could be employed to cluster terms by topics. A topic works as the notion but not the intuitive concrete terms. To break this constraint, each topic will be represented by a cluster of salient terms, which have both statistical and linguistic significance of topics.

— What are the possible operations to improve the utilities of topic clusters for ontology learning ? Only the term clusters that are close to the core concepts are helpful for the concept formation task. The operations might include corpus pre-processing, prior knowledge embedding, or even the adjustments over model training procedure.

— How to match the numerous topics into the limited core concepts of an ontology ? The favorable situation is to ensure that each topic cluster could correctly correspond to only one core concept. Once the topics could be instantiated as term clusters, this sub-task could be regarded to interpret the term clusters with limited terms(e.g. core concepts or the seeded terms related to core concepts).

III After the concept formation task for the clustered terms, we have to learn more

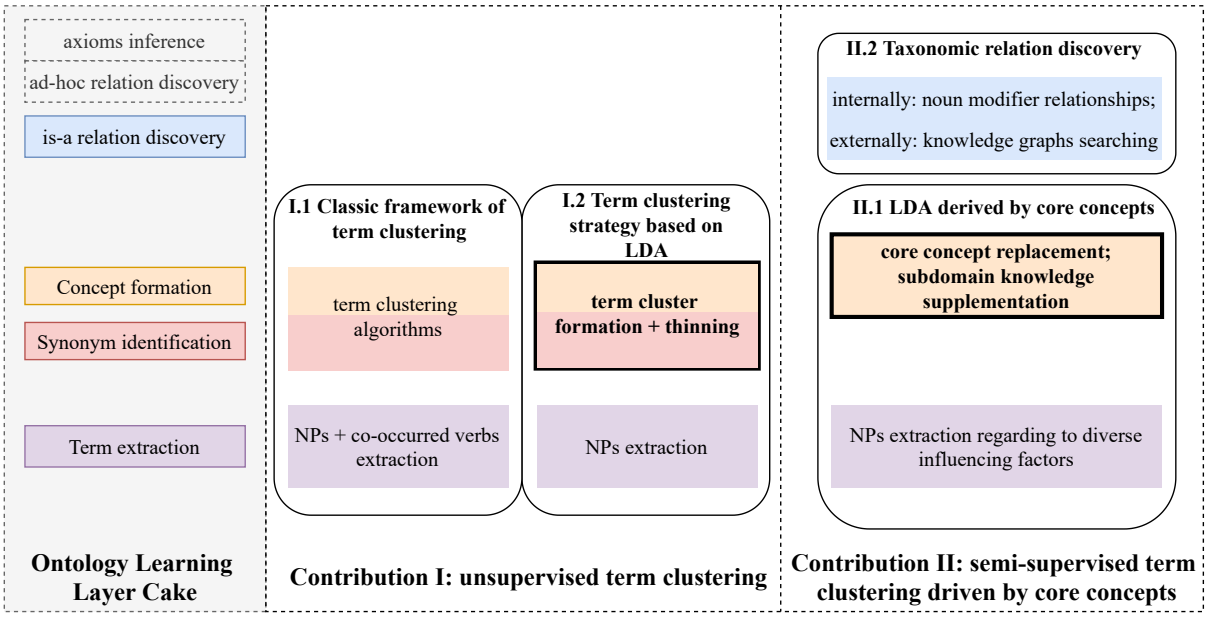


FIGURE 1 – The overview of the contributions in this thesis.

relations from the corpus. Except to discover the text's intrinsic relations, it is possible to take advantage of the common knowledge bases to organize terms with taxonomic relations. However, it is a complicated task to embed the external relations in the most valuable manner. There are many possibilities for embedding practices : in the phase of corpus reinforcement, in the semi-supervised training of the topic model, or in the direct application of taxonomy on ontology construction.

Contributions

To avoid hard and expensive manual efforts to label terms required for ontology construction, we resort to applying the clustering strategies for synonym identification tasks. For the moment, it lacks the systematic work to critically study the possibilities of applying topic models to cluster terms for the purpose of ontology learning. We draw the Figure 1 to show overall contributions regarding the ontology learning layer cake procedures [3]. The first contribution shows how to use the term clustering framework to learn ontology automatically :

- We draw **a framework of classic term clustering strategy** to show the prac-

tical approaches of automatically clustering similar terms. It describes the required components for term clustering and provides guidance about how to combine the feature representations with the clustering algorithms as a way to achieve better term clusters.

- We propose **a term clustering strategy based on LDA**. It introduces the term clustering formation and thinning steps to solve many problems, e.g., the different term filtering techniques in different stages, the distinct impacts of parameters of LDA learning, and the various criteria to select terms into clusters. With these techniques, this proposal could capture the semantic features of the most salient terms to help aggregate the synonyms.

The classic clustering method is then compared to the term clustering strategy derived by the topic model. The result shows that the topic features has a better performance on the majority of clustering algorithms and the clustering strategy based on LDA reached the best presentation of term clusters. In brief, it shows that the clustered terms are capable of representing a comprehensive concept as a means towards ontology learning.

The second central theme of this thesis is to explore diverse approaches to optimize topic models' performance by embedding prior knowledge, so as to semi-supervise the terms clustering towards the core concepts. (The inclusion of the few labeled data makes it 'semi-supervised', it could work as the embedded prior knowledge in LDA, or the seed terms used in other extensive LDA) :

- We examine **the diverse employments of LDA with prior knowledge embedding**. There are many possible phases that affect LDA as a clustering strategy :
  - — We study the influence of terms' syntactic features during corpus reconstruction.
  - — We explore the different approaches to embed the prior knowledge into the reconstructed corpus and compare it to other extensive LDA models embedded with the seed information.
- We experiment to **acquire knowledge by internally using subcategories frames between NPs and externally exploring the common knowledge bases**. The taxonomic relations play the role of the backbone of the related structure. Regarding this :
  - — We introduce the structural approach (i.e. subcategorization frame) to discover the taxonomic relations between the residual terms, by taking into ac-

count the features of center terms in noun phrases.

— We also grasp the relations from the common knowledge bases to enrich the taxonomic structure in the ontology, by mapping the terms from the corpus to knowledge bases.

Thesis Structure

We start to introduce the fundamental notions of ontology constructions with instances and present the related works of ontology learning with specific targets in Chapter 1.

The first contribution relies on the exploration of the unsupervised term clustering strategies, which works for the synonym identification task of ontology learning. Chapter 2 lists out the related works of the essential components for term clustering, i.e. word representation techniques, the typical clustering algorithms, and the evaluation metrics for term clusters. Then in Chapter 3, we discuss feature selection and feature extraction techniques to construct feature space from the corpora. Based on those feature representations, we experiment term clustering with different classical clustering algorithms, e.g. K-Means, k-medoids, DBscan, affinity propagation, and co-clustering. Meanwhile, we also propose the term clustering strategy based on LDA. A comparison is made between these two different kinds of clustering strategies, with regard to generating the more meaningful term clusters.

The second contribution talks about the semi-supervised modular ontology learning with topic modeling driven by core concepts. It expands our vision to topic models' utilities for modular ontology learning. Chapter 4 shows the variations of topic models from a simple statistic model to the basic LDA model, and even to the extensions of the LDA model, which take advantage of seed information to acquire the desired topic features. In Chapter 5, we describe the adaptations of the basic LDA, in which we apply core concept replacement and sub-domain knowledge supplementation as the supportive information embedding techniques over the corpus. Except for the knowledge embedding techniques, we also study other impacts to a good performed LDA model, e.g., the syntactic roles of NPs, the inclusion of verbs occurring with NPs, and the number of LDA training times. According to those explorations, we discover what is the best manner to embed prior knowledge to LDA. Comparatively, we evaluate other extensive LDA models with seed information, i.e. z-label LDA [10] and seeded LDA [11]. Besides,

the taxonomic relations acquired from subcategories frames and external knowledge bases also contribute to modular ontology building jointly.

At the end of this thesis, a summary and the expectation of future works are given.

CHAPITRE 1

ONTOLOGY CONSTRUCTION AND LEARNING

This chapter starts by introducing what an ontology is and exemplifying what the ontology looks like by using two typical kinds of ontology (Section 1.1) : *core ontology* and *modular ontology*. In order to address the ontology construction problems, then we present the *ontology construction layer cake* in Section 1.2 by decomposing an ontology into the different components. Subsequently, we summarize the ontology construction works on two sides : with the consideration of prior knowledge and without the consideration of prior knowledge. Finally, we go into the techniques of *learning an ontology from text* in Section 1.3 and provide the concrete example to directly illustrate the utilities for the techniques.

1.1 Ontology

In philosophy, an ontology is a systematic account of existence, which deals with the nature and structure of "reality" [12]. For informatics, what 'exists' is that which can be represented ; for this reason, an ontology is considered a special kind of information object or computational artifact [2]. In information systems, when a domain's knowledge is represented in a declarative formalism, it ought to follow the rules of the universe of discourse, e.g., concepts, relations, functions, or other objects. For example, a system could be instantiated by a company with all its employees and their interrelationships. To represent this company, we organise its general beings into *concepts* and *relations*. When we focus on the human resource of a company, then *Person*, *Manager*, and *Researcher* might be relevant concepts, where the first holds *subsumption* relation to the latter two, and persons hold *cooperates_with* and *reports_to* relation between them. A concrete person is treated as an *instance* of its corresponding concept.

Given an example of this simplified representation, accordingly, we could go deep into the more prevalent definitions of ontology in computer science. Originally, *Gruber et al. 1993* [13] defined an ontology as an "explicit specification of a conceptualization." Based on Gruber's description, *Borst 1999* [14] defined this notion as a "formal specification of a shared conceptualization," in which the shared view is posed expecting that the knowledge becomes consensus rather than independent. Eventually, ontology can be expressed by merging these two definitions, giving rise to the following statement [15], "An ontology is a formal, explicit specification of a shared conceptualization."

To specify the conceptualization, there are many concerns according to the means of representation. First of all, we focus on one language in case of a communication gap and intentionally constrain such a language's interpretations. Then, we need to specify what such possible extensions are to explicitly specify the implicit mind of people. In this way, we can list those extensions in correspondence with selected stereotypical world states [2]. Lastly, to guarantee that the expressions are machine-readable, the formal language is distinguished from the natural language. Compared to the previous conceptualization procedure, in this step, ontology engineers would try to find out the explicit contents inside the conceptualization structure and depict its structure by the formal language.

One may argue that it is impossible to share the whole conceptualization of one individual with other individuals. What can be shared are approximations of conceptualizations based on the mutual agreement on the primitive terms in one domain. Consequently, an ontology formally specifies a domain structure under the limitation that the related users properly understand the primitive terms. The components of an ontology are terms, synonyms, concepts, relations, axioms, and instances. Once the basic primitives are well-chosen and axiomatized to be generally understood, the related ontology has high probabilities to support large-scale interoperability in the future [2]. Our work is started with the primitives of a domain to ensure that the ontology structure would be stable and interoperable.

1.1.1 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 a basic ontology composed only of the minimal concepts (i.e core concepts) and relations between them that allows defining the other

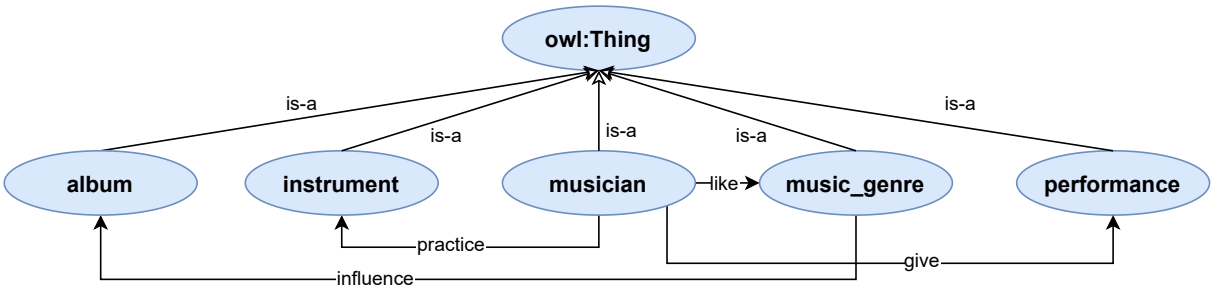


FIGURE 1.1 – a core ontology of the music domain and its core concepts.

concepts of the domain [16]-[18]. Scherp [19] considers that a core ontology should be characterized by a high degree of axiomatization and formal precision. Figure 1.1 shows a core ontology of the music domain.

Furthermore, in a core ontology, generally, each core concept refers to (conceptualizes) a sub-domain of the ontology domain, and it could be related by core relations to other core concepts (see Figure 1.1). A core ontology could be considered as an upper ontology (i.e., top-level ontology or foundation ontology [16]) of domain ontology, which provides the high possibilities to be reused for extensive purpose. Therefore in most cases, a core ontology is predefined by a domain expert, so as to provide guidelines in terms of domain ontology construction.

1.1.2 Modular Ontology

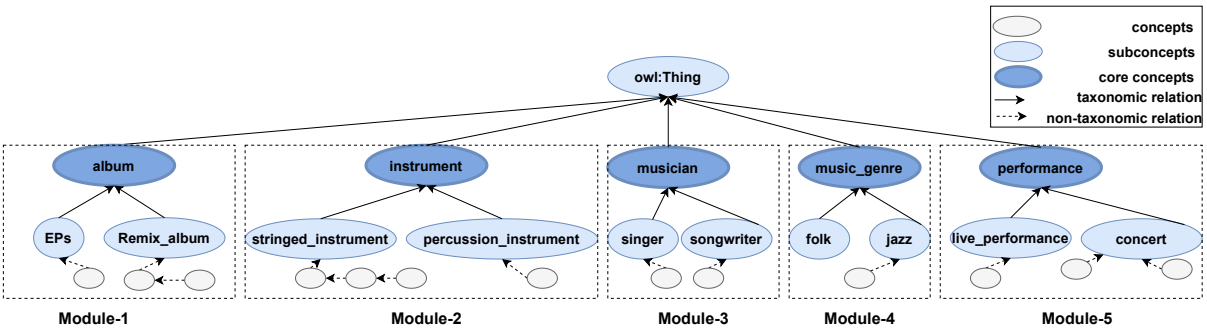


FIGURE 1.2 – An excerpt of a modular music domain ontology.

Modular ontology is considered a major topic to facilitate and simplify the ontology engineering process [20]. In the case that it is required to alter the structure of the ontology, we can remove, add, or enrich the target modules in modular ontology, without

interference to other remaining parts of ontology. For ontology experts, the modular representations are easier to understand, reason with, extend and reuse [21]. Thus it reduces the complexity of designing and facilitates ontology reasoning, development, and integration [22]. Actually, a modular ontology is inseparably intertwined with a core ontology. Based on core ontology, it is interesting to obtain a well-structured taxonomy where each sub-domain is defined by a separate module (Figure 1.2). It becomes easier to define a modular regarding each core concept that represents its sub-domain. In this manner, inside each modular, a core concept could be extended to its sub-concepts with the 'is-a' relations. (seeing the bottom layer in Figure 1.2).

In brief, even though building a complete ontology seems to be complicated to do, we are able to simplify it starting from a core ontology and ending with a modular domain ontology.

1.2 Ontology Construction From Text

Along with the development of digital communications, more and more textual snippets are generated, transmitted, and recorded through the network. The textual information is easier to be accessed and be summarized into the knowledge. Thus we will focus on how to construct ontology from the text.

1.2.1 Ontology Construction Approaches

Ontology construction from texts can be done manually or automatically. In both cases, it needs to respect the structure of the ontology. The construction can be performed following a top-down approach or a bottom-up approach. For the top-down approach, the basic starting point is to consider several core concepts of an ontology as the philosophical guidance for the prospective engineering artifact [23]. The resulting ontology could be seen as the knowledge representation with rich semantics in a specific domain [24], i.e. the CIDOC Conceptual Reference Model (CRM) [25] and CORA ontology [26].

It facilitates interoperability among ontologies, but is considered to be too abstract to understand. For the bottom-up approach, without starting with a blank slate, it reuses exiting information or knowledge in order to capture experience from a group of individuals. The resulting ontology is less structured but dynamically close to the application.

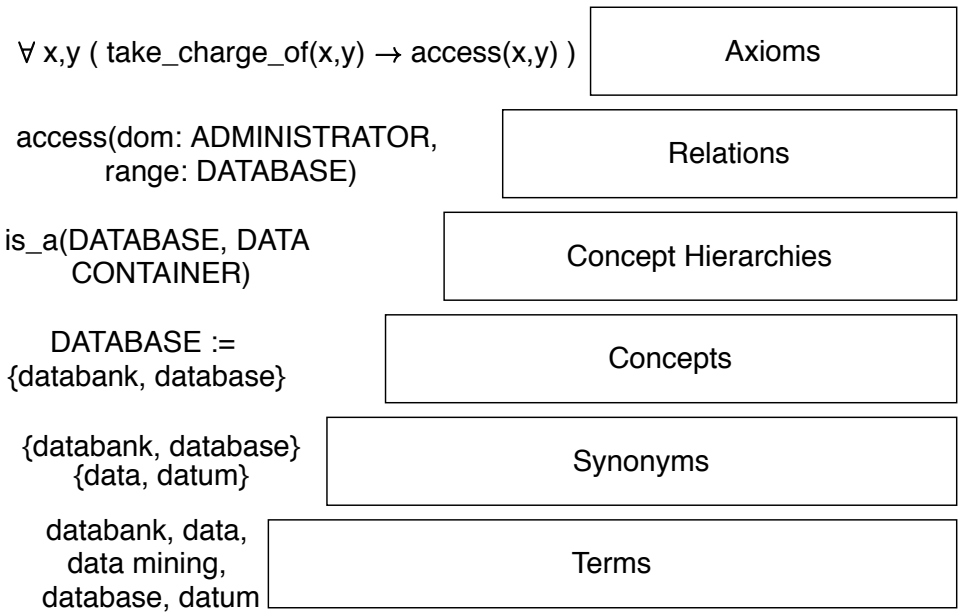


FIGURE 1.3 – Ontology Construction Layer Cake.

This approach is applied in a 'BRAINS' system of crime investigation tool for Dutch police [24].

As we have illustrated, the development team could follow their own set of principles, design criteria, and approaches in the ontology development process. However, the absence of structured guidelines causes some problems in the capabilities to share, extend and reuse ontologies. Until now, few domain-independent methodological approaches have been reported for building ontologies [27], where MethOntology [28] is one of the most representative for ontology construction. MethOntology is a well-structured methodology to build ontologies from scratch and prone to reuse the existing ontologies [28]. It seeks to build the ontology incrementally using a life cycle based on evolving prototypes. The methodology centers on the process of building conceptual models by defining concepts, organizing taxonomies, defining relations, defining concept axioms, and defining formulas. Also, it includes verification and validation of the ontology at the knowledge level.

To address ontology construction, we need to learn about the relevant subtasks, either manual or with any level of automatic support. Primarily, we center on the concepts and relations between them and the axioms used to refer to them.

More specifically, ontology construction is composed of 6 subtasks allowing the acquisition of ontology components. The latter are represented as a layer cake whose

subtask complexity is roughly increasing from the bottom up, as displayed in Figure 1.3, which is adapted from *Buitelaar, Cimiano et Magnini 2005* [3].

The right side of the figure represents the different ontology components acquired following the sub-tasks of ontology construction :

i) **Term** extraction is the first step of ontology construction in a bottom up approach.

ii) The acquisition of **synonyms** will gather the semantic similar terms with variant linguistic expressions in the text corpus. The extraction of concepts can be specified in three ways : instances, terms that refer to it, and the list of its properties. These ways assist to interpret or specify a concept. When we deal with ontology from texts, we consider the second way : a concept as set of terms that refer to it, which gives rise to concept formation.

iii) The is-a relations are extracted between concepts, which contribute to building the **concept hierarchies**.

iv) Except for the is-a relations, it exits further more **relations** (i.e. ad-hoc relations) to be discovered during the ontology construction process.

v) Eventually, the extraction of rules or **axioms** could address lexical entailments' problems after the formation of concepts and their relations.

On the left side of this figure, we have *databank, data, data mining, database, datum* as the extracted terms in the bottom layer. Then we recognize the two synonym groups which are shown into the independent braces. One synonym group could be associated with the concept *DATABASE*, also it possesses a taxonomic relation between the concept *DATABASE* with the concept *DATA CONTAINER* and a non-taxonomic relation namely *access* between the concept *ADMINISTRATOR* and the concept *DATABASE*. At the top level, an axiom could be induced in the case that, once *y* is *taken in charged* by *x*, then it is apparent that *x* is able to *access* to *y*.

In order to achieve the construction of each sub-task, many proven techniques have contributed to ontology construction, i.e. information retrieval, data mining, natural language processing, and knowledge representation. Section 1.3 will use an example to present the major combinations of those techniques, to provide a general overview of constructing an ontology following the layer cake steps.

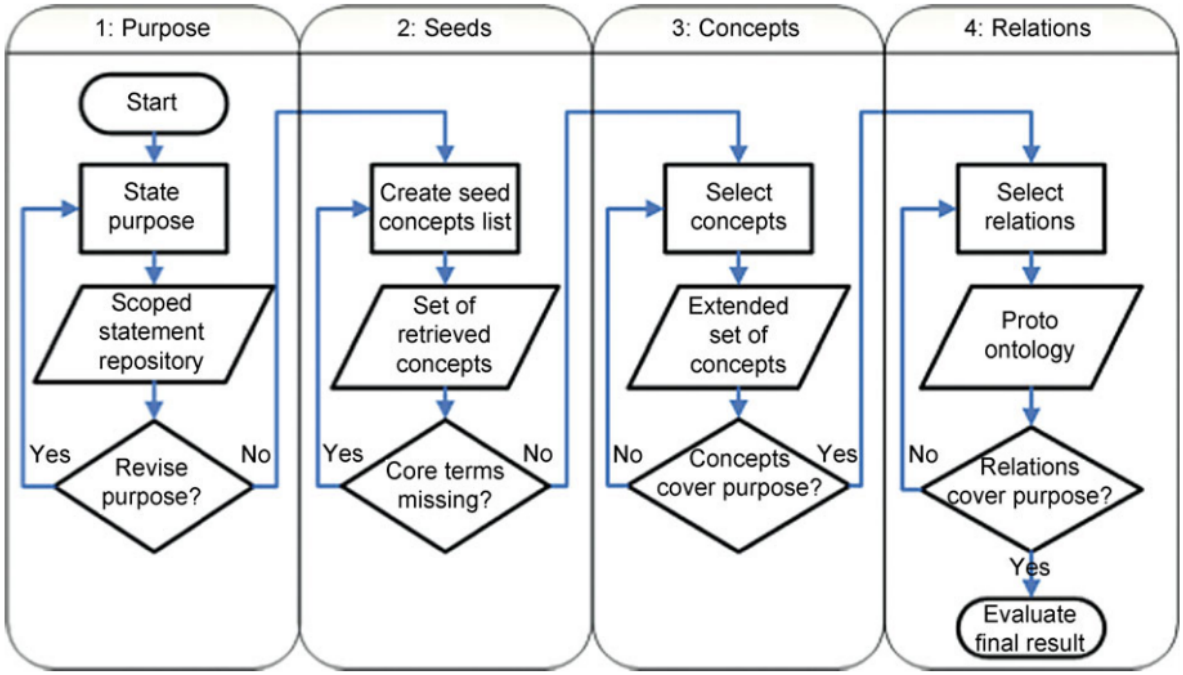


FIGURE 1.4 – The practical procedure to build an ontology with seed knowledge, extract from *Medelyan, Witten, Divoli et al. 2013* [29]

1.2.2 Ontology Construction With/Without Prior Knowledge

With Prior Knowledge

To build an ontology, the researchers are confronted with corpus selection, seed terms retrieval, concept formation, relation discovery, axioms induction, and instances definition.

Several works propose to reuse a core ontology to identify and further define the core concepts by specialization [16], [30], [31]. Based on the **Core Ontology** (Section 1.1.1), *Medelyan, Witten, Divoli et al. 2013* [29] propose a ontology construction approach, which extends from the application of core concepts into the seed concepts lists, seeing the guidance of Figure 1.4. The seed terms are the concrete expression of the core concepts, in which one core concept consists of several seed terms. Both of them convey the supervision information thus play the role as prior knowledge. The seed terms could be regarded as one significant kind of prior knowledge in the construction process.

On the one side, once the seed terms were determined as core concepts, either manually or automatically, the further terms could be identified by computing their co-

occurrence probability with the seed terms [32]. On the other hand, if the seed terms are determined randomly from the pool of content words, the terms that are close to seed terms could be clustered into semantic categories, using pattern analysis between seed terms and the core concepts [33]. *Gruber 1993* [30] suggests using the core ontology of a domain to build domain ontology, and *Gangemi, Catenacci, Ciaramita et al. 2006* [34] and *Kutz et Hois 2012* [35] agreed that mapping a core ontology to a domain ontology could improve the modularity of ontology. For instance, *Burita, Gardavsky et Vejlupek 2012* [16] map NEC (Network Enabled Capabilities) core ontology to the NEC domain ontology.

With seed terms as prior knowledge, an ontology could be learned into different sub-domains, which conforms to the structure of **Modular Ontology** (Section 1.1.2). *Besbes et Baazaoui-Zghal 2015* [36] defined different sub-domains developed from their core concepts (i.e. seed terms) and developed taxonomic relation and conceptual relation between terms within a partition ; in parallel, the topic feature from documents can also lead to the sub-domain representation of an ontology. *Mustapha, Aufaure, Zghal et al. 2012* [37] proposed the topic ontology where topic and relation definitions are specified in advance ; then, each topic would go deep to relate terms (i.e. entities) inside that partition.

However, it results in difficulties in tackling the problem of scalability in subdomains of ontology. In general, we notice the strong correlation between domain ontology and seed terms as the ontology construction approaches.

Without Prior Knowledge

Imagine an algorithm that can read large amounts of text and build ontologies based on the information itself, just as like people read books for knowledge. To achieve this, firstly it ought to identify the concepts of interests and then learn the facts and relationships associated with them. However, unlike the understanding process of the human, most of the ontology construction tasks are not starting from prior knowledge.

It exists bottom-up approaches for constructing ontology from unstructured text [38]. They identify concepts by detecting terms of interest and clustering them based on similarity measures in the feature space. Next, they use the document partitions to cluster co-occurring concepts and use them as the basis for the deeper relationship extraction.

This work combines the ideas of clustering techniques, and words document allo-

cation appearance. It raised the importance of clustering techniques and topic models for ontology construction. Besides, *Poon et Domingos 2010* [39] proposed to build a probabilistic ontology in a unified approach. To identify concepts and their relations, they used a semantic dependency parser to analyze the sentences. From this parser, they built a non-deterministic ontology by the logical forms of sentences.

For other facts extracted from webs, they operate on terms rather than a concept level. For example, the never-ending language learning (NELL) [40] utilize masses of unstructured text crawled from the Web to bootstrap the extraction of millions of facts. In this way, the massive occurrence of fact plays a significant role in finding the relations between terms and constructing ontology.

1.3 Ontology Learning From Text

In the last section, ontology construction can be done manually or automatically. Different from it, the techniques of ontology learning focus on automatically constructing an ontology. Over the past decade, many proven techniques have contributed to ontology learning progress from established fields, such as information retrieval, machine learning, data mining, natural language processing, and knowledge representation. Information retrieval provides diverse algorithms to analyze the associations between concepts in the text. Machine learning and data mining contribute to extract patterns out of massive datasets based on extensive statistical analysis in a supervised or unsupervised manner. Natural language processing provides many tools to deal with the source text for different linguistic purposes, e.g., morphology, syntax, and semantics, to uncover the concept representation and their relations from linguistic behaviors. Knowledge representation enables the ontological elements to be formally specified and represented, which stimulates knowledge integration with the existing knowledge bases.

From the perspective of techniques, they were first introduced to solve the specific tasks. However, in ontology learning, those techniques will finally combine and serve for the different stages. Generally, the techniques could be classified into linguistics-based, statistics-based, or hybrid, where the hybrid approach is mainly used in current research. Figure 1.5 presents the complex connections between the commonly used techniques and the different sub-tasks. In the "output layer", the components in blocks conform to the components of the ontology construction layer cake in Section 1.2. Unlike before, the components are connected with the chain relations from 'terms' to

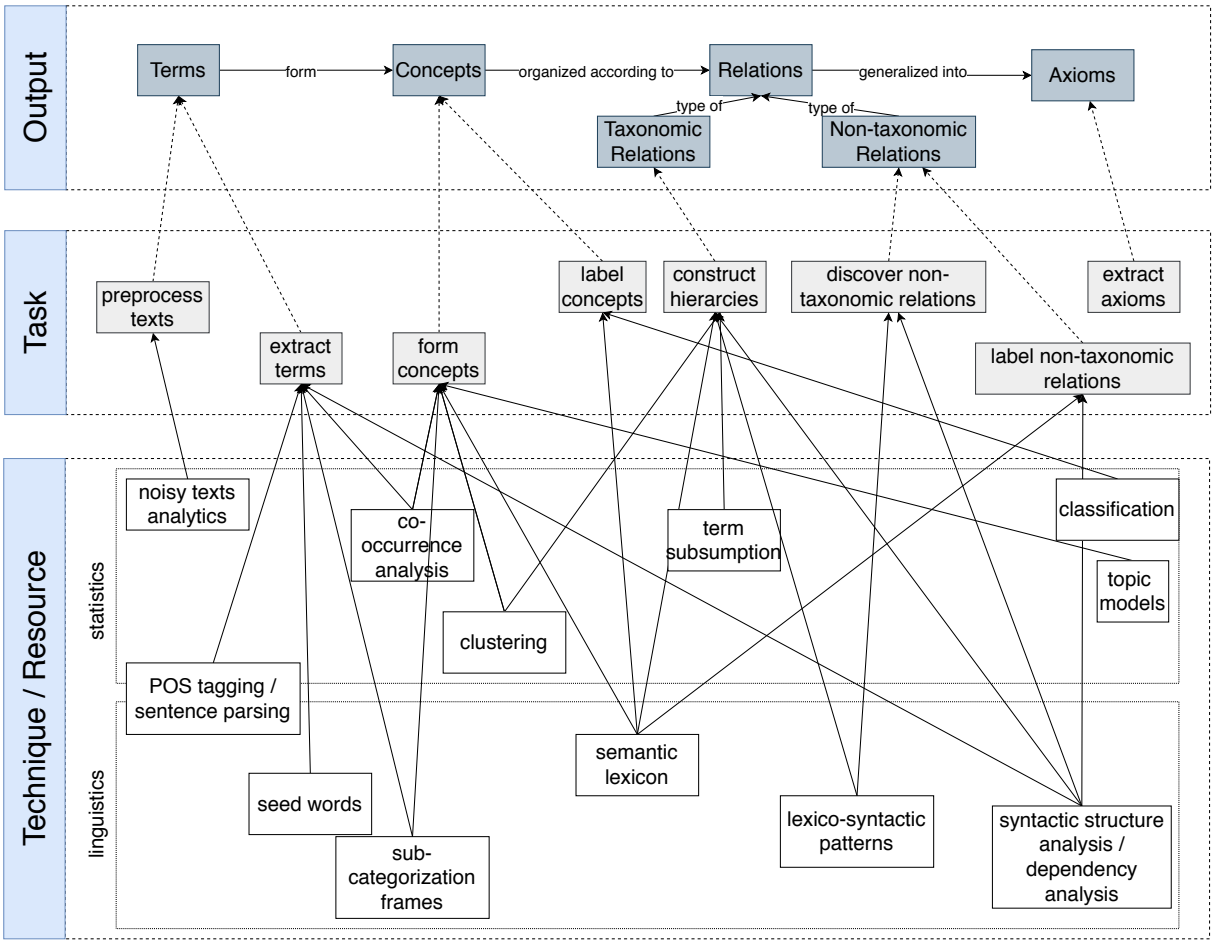


FIGURE 1.5 – An overview of ontology Learning.

Notes : this figure is re-drawed from the work of Wong, Liu et Bennamoun 2012 *[41].*

'axioms' and the 'relations' are divided into 'taxonomic relations' and 'taxonomic relations'. In the "task layer", it describes the corresponding tasks to the components of ontology. For example, pre-processing texts and extracting terms result in 'terms component', forming concepts and labeling concepts contribute to 'concepts components', and so on.

Before we proceed, I present a textual example to present the contents of ontology components from the usage of the ontology learning techniques. As the text shown in Table 1.1, this snippet talks with the relations between human and their pets, token out from an article [1] in BBC Earth. First of all, we narrow down our focus from any **terms** to only nouns or noun phrases, seeing the bold terms in Table 1.1.From the point of view of humans, it is not difficult to conclude that a pet is a **synonym** of domesticated animals. The main **concept** of this snippet is about pet and pet owner. We know that pet has the **is-a relation** with dogs, hamsters, and parakeets. Also, we can learn that *your furry love-bundle* holds the **relations**, i.e. *understand*, to *a joke*.

In the following sub-sections, we will use this example to present how the statistics-based and the linguistic-based techniques help to recognize the knowledge in the manner of human's interpretation.

1.3.1 Linguistics-Based Techniques

In the field of knowledge acquisition from the text, it is apparent that the functional entities of sentences and their clauses constitute the dominant linguistic elements for syntagmatic information collection. *Cimiano P. et Saitia 2004* [42] describes the local context by extracting triples of nouns, their syntactic roles, and co-occurred verbs. They consider only verb/object relations to emphasize partial features of terms working as an object by a conditional probability measure, which calculates the conditional probability that a certain term appears as head of a certain argument position of a verb. Similarly, *Jiang et Tan 2005* [43] and *Rios-Alvarado, Lopez-Arevalo et Sosa-Sosa 2013* [44] formed the triple term structure of noun phrases and verbs, in shape as a NP as subject, a verb, a NP as object.

Moreover, ASIUM [45] 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 :

1. https ://www.bbcearth.com/blog/ ?article=is-your-cat-laughing-at-you

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

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

$$< to\,travel > \quad < subject > \quad < \textbf{Bart} >$$
$$< by > \qquad < \textbf{boat} >$$

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 the preposition 'by'). Except for the extraction of nouns and verbs, some works consider the involvement of adjectives, which would be considered as keywords of ontology learning [46], [47].

With the assistance of the natural language processing tools, linguistics-based techniques (the linguistics box of Figure 1.5) can support the interpretation of the text :

- **Tagging and Parsing**. They are the basic and widely used method to uncover terms and relations in a sentence. As for instance, the first sentence is parsed with part-of-speech tags in Figure 1.6, which is generated by 'displaCy Dependency Visualizer' [2]. In Figure 1.6, the common nouns or noun phrases(NPs) are tagged as NOUN. While the proper noun has PRON tag. The single verbs *know* and *develop* are tagged as VERB, in which *develop with* is a verb with its preposition(VPC) for human. For the rest term *As*, it is an auxiliary of verb tagged as ADP. In addition, in order to study the syntactic structures of this sentence, in Figure 1.6, from the 'nsubj' arrows, we could learn that *any pet owner* is the subject of verb *know*, *you* is the subject of verb *develop*. The 'dobj' arrow indicates that, *a distinct emotional bond* is the direct object of *develop*. The 'pobj' arrows show that, *your animal companion* is the object of preposition *with* and *choice* is the object of preposition *of*. It is easy to conclude that the POS tagging techniques assist for **term extraction task**, i.e. noun or NPs recognition, and the syntactic parsing techniques support the ad-hoc relation discovery task.
- **Semantic Lexicons**. It provides easy access to the predefined concepts and relations of a large collection. For example, WordNet [48], a famous dataset for semantic lexicons, is capable of providing a set of similar words (i.e., synsets) to a concept from semantic lexicons. Except for synonyms, the lexicon

2. https ://explosion.ai/demos/displacy

As **any pet owner** will know, you develop **a distinct emotional bond** with **your animal companion** of **choice**. You chat with **the dog**, remonstrate with **the hamster** and tell **your parakeet secrets** you would never tell anyone else. And, while part of you suspects that **the whole endeavour** might be completely pointless, another part of you secretly hopes that somehow **your beloved pet** understands. But what, and how much, do **animals** understand? For instance, you know that **an animal** is capable of experiencing **pleasure**, but do they experience **humour**? Can **your furry love-bundle** understand **a joke** or stifle **a guffaw** when you drop **a heavy item** on **your toe**? Do **dogs** or **cats** or any animal laugh in the same way that we laugh?

TABLE 1.1 – The snippet example extracted from the BBC Earth.

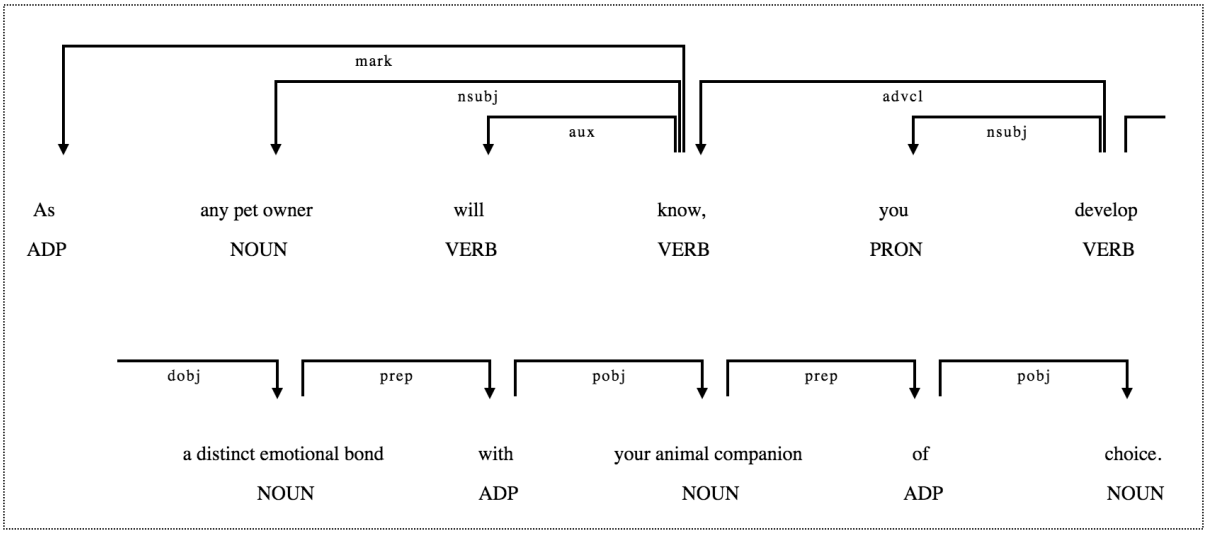


FIGURE 1.6 – A sentence is parsed by the displaCy Dependency Visualizer.

defines more associations, such as hypernym-hyponym (i.e., parent-child relation), meronym-holonym (i.e., part-whole relation), etc.

- **Lexico-Syntactic Patterns**. Some patterns are evident for hypernym relations, e.g., 'NP such as NP', 'NP,..., and NP', or meronym relations, e.g., 'NP is part of NP'. Based on these surface information in a sentence, the use of lexico-syntactic patterns has been proposed by *Hearst 1998* [49] to extract hypernyms and meronyms. Thus many works have been employed to generate more detailed rules and conditions automatically, rather than the hand-craft patterns from Hearst, to extract the taxonomic or non-taxonomic relations.
- **Noun Modifier Relationships**. Also for the **task of taxonomic relation recognition**, relatively, another approach noun modifier relationships [7] could be

applied to exploit the internal structure of noun phrases (i.e. 'emotional bond') to derive taxonomic relations between classes expressed by the head of the noun phrase and itself that can be derived from a combination of this headword (i.e. 'bond') and its prefixes (i.e. 'emotional'). From another point of view, it is prevalent to use the prior knowledge from the well-known knowledge bases to acquire the taxonomic relations between the terms in the corpus, which turns to be more convinced and knowledgeable compared to other methods.

- **Seed Words**. It is a common practice to use seed words to guide ontology learning. Since it assists in focusing on a particular domain and discover the relevant terms in the predefined direction.

1.3.2 Statistics-Based Techniques

In the early stages of ontology learning, statistical techniques are prevalent when the semantics and relations of text are considered. The key notion behind these techniques is that the lexical unit's co-occurrence indicates the existence of semantic identity among these related terms. Let's look at the statistics box in Figure 1.5 :

- **Topic Models**. The term frequency-inverse document frequency (TFIDF) [50] provides the relevance between documents and terms. Except for the document mutuality, the language modeling and probabilistic representation also uncover the specific relevance between terms. However, the raw data of the corpus cause the sparseness of term-document matrices. The topic model aims to regenerate the documents with words along with the predicted distribution. It introduces the topic to express terms instead of document representation and symbolize documents instead of terms' representation. In this manner, the symbolize of topics carries out the dimension-reduction effects to those sparse matrices. There are many typical topic models developed from simple ones to sophisticated ones, e.g. Latent Semantic Analysis (LSA)[51], Latent Semantic Indexing (LSI) [52], Latent Dirichlet Allocation (LDA) [6], et al. From front to back, those models turn to be more generative, less overfitting and easier for calculation. Furthermore, the inherent relations between terms can be revealed by analyzing their topic probabilities, leading to **concepts formation**.
- **Cooccurrence Analysis**. The purpose of the lexical unit's co-occurrence is to either extract related terms or form concepts of the close terms. The co-

occurrence form could be expressed as the co-occurrence of a sequence of words (i.e., a couple of noun phrases) beyond chance within a segment of a document. If we take advantage of the direct co-occurrence of terms, the joint presence of verbs will support to detail the meaning of nouns, additionally, the verbs could even serve as features for nouns. In this geometric feature space, the nouns co-occur with an identical verb in the corpus are prone to close to each other. If we take advantage of a larger window for co-occurrence terms, which means that two terms are regarded as co-occurred even though they are n-positions far from each other in the context, one term could be represented by the facts of its surrounding terms in the feature space. Accompanying with the clustering techniques, it is convinced that the clusters of terms are relevant and even lead to a certain degree of **synonym identification** [53]. To measure the association strength between the co-occurred terms, there are many popular measures, including dependency measures (i.e. mutual information [54]) and statistic measures (i.e. frequency on co-occurrence matrix).

- **Clustering**. Clustering aims to discover concepts or construct hierarchy by gathering terms together respecting their semantic similarities [55]. The clustering procedure is either to group the most related terms (i.e., agglomerative clustering) or to divide all terms into smaller aggregations to maximize the in-cluster similarity (i.e., divisive clustering). As a beginning, the word representation turns to be the main issue. For example, the features could be syntactically related (e.g., *Rabbit eat carrot*. a verb links two nouns) or semantic related (e.g., *King and Queen live in the castle*. King co-occurs with Queen in context). If we simply rely on the syntactic feature of verbs, the *Rabbit* and *carrot* might be clustered together, which seems to be less useful than semantically related terms for humans.
- **Classification**. The classification algorithm is widely used to train models. If the well-labeled dataset is available, the classification methods help form concepts and even provide labels for terms' aggregation.
- **Term Subsumption**. Conditioning to the documents that a term occurs, the conditional probabilities of this term are employed to discover the hierarchical relations between this term and other terms in documents [56], [57]. The subsumption measure is to quantify the extent of a term being more general than another term by calculating their joint and conditional probabilities.

1.3.3 Summary

Developing an ontology is comparable to a craft. It is largely left to the personal skills and preferences of the ontologist [58]. Meanwhile, it is difficult to highlight our interests in ontology building, specifically when we deal with a big corpus. This puzzle could be alleviated if the ontologist could start from and focus on the shared and foundational concepts. Therefore, we are interested in exploring the term clustering techniques for concept formation tasks with and without prior knowledge. With the addition of the taxonomic discovery techniques, the modular ontology would be constructed with the formed concepts respecting the predefined core concepts.

In the following chapters, we will start by studying and analyzing the detailed approaches related to term clustering techniques.

PREMIÈRE PARTIE

Contribution : Framework for Term Clustering as a Task towards Ontology Learning

CHAPITRE 2

RELATED WORKS ABOUT TERM CLUSTERING

This chapter is concerning with term clustering approaches for ontology learning. Harris hypothesis [59] suggests that terms sharing the same context tend to have a similar meaning. The supervised techniques usually use a training dataset to learn a model for predicting the classification of a term or semantic relation between a pair of terms, while the unsupervised techniques gather terms semantically close on the same cluster by either measure-based approaches or clustering-based approaches. Generally, the supervised approaches outperform non-supervised approaches [60]. However, unlike unsupervised approaches, the supervised approach requires extra effort on building a training dataset, which is not suitable for a big corpus. Therefore, rather than the classification approaches, the clustering approaches are widely applied to build ontology automatically.

In the process of ontology learning, we would achieve the goal of uncovering synonyms between terms. This step could be achieved by employing some clustering algorithms to assign terms into groups. The major issues of term clustering locate in the approaches of feature extraction, the solutions of the high-dimensional matrix, and the choice for clustering algorithms.

Accordingly, we study term clustering into four sections : in Section 2.2, we discuss the word representations in the three different aspects : linguistic, statistical, and hybrid features ; in Section 2.3, we introduce the classic clustering algorithms by specifying their drawbacks and weaknesses on clustering ; in Section 2.4, we divide the evaluation metrics into two categories regarding the presence or absence of gold standard ; finally, in Section 2.5, we investigate the related works that apply the clustering-based approaches for ontology learning.

2.1 Term Clustering

Clustering is the task of grouping a set of objects in such a way the objects in the same group are more similar to each other than to those in other groups [61]. To accomplish a clustering task, a targeted clustering algorithm is chosen in advance. The inputs of a clustering algorithm include the objects to cluster, their features, and/or the number of clusters, and the parameters of the selected algorithm. As a result, each object is assigned to a group.

To support the terminology's comprehension, we list the substituted terms in this thesis that refer to the same meaning :

- objects : individuals, observations, items
- groups : clusters, partitions, aggregations
- feature matrix : word representation, feature space, vector space

In particular, term clustering is to group a set of 'terms' based on the similarity of their features. Many surveys have been generated to discuss the term clustering techniques. Feldman and Sanger [62] presented a comprehensive discussion in text mining in 2006. Specifically, in chapter 5, they studied the advanced text techniques that support the clustering approaches and summarized the previous in-depth algorithms for text analysis.

With the efforts of more detailed studies on term clustering, Aggarwal and Zhai [63] published a conclusive book *Mining Text Data* in 2012. They introduced the systematical procedures of text mining, including information extraction from text, text summarization, text clustering and classification, dimensionality reduction, probabilistic models, et al. Especially, they conducted a survey of term clustering techniques in details in chapter 4 [64], which detailed the feature selection and transformation methods for text and explored the clustering algorithms on the distance and word orientations. With the development and refinement of the new technologies for text mining, Aggarwal wrote an enhanced book *Machine Learning From Text* [65], which supplemented and ameliorated the term clustering techniques regarding the needs of application scenario.

2.2 Term Representation Techniques

Word representations are found to be useful for measuring semantic similarity, and for solving proportional analogies [66]. Before going into depth of word representation

techniques, it is necessary to make clear which terms are interesting to be represented in the feature space. In other words, the preliminary goal is to extract a set of terms that are related to the main topics discussed in a given document [67]-[70] and present them in their related feature space. Thus we use 'term representation' in this section to be concrete. We present the operations typically into two steps : 1) explore the ways to extract the particular kind of words ; 2) study what kind of features are important and interesting for term representation.

In our expression, *terms* could be nouns, verbs, noun phrases (NPs), verb preposition combinations (VPCs) or other words ; *tokens* only refer to a single word ; *individuals* here are the terms that are selected and will be represented in the feature space.

2.2.1 Term Selection

The first step is designed to avoid trivial information and keep the number of individuals to a minimum in the representation task. The typical approaches include : 1) recognize words with certain part-of-speech tags or dependency parsing (e.g. nouns, verbs) [71], [72] ; 2) extract n-grams that also appear in Wikipedia article titles [73] ; 3) extract noun phrases (NPs) [74] or other phrases that satisfy pre-defined lexico-syntactic patterns [75].

For a sentence from the text, the nouns or noun phrases (NPs) are worth to be highlighted because they cover most of the descriptive information of this sentence. At the same time, the components of the context of NPs, i.e., verbs or verb preposition compositions (VPCs), could also present the concrete connection between NPs. To be specific, a noun phrase (NP) is a phrase that has a noun as its head or performs the same grammatical function as a noun ; a verb preposition composition (VPC) is the combination of two or three words from different grammatical categories to form a single semantic unit on a lexical or syntactic level, e.g. *turn down, run into*, et al. In our thesis, we will emphasize noun phrases(NPs) working as the terms to be represented in the feature space.

2.2.2 Feature Space Construction

The second step (feature selection) is intended to convert words or phrases from vocabulary to a corresponding vector of real numbers, which is used to capture the useful

	success	is	not	final	failure	fatal
success	0	1	1	0	0	0
is	1	0	2	2	1	1
not	1	2	0	1	2	1
final	0	2	1	0	1	0
failure	0	1	2	1	0	0
fatal	0	1	1	0	0	0

FIGURE 2.1 – An example of co-occurrence term representation.
Notes :It is generated from sentence "Success is not final, failure is not fatal". The co-occurrence window size is two (2 positions ahead and 2 positions behind).

syntactic and semantic properties of words [76]. Two main approaches for computing term representations can be identified in prior work [77] : co-occurrence term representation and word embedding representation. We start by introducing which kind of features to be represented for terms and then discuss how to present the feature matrix in the condensed manners for the convenience of clustering algorithms.

Co-occurrence term representation

Many previous works are proposed to apply the co-occurrence term representation techniques of interesting terms, which could be divided into three main parts : 1) the linguistic representation [78], [79], where the terms or phrases co-occur with the others in certain syntactic or grammar rules ; 2) the statistical representation [80], where a term co-occurs with other terms in a given window size ; 3) the hybrid representation, which combines the previous two representations. For example, in Figure 2.1, we present the co-occurrence matrix of a sentence with the statistical representation.

After the recognition of the co-occurrence situation between terms, the raw count of terms and their co-occurred terms will fill into the co-occurrence matrix. In Table 2.1, we depict the co-occurrence representation techniques into these three sections. The column 'terms' specifies the targeted terms that would be represented by feature vectors. The column 'co-occurred terms' means the terms that satisfy the co-occurrence's rules with their targeted terms. The column 'co-occurrence's rules' describes the co-occurrence situation to locate the co-occurred terms.

To benefit from the linguistic features of terms, some works thought that the noun pairs are more interesting, thus they tried to employ different restrictions to extract these pairs. Based on a predefined list of domain-specific concepts, *Clariana et Koul*

TABLE 2.1 – The co-occurrence techniques of term representation.

	terms	co-occurred terms	co-occurrence's rules
linguistic	nouns	nouns	in the same sentence [81] appearing respectively as subject and object in the same sentence [82]
		verbs	dependency relations [74]
statistical	any terms	any terms	within a fixed window size [83]
hybrid	NPs	NPs	within a fixed window size [84]

TABLE 2.2 – The condensed term representation.

	original value	resulting value	examples
co-occurrence representation	raw occurrence	a certain weight of occurrence (original dimension)	tf-idf [85], PMI [54]
	raw occurrence	the condensed probabilities (condensed dimension)	NMF[86], LSI [52], LSA [51], LDA [87]
word embedding representation	terms within a certain window size of a specific term	dense, low dimensional and real values (condensed dimension)	NNLM [88], word2vec [89], [90]

2004 [81] extracted noun pairs only if both of them appear in the same sentence. With the same idea but more confined, *Punuru et Chen 2012* [82] extracted noun pairs only if when they appear respectively as subject and object in a sentence. Other works explored the diverse utilities of the co-occurred verbs and nouns, for instance, *Leake 2006* [74] used nouns/NPs to extract concepts and used the co-occurred verbs/verbal phrases to extract relationships. From the statistical concerns, *Matsuo et Ishizuka 2004* [83] recognized the co-occurrence of words within a certain window size. In the hybrid aspect, within a certain window size, for example, *Barker et Cornacchia 2000* [84] made use of the occurrence of their extracted NPs.

In the co-occurrence term representation, for a targeted term, the counts of all its co-occurred terms compose a vector of this targeted term. Hence the feature size is equal to the size of the vocabulary. This kind of term representation suffers from data sparsity, which turns to be difficult for term clustering. To transform the co-occurrence representation into the term representation favored by clustering, we apply the dimensionality reduction techniques to condense the sparse feature space.

In the upper side of Table 2.2, from the raw occurrence matrix, tf-idf [85] could produce the weighted co-occurrence representation ; *Church et Hanks 1990* [54] proposed to apply PMI (pointwise mutual information) weighting to reduce bias in rare contexts, in which values below 0 are replaced by 0. In this situation, the dimension is

not reduced but helps to reduce the bias of word counts. While other works suggested transforming the value of straightforward occurrence into the latent features. For example, NMF (non-Negative Matrix Factorization) [86] was dedicated to solving the dimensionality reduction problem by performing feature compression ; besides, many topic models transform the document term occurrences into latent topic features, which lead to the condensed term representation in probabilities, e.g. LSI [52], LSA [51] and LDA [87].

Word embedding representation

Except for the co-occurrence representation, word embedding representation could directly generate the condensed representation from the text. The word embedding representation firstly assigns each word with a real vector, and learns the elements of those vectors, where the goal is to predict the next word in a given sequence [91]. For instance, the neural network language model (NNLM) [88] uses a multi-layer feed-forward neural network to predict the next word in a sequence, and uses back-propagation to update the word vectors such that the prediction error is minimized. Although this model aims to predict the next word, the term representation is also learned to capture the semantics at the same time.

However, training multi-layer neural networks using large text corpora is time-consuming. To overcome these limitations, many methods that specifically focus on word co-occurrences in large corpora have been proposed, where all the words in a contextual window will participate in the prediction task. The skip-gram model [90] predicts the words c that appear in the local context of a word w, whereas the continuous bag-of-words model (CBOW) [89] predicts a word w conditioned on all the words c that appear in w's local context. Overall, the word embedding representation has shown to outperform co-occurrence representation [77]. In the lower side of Table 2.2, these two models are named word2vec [89], [90], which could create the low dimensional and real-valued feature vectors for term representation.

Word2vec is a two-layer neural net that processes text by "vectorizing" words, which is useful in capturing semantic meanings of words. However, it fails to capture higher-level information that might be even more useful. Because it generates the same embedding for the same word in different contexts, it is given as *static word embedding*. Comparatively, *contextualized words embedding* aims at capturing word semantics in different contexts to address the issue about the context-dependent nature of words,

which could be achieved by some popular language models, e.g. Contextualized Word-Embeddings(CoVe) [92], Embeddings from Language Models(ELMO) [93], Transformer [94], and Bidirectional Encoder Representations from Transformers(BERT) [95].

Let's focus on Transformer language model [94]. The ideas of producing better contextualized words embedding has proved very successful in many NLP tasks, by applying the recurrent neural networks (RNNs) and convolutional neural networks (CNNs). However, due to their recurrent and sequential nature as neural language models, they tend to be slow to train and very hard to parallelize. While Transformer could get rid of these drawbacks. A Transformer is essentially composed of a stack of encoder and decoder layers. The role of an encoder layer is to encode the sentence into a numerical form using the *attention mechanism*, while the decoder aims to use the encoded information from the encoder layers to give the translation for this sentence. This model was firstly designed for natural language translation, meanwhile, it could also be applied for terms representation. In this thesis, we will not solve the issues of polysemy, which could be studied with multiple layers in language models, so we will only apply the static word embedding techniques, i.e. word2vec, in the following experiments.

2.3 The Clustering Algorithms

From the last section, the linguistic and statistical information from the text has already been translated into the word representations. Then term clustering will achieve by employing the word representations to calculate the similarity between terms, to group similar terms. Many previous works managed to apply the well-known clustering algorithms over texts in different manners. This section will introduce the clustering algorithms in the orientation of textual exploration, e.g. K-Means [96], K-Medoids [97], Affinity Propagation [98], DBscan [99] and Co-clustering [100]. The reason to choose these clustering algorithms is that they are representatives because of their diverse clustering methods. In Table2.3, we have presented the specialties of the five different clustering algorithms, their benefits and the drawbacks.

2.3.1 K-Means

The most typical clustering technique is k-means, which starts with randomly selected centroids and performs iterative calculations to change the centroids' positions

for better clustering performance [96]. 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 centroids ; 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. Except for the original usage in an unsupervised manner, there are also some semi-supervised variants of k-means clustering algorithms, i.e. seeded k-means and constrained k-means [101]. These algorithms explored the use of labeled data to generate initial centroids or the constraints to guide the clustering process. For instance, based on the term document probabilities, *Buatoom, Kongprawechnon et Theeramunkong 2020* [102] applied the different term weighting methods and selected the salient feature constraints to guide the seeded k-means clustering process to find the similar documents [102], [103].

2.3.2 K-Medoids

Similar to the k-means clustering algorithm, k-medoids also attempt to minimize the distance between centroids. In contrast to k-means, k-medoids choose the starting centroids as priori before calculation [97]. K-medoids provide many favorable properties : 1) it presents no limitations on input types, which means it is capable of numerical, categorical, and binary input matrix, while most of the other clustering algorithms only dedicate to the numerical matrix. 2) the choice of centroids is dictated by the location of a predominant fraction of points inside a cluster and therefore, it is less sensitive to outliers' presence. Briefly, it is more robust to noise and outliers as compared to k-means. However, this algorithm suffers from the negative effects of global issues because it does not reassign centroids to other clusters but only the initial cluster of the centroids [104]. Nevertheless, it could be a preferable clustering algorithm once we acknowledge the proper starting seed for each cluster. Similarly, many works applied the k-medoids algorithm for document clustering on the term occurrence matrix [105]-[107]. Also, it could be used for term clustering over a dimension reduced matrix [108].

2.3.3 Affinity Propagation

Like k-medoids, the affinity propagation (AP) clustering algorithm finds centroids to represent their located clusters during iterations. Unlike the dissimilar distance in k-

medoids, affinity propagation uses graph distance that performs in a 'message passing' way between data points [98]. It firstly measures the similarity between pairs of items. The 'message passing' means that this algorithm exchanges the real-valued messages between items and expects each item to choose its well-suited centroid with the lowest cost until a high-quality set of centroids and corresponding clusters gradually emerges.

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, and the amount of clustered elements. Nevertheless, affinity propagation is still interesting as a clustering algorithm for normal-size datasets.

The particularity of the AP algorithm provides the opportunities to group terms in a different way. *Li, Li, Xu et al. 2012* [109] [109] used affinity vector rather than a context vector for sentiment classification of terms. In a rather normal way, *Qasim, Jeong, Heu et al. 2013* [110] applied AP to cluster terms according to their relationships from text documents. They started by extracting the candidate nouns/noun phrases and applying the self-defined algorithms to extract their relationships of verbs/ verb phrases. According to these relationships, the similarity between terms is measured and used to cluster the related terms by the AP clustering algorithm. Then they assigned the relationships between clustered terms, in order to organize the knowledge in a graphical node-arc representation [110].

2.3.4 DBscan

Despite those distance-based clustering methods, DBscan (Density-based spatial clustering of applications with noise) [99] is distinguished as a density-based clustering method. It groups together closely packed points and mark the low-density points as outlier points to accentuate the high-density points into clusters and eliminate the negative impacts of outliers. DBscan clustering algorithm has some special benefits : 1) It can 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. On the opposite, DBscan allows for points to be part of more than one cluster, which might induce overlapping between clusters. It requires the knowledge of a domain expert during selecting key parameters, such

as the minimum number of points required to form a dense region (i.e., minPts) and the radius of a neighborhood concerning some points (i.e., eps). It is desirable to apply DBscan clustering algorithm even with several pre-experiments for the selection of parameters.

Interestingly, *Cui, Liu, Li et al. 2019* [111] used the DBscan algorithm to cluster the documents according to their relevance to a specific domain, where each cluster of documents is expected to support a category of this domain. Then they finished by extracting the most important sentence within documents as a subject of this cluster. In this manner, DBscan helps for document clustering and categorization.

2.3.5 Co-clustering

In a co-clustering algorithm (also called bi-clustering, block clustering), the individuals and the features of the individuals can be clustered simultaneously, which preserves the existing relation between individuals and their features. We are interested in the bi-clustering over the contingency table [100]. Typically, the input matrix would be arranged as a two-way contingency table. This algorithm shows the encouraging performance of 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 iteration [112]. Despite these facts, the co-clustering algorithm is attractive because it considers the relation between clustered elements and their features. For example, *Łancucki, Foszner et Polanski 2017* [113] aimed to extract the key terms and use the bi-clustering algorithm to separate the list of key terms into sub-categories of a domain. The term occurrence matrix for bi-clustering is made up of the counts of the pair of terms once they appear in the same document, in which the matrix has the same dimension as the number of key terms in both directions. Finally, they achieved term clustering purpose for a specific domain.

In brief, many clustering algorithms serve various purposes, e.g. term clustering, document clustering, sentiment classification, and knowledge representation. As shown in Table 2.3, some of the algorithms generate overlapping clusters, which bring many difficulties for a term clustering target. In order to solve this puzzle towards ontology

TABLE 2.3 – The summation of the clustering algorithms on text.

Clustering algorithms	Attribute	# clusters	Overlapping clusters	Utilities of text exploration	Benefits	Drawbacks
K-Means [96]	distance-based clustering	required	No	document clustering [102], [103]	1) it requires an extremely small number of iterations in order to converge	1) it is quite sensitive to the initial set of seeds ; 2) its performance could be strongly impacted by the noisy elements.
K-Medoids [97]	distance-based clustering	required	No	document clustering and term clustering [105]-[108]	1) it presents no limitations on input types ; 2) the choice of centroids is dictated by the location of a predominant fraction of points inside a cluster, therefore, it is less sensitive to outliers' presence.	1) it requires a large number of iterations in order to achieve convergence and are therefore quite slow ; 2) it does not work very well for sparse data
AP [98]	message passing clustering	NOT required	No	sentiment classification, term clustering for knowledge organization [109], [110]	1) the centroid of each cluster is specified after calculation, which turns out to be helpful for cluster interpretation	1) it is not friendly with big datasets because the time complexity of calculation increases dramatically
DBscan [99]	density-based clustering	NOT required but require neighborhood size	Yes	document clustering and categorization [111]	1) It can find arbitrarily shaped clusters because of the reduced single-link effect	1) it allows for points to be part of more than one cluster, which might induce overlapping between clusters.
Co-Clustering [100]	high dimensional data clustering	required	Yes	term clustering [112], [113]	1) it can extremely decrease the dimension of clustering, also appropriate to measure the distance between the tests ; 2) it is mining more useful information and can get the corresponding information in both clusters.	1) the problem of local optimization to each co-cluster individually ; 2) the lack of a well-defined global objective during each iteration.

learning, it requires an intermediate process to remove the obscure items and clearly separate the terms into clusters. Besides, several algorithms need the setting for the number of clusters as an input value. To find the optimal number of clusters, we need the evaluation indices to guide us for the identification task.

2.4 Evaluation Indices

A large number of indices provide possibilities to assess the quality of clusters, which are mentioned in this survey[64]. To simplify the discrimination process, we select two aspects of indices, respectively for internal evaluation and external evaluation. The internal evaluation indices make use of the intrinsic features of terms in vector space. For instance, *Liu, Li, Xiong et al. 2010* [114] summarized many internal evaluation indices based on the compactness and separation criteria of clusters, including silhouette width [115] and Dunn Index [116].

However, in the previous scenario, the clusters are difficult to be interpreted into some human-understandable concepts. In contrast, the external evaluation indices use the Gold Standard constituted of manually labelled terms, where the term clusters are measured to explain their concepts according to human's knowledge. *Amigó, Gonzalo, Artiles et al. 2009* [117] made a survey work to summarize the external indices into different aspects, which including evaluation by set matching (i.e. precision, recall and F-1 of macro/micro metric [118] and matthews correlation coefficient [119], [120]), indices based on counting pairs(i.e. asymmetric rand index [121] and pairwise metric [118]) and indices based on entropy (i.e. adjusted mutual information score [122]), etc.

Indices for Internal Evaluation

One intuitive approach to evaluate term clusters is to measure the compactness and separateness from the feature similarity of terms' observations. Even without any extra knowledge assistance, the cluster could be evaluated by some distance-based indices, namely internal evaluation. For instance, after applying the clustering algorithm over the terms' feature space, the assignment of terms could be directly evaluated by the internal indices.

Silhouette Width, proposed by *Rousseeuw 1987* [115] is considered as a pro-

minent metric for compactness and separateness. Silhouette method specifies how well each observation lies within its cluster :

$$s(i) \quad = \quad \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{2.1}$$

As presented in Equation 2.1, 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. On 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 âneighborâ cluster. A high average silhouette width indicates a good clustering according to features.

Dunn Index, proposed by *Dunn 1974* [116], was dedicated to the identification of "compact and well-separated clusters." In this manner, a ratio between compactness and separateness is used. Higher values are preferred, which leads to the best possible solution. The Dunn Index of each clustering situation is given by :

$$DI = \frac{\min_{1 \leq h < h' \leq k}[separateness(\mathbf{C_h}, \mathbf{C_{h'}})]}{\max_{1 \leq h \leq k}[compactness(\mathbf{C_h})]} \tag{2.2}$$

The *separateness* between two generic clusters $\mathbf{C_h}$ and $\mathbf{C_{h'}}$ is measured by the minimum Euclidean distance between the pairs across these two clusters. The *compactness* of a generic cluster $\mathbf{C_h}$ is measured by the distance between the furthest observations belonging to this cluster, in other words, to calculate the diameter of this cluster itself. It seems that both the worst separateness and the worst compactness are considered for evaluating the quality of clusters [123]. Notably, the Dunn Index does not exhibit any trend concerning the number of clusters. This property is exceedingly welcomed since the number of clusters varies in different iterations.

Indices for External Evaluation

For external evaluation, the indices are slightly different from the former because of the necessity of a gold standard. The observations of clusters here could be marked with different classes assigned in the gold standard. The external indices are applied to measure the coherence between the clustering labels and the assigned classes.

The Asymmetric Rand Index, proposed by *Hubert et Arabie 1985* [121], provides the comparison between the result of clusters and the correct classification of the items in clusters. This index is developed from the idea of the typical Rand Index (RI). Instead of counting a single observation, the typical Rand Index (RI) [124] counts the pairs of observations that are classified correctly, which is calculated by :

$$RI = \frac{a+b}{\binom{n}{2}} \tag{2.3}$$

, where $\binom{n}{2}$ is the number of disordered pairs in a set of n observations ; a refers to the number of pairs of observations that are in the same class and in the same cluster and b refers to the number of pairs of observations that are in different classes and in different clusters. Hence the score of RI depends on both the assigned number of clusters and of observations [125]. In fact, we cannot get the lowest value (e.g. zero) by the typical Rand Index, which indicates the worst situation like two random clusters. *Hubert et Arabie 1985* [121] made a modification to satisfy the null hypothesis, which means the value of Adjusted Rand Index (ARI) is expected to be 0 for two independent or random clusters and 1 for two identical clusters. The Adjusted Rand Index (ARI) [121] is defined as follows :

$$ARI = \frac{\sum_{i=1}^{k}\sum_{j=1}^{l}\binom{m_{i,j}}{2} - t_3}{\frac{1}{2}(t_1+t_2) - t_3} \tag{2.4}$$

, 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 class j. The m_{ij} indicates the number of observations in cluster i matching to class j. The $|C_i|$ and $|C'_j|$ represent the total number of observations for each cluster i or for each class j, respectively.

Additionally, ARI allows evaluating the quality of clustering, even if the number of clusters is different from the number of classes in the gold standard classification [126]. During experiments, the number of clusters varies a lot. Therefore the application of ARI allows us to perform a more accurate analysis.

Macro, Micro, and Pairwise Metrics were applied by *Galárraga, Heitz, Murphy et al. 2014* [118] in their work, to evaluate the clusters to the gold standard without considering the assigned cluster ID. This method directly links the clusters $c \in C$ to the

classes of Gold Standard $s \in S$ and measures the performance in three different ways, which are called macro analysis, micro analysis, and pairwise analysis.

- In the macro analysis, we define the macro precision as the fraction of pure clusters, where all the terms in a cluster are linked to the same class of the Gold Standard, as depicted in Equation 2.5 :

$$precision_{macro}(C,S) = \frac{|c \in C : \exists_{=1} s \in S : s \supseteq c|}{|C|} \tag{2.5}$$

  The macro recall is calculated by swapping the roles of the Gold Standard and the resulting clusters :

$$recall_{macro}(C,S) = precision_{macro}(S,C) \tag{2.6}$$

- In the micro analysis, we assume that the most frequent belonging of terms in a cluster is in the correct class. The purity of the resulting clusters is evaluated among all terms in Equation 2.7 :

$$precision_{micro}(C,S) = \frac{1}{n}\sum_{c \in C} max_{s \in S}|c \cap s| \tag{2.7}$$

  The micro recall is calculated by swapping the roles of the Gold Standard and the resulting clusters :

$$recall_{micro}(C,S) = precision_{micro}(S,C) \tag{2.8}$$

- In the pairwise analysis, we measure all the pairwise individuals' precision inside a cluster, whether both observations of a pair belong to the same class of Gold Standard. If it satisfies the condition, we name this pair as a *hit* in Equation 2.9 :

$$precision_{pairwise}(C,S) = \frac{\sum_{c \in C} \#hits_c}{\sum_{c \in C} \#pairs_c} \tag{2.9}$$

  The pairwise recall is calculated by dividing the number of all the correct pairs in Gold Standard :

$$recall_{pairwise}(C,S) = \frac{\sum_{c \in C} \#hits_c}{\sum_{c \in S} \#pairs_s} \tag{2.10}$$

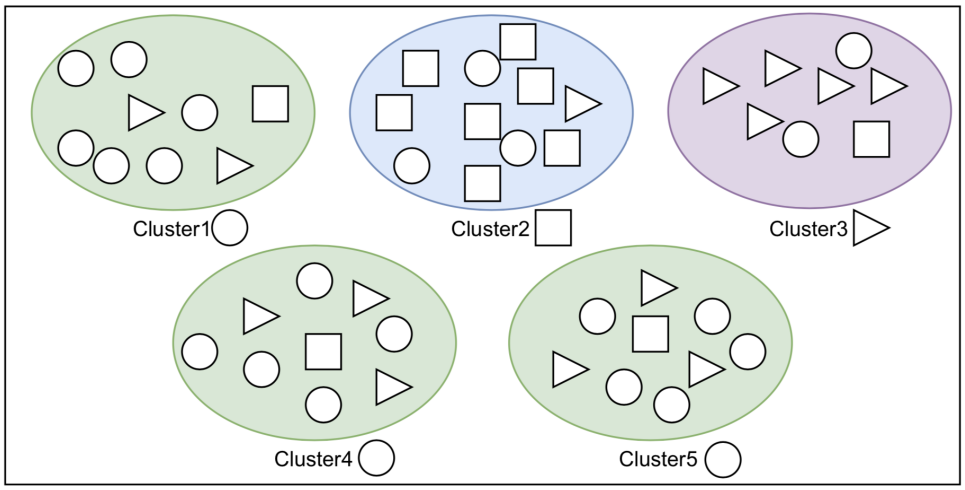


FIGURE 2.2 – An example of term clusters.

- In all cases, the F1 measure is defined as the harmonic mean of precision and recall :

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2.11}$$

To help understand these metrics, we use the example of Figure 2.2 to calculate the precision in these three metrics. Initially, from the result of term clustering, we get the grouped terms. Then terms would be labeled by the Gold Standard. In the figure, the items which are represented by the same symbol belong to the same classes. Each cluster will be assigned to a class (a symbol in the figure) that has a dominant occurrence in it. For the visualization purpose, the clusters tagged with the same class are partitioned into the same color.

For the macro precision, it is obvious that there is no pure cluster whose all terms tagged with the same label :

$$precision_{macro} = 0 \tag{2.12}$$

For the micro precision, we evaluate the purity of the labeled clusters. In Figure 2.2, we observe that each cluster has already voted for its dominant class, i.e. circle for 'Cluster1', 'Cluster4' and 'Cluster5', square for 'Cluster2' and triangle for 'Cluster3'. The same ratio is applied to all the other five clusters, until all the correct tagged items are added up in the numerator and the total amount of items is counted in the denominator.

As calculated in Equation 2.13 :

$$precision_{micro} = \frac{6+6+5+5+5}{9+11+8+9+9} = \frac{27}{46} = 58.70\% \tag{2.13}$$

For the pairwise precision, we calculate the number of hit pairs inside a cluster divided by the total number of pairs in a cluster by Equation 2.14 :

$$precision_{pairwise} = \frac{C_6^2 + C_6^2 + C_5^2 + C_5^2 + C_5^2}{C_9^2 + C_1 1^2 + C_8^2 + C_9^2 + C_9^2} = \frac{60}{191} = 31.41\% \tag{2.14}$$

The micro pairwise metric and the ARI metric have the same target, that is to measure the portion of the favorable pairs, in which both items are from the same cluster and also belong to the same class. However, the most obvious difference is that micro pairwise metric focuses on the hit pairs inside a cluster, while the ARI metric concerns all the possible pairs regardless of the clusters' boundary.

Besides the precision, we could also calculate the recall and F1 score accordingly, so as to evaluate the clusters with the assistance of the Gold Standard.

The **Matthews correlation coefficient (MCC)** is used as a measure of the quality of binary and multi-label classifications. When it confronts the binary classification [119], this metric will consider the values of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) in a confusion matrix, as depicted in Equation 2.15. If any of the four sums in the denominator is zero, the denominator can be arbitrarily set to one. The correlation coefficient value ranges between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 an average random prediction, and -1 an inverse prediction.

$$MCC(binary) = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{2.15}$$

Also, the MCC has been generalized to the multi-label case [120]. The generalization will take into account the $K \times K$ confusion matrix C, where K represents the total number of different classes, $t_k = \sum_i^K C_{ik}$ the number of times class k truly occurred, $p_k = \sum_i^K C_{ki}$ the number of times class k was predicted, $c = \sum_k^K C_{kk}$ the total number of samples correctly predicted, and $s = \sum_i^K \sum_j^K C_{ij}$ the total number of samples. The generalized MCC [127] is depicted in Equation 2.16 . The minimum value of multi-label MCC will be between -1 and 0 depending on the true distribution, and its maximum va-

lue is always +1. In addition, this metric is generally regarded as a balanced measure that can be used even if the partitions are of very different sizes.

$$MCC(multi-label) = \frac{c \times s - \sum_k^K p_k \times t_k}{\sqrt{(s^2 - \sum_k^K p_k^2) \times (s^2 - \sum_k^K t_k^2)}} \tag{2.16}$$

The **adjusted mutual information score (AMI)** [122] is an adjustment of the Mutual Information (MI) [128] score to account for chance. Like the development from the well-known Rand Index (RI) to the Asymmetric Rand Index (ARI), it solved the same problem in which the baseline value of information-theoretic measures does not take on a constant value.

It turns to be necessary to calculate the Mutual Information (MI) and entropy. Let (X, Y) be a pair of random variables with values over the space $\mathcal{X} \times \mathcal{Y}$. The marginal entropy of X is calculated based on its marginal distribution $p(x)$:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) = E[\log_2 \frac{1}{p(x)}] \tag{2.17}$$

The conditional entropy $H(X \mid Y)$ is calculated based on the marginal distribution $p(y)$ and the conditional entropy conditioned by the occurrence of sample y :

$$H(X \mid Y) = \sum_{y \in Y} p(y) H(X \mid Y = y) \tag{2.18}$$

The MI could be obtained by the difference between the entropy and its conditional entropy, as depicted in Equation 2.19 :

$$MI(X;Y) = H(X) - H(X \mid Y) \tag{2.19}$$

Then the formula of adjusted mutual information score (AMI) could be depicted in Equation 2.20 :

$$AMI(X;Y) = \frac{MI(X;Y) - E[MI(X;Y)]}{avg(H(X), H(Y)) - E[MI(X;Y)]} \tag{2.20}$$

2.5 Clustering based Approaches for Ontology Learning

Clustering approaches are widely applied to build ontology automatically.

Among all the clustering algorithms, **k-means clustering algorithms** are widely applied for ontology learning. *Ganz, Barnaghi et Carrez 2014* [129] tried to use topical ontology to interpret the real-world data, i.e. raw sensor data. They used an extended k-means clustering method and apply a statistic model to extract and link relevant concepts from the raw sensor data and represent them in the form of a topical ontology. Finally, they succeed to automatically create and evolve topical ontologies based on rules that are automatically extracted from external sources. Based on the same idea of topical ontology, *Fortuna, Mladenič et Grobelnik 2005* [130] applied k-means clustering algorithm over the term-document distribution matrix to give suggestions for the terms to help user taking the decisions as the means for semi-ontology building.

In the using of the **other typical clustering algorithms**, Louge et al. [131] implemented affinity propagation clustering algorithms [132] upon string similarity measurement for the construction and population of ontology. *Togatorop, Siagian, Nainggolan et al. 2020* [133] applied the POS tagging techniques to identify the dedicated terms and built up the word2vec word representation for them. Then they applied the DBscan clustering algorithm over the word representation, which resulted in several term clusters with lists of terms inside to prepare for ontology construction. *Giannakidou, Koutsonikola, Vakali et al. 2008* [134] utilized the co-clustering method to yield a series of clusters, each of which contains a set of resources together with a set of tags. The extracted concepts or relations will enrich the content of ontology.

In particular, **hierarchical clustering algorithms** show their specialities on taxonomic relation discovery of ontology [135]. For example, Huangfu et al. [136] chose the hierarchical clustering algorithms for knowledge organization. Their work examined the different word representations techniques at the same time, i.e. word2vec, doc2vec, and gloVe embeddings, and organized those clusters regarding their containing keywords to build up a hierarchical knowledge organization system. *Ozdikis, Senkul et Oguztuzun 2012* [137] applied the agglomerative text clustering to cluster hashtags of tweet contents. They analyzed the contexts of hashtags and their co-occurrence statistics with other words and identified their paradigmatic relationships and similarities. In this way, they are able to capture statements that actually refer to the same concepts.

For **other clustering scenarios**, different clustering strategies are also used for ontology learning. *Koskela, Smeaton et Laaksonen 2007* [138] presented a entropy-based clustering method for modeling semantic concepts of multimedia repositories. It introduced the procedures of exploiting the structure of a multimedia ontology and discovering the existing inter-concept relations. To be unified, *Niekler et Kahmann 2016* [53] proposed a workflow to extract information from collections of text to create knowledge bases for medicine domain. It started by focusing on the co-occurrence statistics and then inferred a probabilistic graph-based data structure to discover more clusters. However, this workflow has the limitation that it works only on the required graph-based structure for term clustering, and it is difficult to be extended to other term clustering methods.

In addition, there is **an interesting survey paper** by *Sarwar, Ahmed, Habib et al. 2020* [139], who exploited a framework to build up an appropriate ontology according to user requirements in the unsupervised approaches. They started from the term document matrix and applied the term weighting schema to get the extended numerical matrix, e.g. binary, tf-idf (term frequency-inverse document frequency), entropy, and the other two variants of tf-idf. Based on the generated word representations, they employed the k-means, k-medoids, and fuzzy c-means clustering algorithms to cluster the selected terms of a domain ontology. Then they selected the best combination of word representation and clustering algorithms with the highest accuracy. Eventually, the ontology had been constructed by the groups of terms. However, the evaluation work only relied on the accuracy metric without considering other worthy metrics, in which the clustering results are not examined thoroughly in the different aspects.

2.6 Summary

In previous work, we notice that the facilities of clustering algorithms and word representation are inseparable. To achieve the term clusters, the individual clustering algorithms could combine with the independent word representation techniques ; also the new clustering algorithms could be derived directly from the features of word representation, without using any typical clustering algorithms.

It is evident that there is not a systematic framework to discuss the linkage between different phases of term clustering from text and evaluate the clustering results comprehensively. To fill this blank, on the one hand, we provide a complete frame-

work in Chapter 3 of this dissertation. This framework does not purely examine the performance of term clustering but also explores the word representation possibilities with statistical and syntactic concerns. On the other hand, we explored the new clustering algorithms derived from the specialties of word representations. The resulting term clusters will then be evaluated by the gold standard of ontology's core concepts. In brief, the whole procedure helps to dig out as many linguistic features for clustering purposes and guarantees that the terms in clusters conform to the concept formation of ontology learning.

CHAPITRE 3

THE PROPOSAL FRAMEWORK FOR TERM CLUSTERING TOWARDS ONTOLOGY LEARNING AND ITS DEPLOYMENT

In this chapter, we firstly propose a framework that describes the task or process of term clustering as a component of the ontology learning process, which helps the user to apprehend its inputs and outputs, tune its parameters, and adapt techniques for performing it. In Section 3.1 we will present the framework including its process, the inputs, the outputs, and the applied techniques, in which a cluster could be considered as a group of terms referring to the same concept of the ontology. We propose also a new term representation method in the linguistic perspectives, which remarks the relation between noun phrases and their co-occurred verbs. This method generates the NPs representations for the clustering task, where the co-occurred verbs contribute to the feature vectors of the noun phrases. Except for that, we also examined other term representation techniques from a statistical perspective, i.e. word2vec and LDA techniques. Subsequently, we explored the performance of the different combinations between the NPs representations and the various clustering algorithms.

Therefrom, we notice that the term representation techniques of LDA have prominent performance in term clustering tasks, thus in Section 3.2, we introduce a new strategy of adapting the topic model LDA for term clustering as a task for ontology building, in which the algorithm transforms the topic-term probabilities distribution toward a partitioning of the set of terms into disjoined parts (clusters). In Section 3.3, we present the details to conduct the different experiments for various concerns, i.e. practical parameters, the clustering randomness, and the size of clusters. In Section 3.4, we analyze the influence of cluster numbers and explore the optimal cluster numbers for different clustering approaches. Also, the comparisons between these two different clustering strategies are conducted in terms of the clustering performance. A summary of this

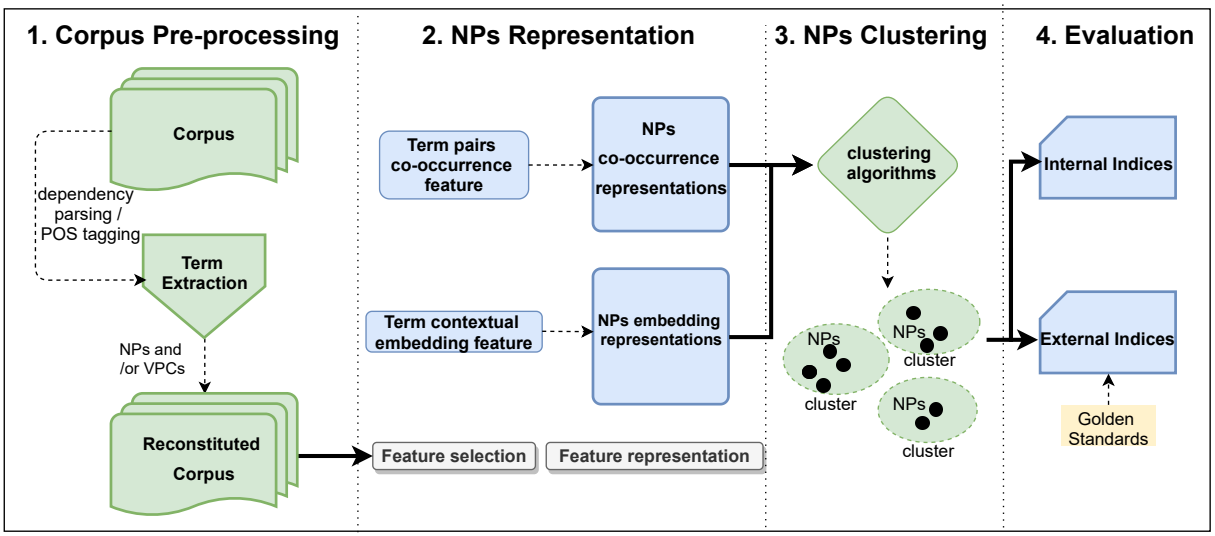


FIGURE 3.1 – The framework of term clustering for ontology building.

chapter is provided in Section 3.5.

3.1 The Proposal Framework of Term Clustering

To have a clear view, the framework of term clustering is visualized in a workflow, as shown in Figure 3.1. The workflow is comprised of 4 stages, which gradually transform the plain text into the dedicated term clusters and finish with cluster evaluations. The corpus pre-processing (stage 1) provides textual resources to extract the NPs and/or VPCs, by the identification of dependency parsing and POS tagging. This step finishes by storing the extracted terms into the reconstituted corpus. In the representation stage (stage 2), we apply the feature selection techniques over the reconstituted corpus and select two kinds of features : 1) term pairs co-occurrence features 2) term contextual embedding features. Then with the feature representation techniques, two kinds of NPs representations are generated preparing for the clustering tasks. Given the clustering algorithms (stage 3), the NPs representations could be executed into NPs's clusters. In the evaluation stage (stage 4), the clusters of NPs would be examined by two different indices for clustering analysis.

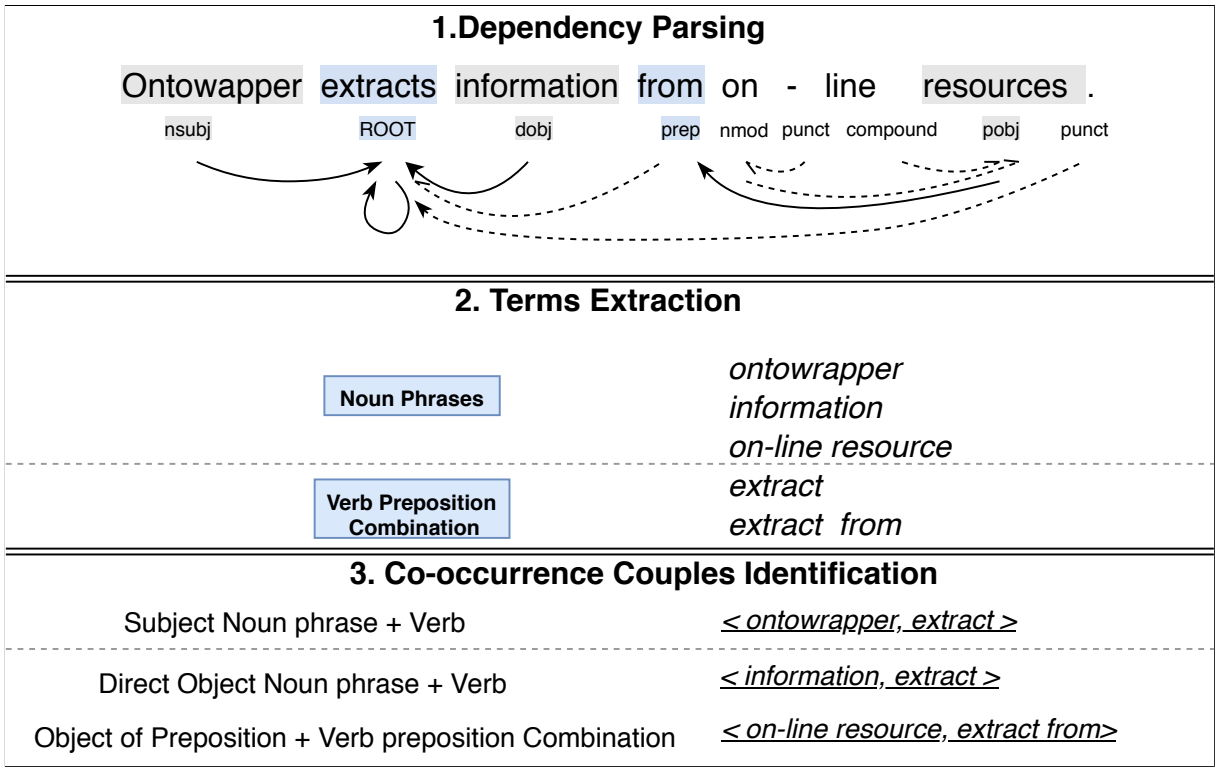


FIGURE 3.2 – The instantiated co-occurrence couples extraction. Extracted from Xu et al. [140]

3.1.1 Corpus Pre-processing

From the given corpus, we firstly analyze the relations between terms in their context. Regarding the utility of syntactic roles, a sentence's skeleton comprises 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 their contextual components, i.e., verbs or VPCs, could present the concrete connection between NPs. The syntactic information could be extracted to help identify NPs acting as a subject or object and their co-occurred verbs.

To explain how noun phrases (NPs) with subject and object role and verb-preposition combinations (VPCs) are extracted, we propose to use spaCy [1] [141] 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

1. https://spacy.io/

tools, such as cleanNLP [142] and coreNLP [143].

With the parser tool, we provide an instance to show how the co-occurrence couples are extracted in Figure 3.2. As shown in the top of Figure 3.2, 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. In contrast, 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.

Then in the middle part of Figure 3.2, with the assistance of head pointers, noun phrases (NPs) and verb preposition combinations (VPCs) could be lemmatized and extracted in the compound format. In the bottom part of Figure 3.2, the NPs-VPCs pairs are identified in the right hand and their tags are listed in the left hand. Furthermore, we need to pay attention to the distinction between passive and active sentences. To simplify the composition of sentences, it is practical to record the passive subject (nsubjpass) as a direct object (dobj).

Finally, the NPs and VPCs, that are extracted from one original document, are recorded as a list of terms in a reconstituted document.

3.1.2 The NPs Representations

We begin to discuss the proposed co-occurrence representation, which takes advantage of NPs-VPCs pairs' raw counts. Then we study the variants of the proposed co-occurrence representations : to apply the weighting strategy or to employ dimension reduction techniques. Finally, we present the embedding representation techniques with word2vec and with LDA, and generate the word2vec embedding representation and the topic embedding representation.

The Proposed Co-occurrence Representations

In Section 2.2.2 of the state of art, we discussed that the terms co-occurrence representations are calculated based on the raw counts of the co-occurred terms' pairs. We summarized that the co-occurrence techniques are divided into two parts according to their co-occurrence's rules : 1) when the term pairs appear within the same sentence or within the special syntactic positions ; 2) when the term pairs appear within a fixed

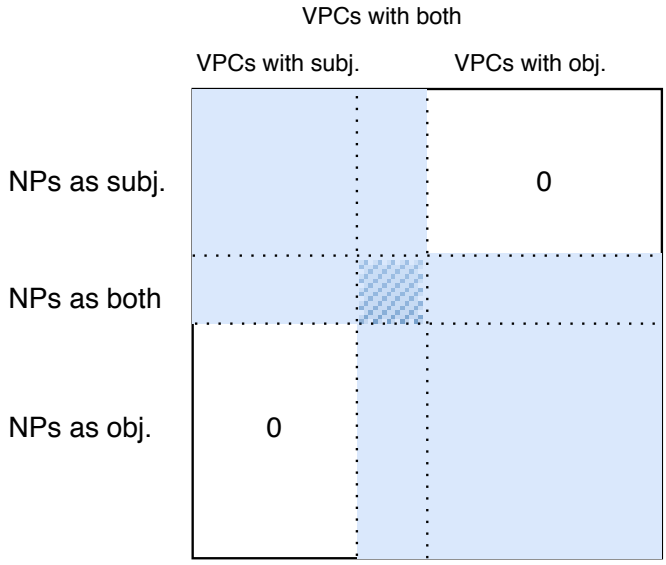


FIGURE 3.3 – The merged co-occurrenc matrix. Extracted from Xu et al. [140]

window size. In this section, we propose a new co-occurrence representation technique following the first co-occurrence rule.

Before introducing the new co-occurrence representation technique, we will discuss how the terms' pairs are represented. As we discussed in the last section, the NPs-VPCs pairs are the only term pairs that we take into account into representation. However, we notice a big difference between the subject role and the object role of NPs to their co-occurred verbs / VPCs. For instance, given a sentence *'human eat pizza'*, the parsing tools could recognize that *human* works as subject of *eat* and *pizza* works as object of *eat*. If we store these two instance pairs together, i.e. $< human, eat >$ and $< pizza, eat >$, their syntactic roles are lost. In this wrong information, we could falsely induce that *'pizza eat something'*.

To overcome this mistake, we need to separate the two kinds of instance pairs in the different repositories. Thus we divide the NPs representation into two parts : the subject NPs co-occurrence representation and the object NPs co-occurrence representation.

In fact, one kind of term' pairs either working as subject or object, could only convey the partial linguistic knowledge from a sentence. Thus it is profitable to deliberately combine subject and object term' pairs, in order to cover the entire linguistic information. To preserve the distinctions between subject and object attributes, we suggest the merged co-occurrence matrix (in Figure 3.3) in our published paper [140]. This model differentiated NPs and VPCs into 'pure subject'(upper-left corner), 'pure

object'(bottom-right corner), and the common part (center shaded rectangle). The common part means NPs and VPCs appear in both subject and object roles. On the whole, the merged matrix comprises nine 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 (the shaded rectangles, also the overlap between the subject rectangle and object rectangle), are filled with the accumulative frequency of the subject and object pairs. For example, in 3.2, the instance pairs $< ontowrapper, extract >$ and $< on - lineresource, extractfrom >$ will locate in the upper-left corner of matrix in Figure 3.3 as 'pure subject' ; the instance pair $< information, extract >$ will locate in the bottom-right corner of this merged matrix.

As long as subject and object co-occurrence pairs join together, the merged matrix theoretically encompasses complete linguistic information. The merged matrix will work as an integrated NPs co-occurrence representation for the further clustering tasks.

The Variants of the Proposed Co-occurrence Representations

The integrated NPs co-occurrence representation is composed of the raw counts of the terms' pairs. As we discussed in the state of art, the original NPs co-occurrence representations could be transformed in two directions : 1) applying the weighting strategy to strengthen the representation of features ; 2) condensing the size of feature with the dimension reduction techniques.

- **Weighted Co-occurrence Representation.** Based on the NPs co-occurrence representation, we would like to weigh the occurrence value to distinguish the different importance of the co-occurrence pairs. TF-IDF is designed for this discriminative purpose. Basically, this algorithm could extract the most descriptive terms from documents, which can be extended to highlight the most significant NPs for the specific VPCs. In this analog, the column of VPCs works as the function as the document. Owing to the application of TF-IDF over the originally proposed co-occurrence representation, on the one hand, the close connected NPs and VPCs are able to be emphasized. On the other hand, this technique supports weakening the weights of the most common and rare NPs and VPCs. However, the weighting strategy cannot help to reduce the dimension of the original representations.
- **NMF Co-occurrence Representation.** Term co-occurrences could be divided into 3 levels according to the identity of words in context [144]. In the first-

order co-occurrence, terms appear together in an identical context, i.e. in a NP-VPC pair, a NP co-occurs with a VPC within a sentence. Two terms, who share at least one-word context and have strong syntactic relations, are associated through second-order co-occurrence, i.e. in the originally proposed co-occurrence representation, two NPs are second-order co-occurred if they have similar count value in more than one VPC column. Besides, terms do not co-occur in context with the same words but between words related through indirect co-occurrences, namely third (higher) order co-occurrence. To capture the features of the third (higher) order co-occurrence, NMF [86] is applied to condense the isolated VPCs into some encoded features. In this way, the NPs associated with the indirect co-occurrence could be presented in the new dense feature space.

The Embedding Representations

- **Word2vec Embedding Representation**. The contextual information of terms allows us to build feature vectors that are adapted for semantic similarity tasks. In the state of art, we discussed that word2vec [89] is the typical embedding algorithm to generate the NPs embedding representations. In addition to the word2vec algorithm, we also resort to the topic model LDA to represent the NPs with the topic embedding.
- **Topic Embedding Representation**. The NPs topic embedding representation was extracted from the term topic probabilities trained by the LDA topic model. In this manner, all of the terms in a document will be considered as the context information for a certain term in that document.

The word2vec embedding representation considers the range of the contextual terms to be within a certain window size (i.e. equal to 5 or 10). Comparatively, the range of contextual terms for the topic embedding representation is further large (i.e. equal to the length of a document). Moreover, the features of topic embedding representation possess the intuitive meaning for humans, e.g. topics, whereas the features of word2vec embedding representation could only provide the indirect sense as co-location context. The relation from terms to documents could be transformed and condensed into the relation from terms to topics. Nevertheless, the same as word2vec embedding representation, the topic embedding representation could be presented with the required dimension of features.

All in all, after the NPs representations are prepared, this framework will be continued for the term clustering tasks. Except for the typical clustering algorithms, we also propose a LDA-based term clustering strategy in the next chapter.

3.2 The LDA-based Term Clustering Strategy

For the term clustering task, the application of the clustering algorithm ignores the quality requirement of modular ontology learning. For example, the resulting clusters ought to have the suitable and balanced size, neither too large (i.e. a cluster contains 90% NPs) nor too small (i.e. a cluster contains only one NP) ; a term cluster should avoid overlapping with other clusters ; a cluster is anticipated to possess the meaning of a certain concept. In order to obtain better clusters towards ontology learning, we propose a term clustering strategy based on LDA. This strategy is composed of term cluster formation schemes and term cluster thinning metrics.

To interpret LDA's performance, there are two directions to analyze the LDA model (the introduction of LDA please see Section 4.3.3), one for document aggregation, another for term aggregation. This chapter concentrates on aggregating terms by the feature of their aligned topics learned from the LDA model. One of LDA's direct outputs is the probabilities matrix between terms and topics $p(w \mid t)$, where each topic is represented by its probabilities of terms. Based on this probability, a term is able to be clustered into a topic group. In this section, I will discuss what are the possible schemes to form term clusters and to refine the term clusters.

3.2.1 Term Cluster Formation

From the probabilities matrix between terms and topics $p(w \mid t)$, each topic is composed of the all terms with different probabilities and each term possesses the probabilities to each topic. To form the term clusters, we assume that each term could only be clustered into one topic, not multiple topics. If one term has a rather high probability to a list of different topics, it would be assigned to only one topic, which has the highest topic-word probabilities. In this manner, eventually, we extract a cluster of terms (a subset of terms) from each topic by selecting the most relevant terms for that topic. And the clusters are disjointed.

I **Problem** : If a topic has low probabilities for every term, this topic is not capable to form a term cluster (We name this kind of topic as the opposite kind of the 'clustering-oriented topic' in the following text).

I **Question** : Is the highest $p(w \mid t)$ of each topic representative to distinguish the clustering-oriented topics ?

To answer this question, we visualized the highest $p(w \mid t)$ of each topic in Figure 3.4. This figure is generated from the topic-word probabilities based on the Computer Science corpus (which will be introduced in Section 3.3). The horizontal axis represents the separate topics (i.e. 200 topics), and the vertical axis represents the value of the highest $p(w \mid t)$. The black dots are the highest $p(w \mid t)$ of one topic, and the ID of this topic is written next to the black dots. The downwards grey arrow shows the range of the top-5 highest $p(w \mid t)$ of each topic.

In Figure 3.4, we can observe that it has 50 evident topics from a total of 200 topics. The results demonstrate that the highest $p(w \mid t)$ of each topic is capable to make a distinction between the clustering-oriented topics and the rest topics.

II **Problem** : The variance between clustering-oriented topics and other topics is not large.

II **Question** : Is the normalized highest $p(w \mid t)$ able to enlarge the gaps between the clustering-oriented topics and other topics ?

$$normalized \;\; p(w_i \mid t_k) = \frac{p(w_i \mid t_k) - min(p(w \mid t))}{max(p(w \mid t)) - min(p(w \mid t))} \tag{3.1}$$

, where $p(w \mid t) = \{p(w_i \mid t_k), \forall i \in V, \forall k \in T\}$, w_i denotes the i-th term in the vocabulary V and t_k demotes the k-th topic for the total T topics. According to the Equation 3.1, we visualized the highest normalized $p(w \mid t)$ of each topic in the Figure 3.5. The vertical axis represents the value of the highest normalized $p(w \mid t)$. The blue dashed horizontal line indicates the averaged value of the top-5 normalized $p(w \mid t)$ for all the clustering-oriented topics.

Comparing to Figure 3.4, we notice that the difference between the clustering-oriented topics and other topics becomes enlarged.

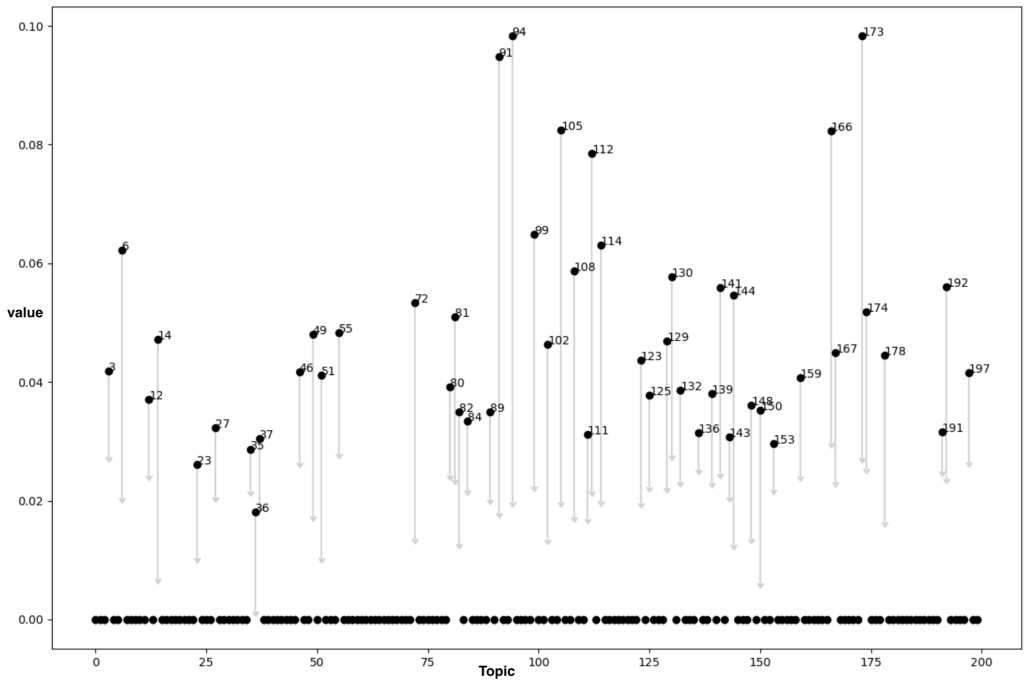


FIGURE 3.4 – The distinction of the maximum term probability in each topic cluster.
Notes : The Y axis represents the value of $p(w|t)$ of each topic in the X axis.

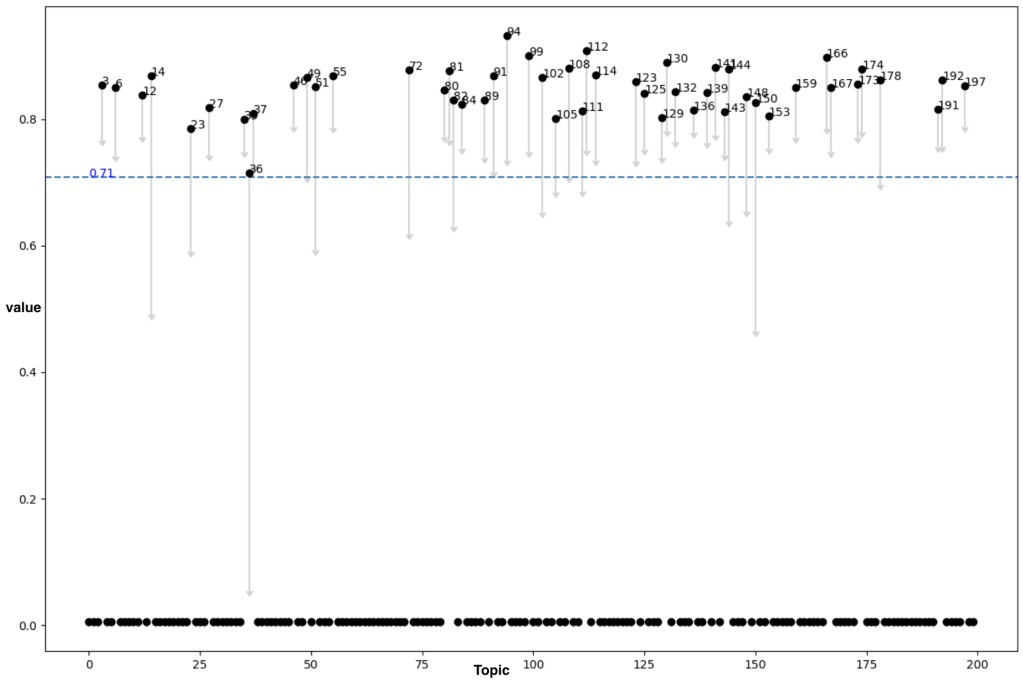


FIGURE 3.5 – The distinction of the normalized maximum term probability in each topic cluster.
Notes : The Y axis represents the value of $normalized\ p(w|t)$ of each topic in the X axis.

3.2.2 Term Cluster Thinning

The term cluster has been formed by aggregating together all terms according to their topic preferences. However, it is easy to notice that the term clusters' size is exceptionally biased since the biggest cluster is hundreds of times larger than the smallest cluster. Therefore, thinning clusters turn out to become necessary to guarantee the balanced term clusters.

In this section, we start by discussing the upper bound of the size. I draw the term cluster figures to show the difference before and after the cluster's thinning with the chosen size. Additionally, we also provide alternative options to help limit the size of term clusters.

III **Problem** : When the size of a term cluster is too large, this cluster will be covered up by the majority of the unimportant terms.

III **Question** : How can we control the upper bound of size (top-n) for all clusters ?

Following the same idea as Figure 3.5, we increase the upper bound value to top-10, top-20, top-30, top-40 and top-50 to draw the normalized $p(w \mid t)$ of each topic. Please check their corresponding figures in Appendix 7.1. Along with the increasing upper bound, we have a clear view that the downward grey arrow is extending to the bottom axis, which shows that the evident value's coverage is enlarged from the highest to the lowest value. When the upper bound reach the top-50, it is remarkable that the value range is almost totally covered ; in other words, the top-50 NPs of a topic can cover the significant value range of that topic. It indicates that we are convinced to choose top-50 as the upper bound for all the clusters.

It is worth mentioning that, in Figure 7.6 of Appendix 7.1, the numerical value in blue indicates the number of terms that are allocated to each topic when we set the upper bound is top-50. We observe that this value is less than 50, it is due to the fact that even though a term possesses the probability within the top-50 highest list of one topic, this term might not be allocated to this topic. In this situation, this term possesses the rather high probabilities of more than one topic. It reflects that the upper bound (top-50) of topic clusters could effectively decrease the size of clusters and highlight the important terms for a cluster.

IV **Problem** : From other perspective, if we want to relax the restriction of size, which threshold could be considered to refine the clusters ?

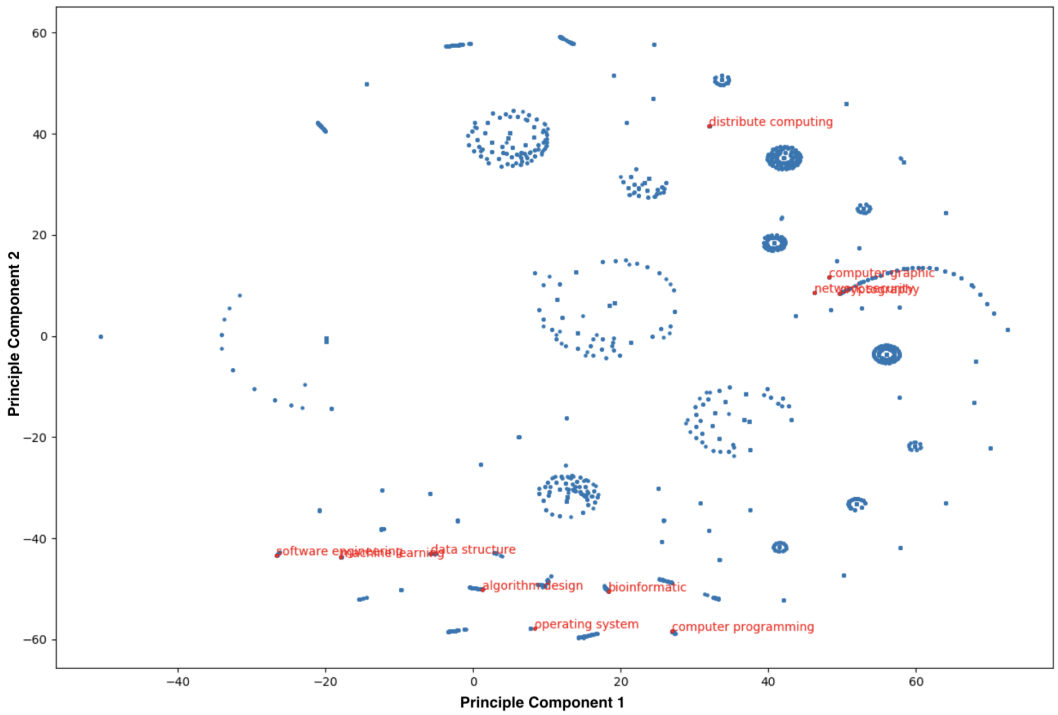


FIGURE 3.6 – The 2D visualization of the clustering for all NPs by T-SNE. *Notes : the core concepts of the Computer Science corpus are positioned as the red dots.*

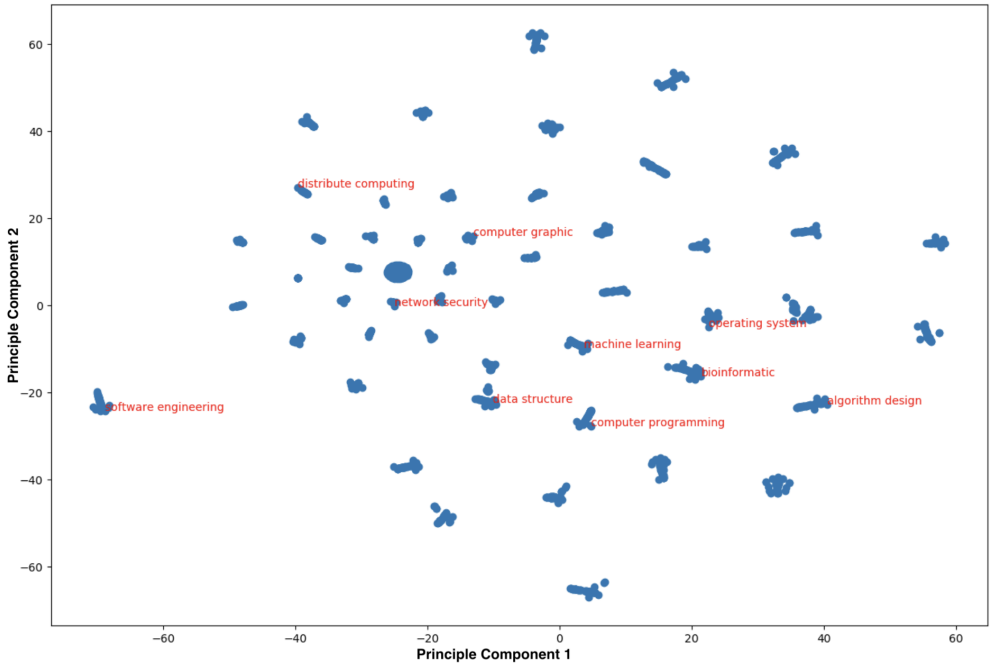


FIGURE 3.7 – The 2D visualization of the clustering for the residual NPs by T-SNE. *Notes : the core concepts of the Computer Science corpus are positioned as the red dots.*

IV **Question** : How to choose the threshold to obtain the significant NPs of the distinguished topics ?

We have drawn the blue dash line to signify the averaged normalized $p(w \mid t)$ of top-n in Appendix 7.1. With the increase of the upper bound value, the dashed line becomes lower, where the average value for top-50 NPs of those distinguished topics is 0.14.

From our observation in Question 3, it is conclusive to choose top-50 as optimal upper bound ; thus, it is probable to set the threshold of normalized $p(w \mid t)$ as 0.1 (used to exclude NPs which is lower than 0.1). Comparatively, this is a more loose and closer restriction than that of 0.14.

To simplify the calculation step, we choose to use this soft threshold on averaged normalized $p(w \mid t)$, which could be seen as an extension of the upper bound of size.

Also, we draw the figure of term clusters before and after thinning as a contrast, i.e. Figure 3.6 and Figure 3.7. In the figures, the feature matrix is transposed from the topic-word probability $p(w \mid t)$. Based on this feature matrix, then the top-two principal components are calculated and used as the axes. Finally, the feature space is represented in cosine distance. It is worth mentioning that, the core concepts of a Computer Science corpus (which will be detailed in Section 3.3) are positioned as the red dots. In Figure 3.6, the T-SNE presents the 2-dimension feature space for all NPs ; while in Figure 3.7, the T-SNE presents the 2-dimension feature space for the remaining NPs, which were pruned by the soft threshold on averaged normalized $p(w \mid t)$.

From the contrast, it is noticeable that the cluster thinning technique has a strong impact on the term clusters' size. The pruning operation also increases the compactness and separateness of clusters in Figure 3.7, which indicates that the remaining NPs possess significant features for their affiliated topics.

In brief, to formulate the discussed procedures above, we provided the related pseudo algorithm in Algorithm 1. The pseudo-code is divided into three main parts, corresponding to the solutions of the related questions that we have proposed in previous procedures.

Algorithm 1 The pseudo-code of LDA-based clustering strategy

Require: A matrix of size $(V \times K)$, $p(w_i \mid t_j)$; The size of topical cluster, n
Ensure: the set of terms in topical clusters, $\mathcal{Z}$;

```
// (Solution of Problem 1&2 : assign terms to topics that have the highest probabilities)
for i = 0 to V do
    j ← argmax_j p(w_i | t_j)
    w_i ∈ z_j ; z_j ∈ Z
end for
// (Solution of Problem 3 : remove non-significant topics)
for j = 1 to K do
    if max p(w_i | t_j)_{∀w_i ∈ z_j} < 1e^{-3} then
        remove z_j from Z
    end if
end for
// (Solution of Problem 4 : reserve top-n for each topical cluster)
for z_j to Z do
    sort all w_i ∈ z_j by p(w_i | t_j) ;
    reserve only first-n w_i in z_j ;
end for
```

3.3 Experiment

3.3.1 Corpus Preprocessing

We choose two corpora in different domains : the news stories and computer science abstracts with the aim of term clustering experiments :

- **The Reuter Corpus** [2], used by *Oramas, Anke, Sordo et al. 2016* [145], is a collection of documents that appeared on Reuters newswire in 1987. We randomly selected 1 000 documents from 10 788 plain text documents, acquired from the dataset of NLTK library [3]. During the text document selection, we guarantee that each document will possess the acceptable length—at least 1000 tokens. This corpus contains 90 topics [4], for the convenience of experiments, we have divided them into 4 main subdomains : *Corporate-Industrial*, *Economics and Economic Indicators*, *Government and Social* and *Securities and Commodities Trading and Markets*. Each subdomain will be represented respectively by a core concept : *company, economic, government, and commodity*.
- **The Computer Science corpus** comprises the abstracts of the academic ar-

2. Reuters-21578 : http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html
3. http://www.nltk.org/nltk_data/
4. http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/a16-rbb-topic/topics.rbb

TABLE 3.1 – The corpus size and statistics.

Corpus	#OriginalDocs	#SampledDocs	#tokens	#NPs	#unique NPs	#VPCs	#unique VPCs
Reuter	10 788	1 000	301 303	10 254	5 043	945	294
Computer Science	6 514	1 000	212 437	8 259	5 139	6 965	1211

ticles in the Computer Science domain, extracted from the Web of Science [146] [5]. In the original corpus, there are 6514 documents in plain text format, from where we randomly extract 1 000 documents with the acceptable length (at least 1000 tokens in each abstract). We separate this domain into 10 subdomains corresponding to 10 core concepts.

As we talked about, the proposed co-occurrence representation in section 3.1.2 focuses on the co-occurrence of NPs and VPCs in sentences, while the embedding representation techniques only interest the appearance of NPs in documents. It is necessary to re-construct the documents only with NPs and/or VPCs. To achieve these procedures, we recognize the NPs and VPCs and pre-process them with a unified format, to avoid the redundancy resulted from the various formats. Finally, each document will be simplified and transformed into a sequence of NPs, namely the re-constituted documents. In this manner, the input for NP representation techniques is guaranteed to be the same. We will apply this kind of documents to calculate the *Word2vec Embedding Representation* and the *Topic Embedding Representation*. Simultaneously, the re-constituted documents would also be used as the input of the LDA-based term clustering strategy. The statistics of the pre-processing steps are shown in Table 3.1, of which the statistical knowledge of NPs and VPCs assist to build up the *the merged co-occurrence matrix* in Figure 3.3. Based on this matrix, we could finally get the *the Proposed Co-occurrence Representation* and the other two *Variant Co-occurrence Representations*.

3.3.2 Gold Standard

The Gold Standard in this thesis is regarded as knowing the truth concerning a specific question or task. They are the ideal judgment of term clusters. For instance, the Gold Standard for our task (term clusters evaluation) is the agreed label for each

5. https://data.mendeley.com/datasets/9rw3vkcfy4/6

TABLE 3.2 – The experimental recordings of the annotators.

Rating part of terms	AnnotatorID	Total time
0-200 NPs	Annotator1	25 mins
201-400 NPs	Annotator2	57 mins
401-600 NPs	Annotator3	27 mins

term, where each term has a corresponding label with the domain knowledge. The domain experts would be involved to provide this kind of knowledge. However, it is inevitable that the domain expert tags the conflict or disagreed labels for the same term. To understand the confidence of the practical Gold Standard, it is necessary to measure the agreement of different annotators.

The precondition is that we have an entire Gold Standard for all the terms, which is labeled continuously by a domain expert. Meantime, we asked for the other annotators to assign labels to a different part of the terms. Then we would like to compare the part of practical Gold Standard with the corresponding part of one annotator. In this way, we could assess the different agreement degrees of different annotators, with the metrics for two human raters.

An Example of Annotation Task

As an example, we present a task description document to explain the tasks and the expected outputs, as shown in Appendix 7.12. To build up the Gold Standard, we have three volunteers with a domain-related educational background. During the evaluation step, the raters are allowed to use networks or books if they need to acquire further knowledge for this task ; meanwhile, the total time to finish the task is recorded in Table 3.2. We can see that it has a big difference in the consuming time for different annotators. And also, annotating work is time-consuming. We assume that the average consuming time of 200 NPs is 36 mins ; it will use 180 mins (3 hours) for 1000 terms and 900 mins (15 hours) for 5000 terms (the size of vocabulary in one corpus) for a single annotator. For this reason, we did not ask the annotating volunteers to label all of those terms.

The agreement degree between annotators

To compare two annotators' results, the confusion table could be a direct and effective method to summarize the accordance and discords. To have a distinctive view of the confusion table's values, we present this table in a heat map where the background of higher value has lighter color. As shown in Figure 3.8, eleven core concepts and two extra tags ("Others" and "Unknown") are used to label the terms in the Computer Science corpus. The vertical axis denotes the labels of Gold Standard, and the horizontal axis locates the labels of Annotator1. In this table, the values in diagonal lines denote the agreed labels between two annotators.

It is clear that except for "Others" and "Unknown," we notice the significant and high agreements on several core concepts, e.g., "Computer graphs," "Data structures," "Machine learning," and "network security." However, not all the core concepts have high accordance. In our view, one possible reason is that the eleven core concepts are not evenly distributed. In addition, this unbalanced phenomenon also could be due to the limited knowledge of domain experts. In fact, all of the annotators from this experiment have at least a master's degree in computer science, but they have the same specialty in "Machine Learning." It could be seen as one subjective factor for this biased phenomenon.

Besides, the diagonal values of "Others" and "Unknown" are weighty, revealing that the agreement for anti-domain knowledge between annotators is considerable. In contrast, we could infer that at least the annotators are agreeable to label the domain knowledge.

The **cohen's kappa** [147] is the most popular metric to measure the agreement between two raters. It is generally seen to be a more robust measure than a simple percent agreement calculation because it takes into account the probabilities of the agreement occurring by chance. In the Equation 3.2 of cohen's kappa, the p_e denotes the hypothetical probability of chance agreement, and p_o is similar to accuracy, which depicts the relative agreement between raters.

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{3.2}$$

For a simple example of Table 3.3, supposing that two volunteers are requested to rate for a case with "yes" or "no". In Table 3.3, the horizontal and vertical direction depicts the opinions of the two individuals. The value in the main diagonal of the matrix

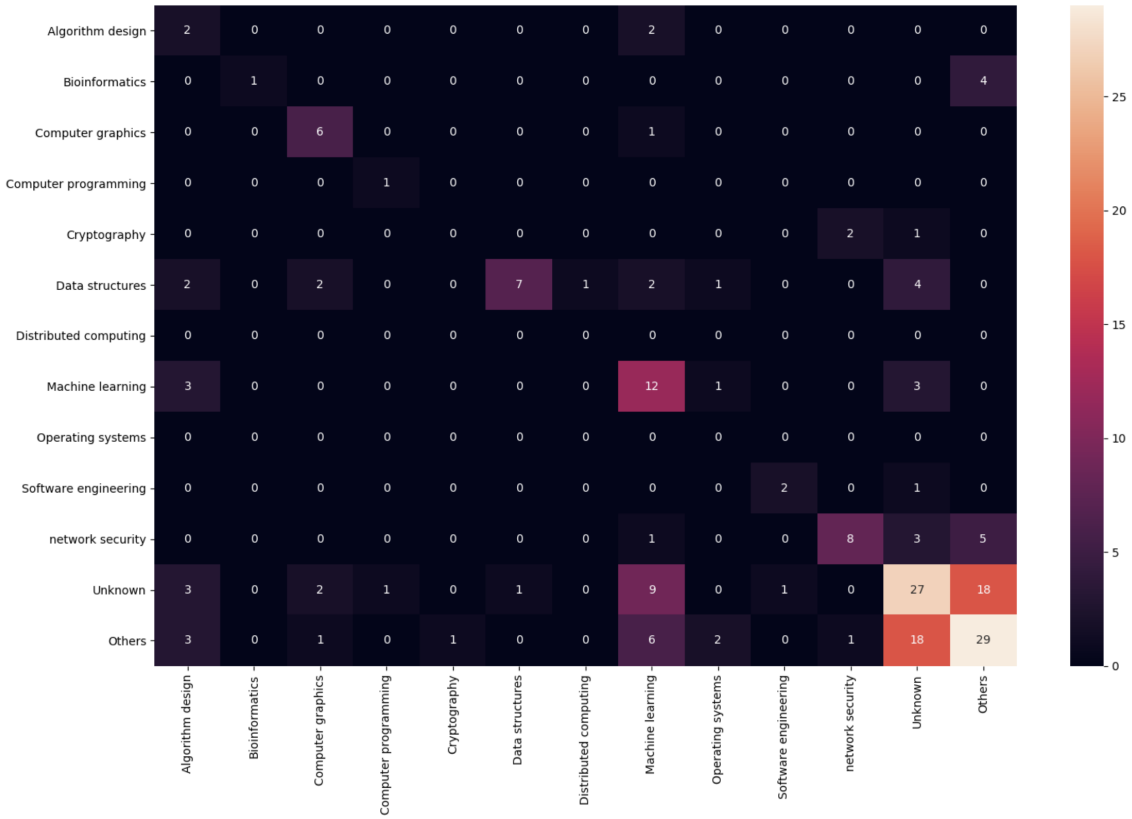


FIGURE 3.8 – The heat map between the practical Gold Standard and the labels of Annotator1.

Notes : The X-axis denotes the labels of Gold Standard, the Y-axis locates the labels of Annotator1. The value in the cell represents the agreements of the labels on two sides.

(a and d) count for the number of agreements and the off-diagonal value (b and c) count for the number of disagreements. The p_o and p_e could be calculated as :

$$p_o = \frac{a+d}{a+b+c+d} \tag{3.3}$$

$$p_e = p_{yes} + p_{no} = \frac{(a+b)\cdot(a+c)}{a+b+c+d} + \frac{(c+d)\cdot(b+d)}{a+b+c+d} \tag{3.4}$$

Overall there are two important metrics to measure the agreement degree between annotators, **the cohen's kappa** and **the agreement score**. The agreement score is also called the accuracy, which is identical to p_o of cohen's kappa. Unlike the simple example in Table 3.3, the real task of human's annotating always be asked to tag with multi-labels rather than solely the binary labels in the last example. Therefore, the confusion table could be applied with the multi-labels to present the agreement between two annotators. The agreement score could be calculated by dividing the sum of diagonal values (denoted as *DiagnalSum*) by the sum of all values (denoted as *AllSum*).

As shown in Table 3.4, the agreement degree is presented in the different metrics and different ranges of samples. Since the samples contain the domain-unrelated terms (labeled as "Unknown" and "Others"), the evaluation of the entire samples cannot completely reveal the agreement degree for the domain-related labels. It is meaningful to evaluate the agreements in the partial samples, which exclude the domain-unrelated terms. Accordingly, two different ranges of samples (i.e. entire samples and partial samples) with the same metrics are shown in Table 3.4. We notice that cohen's kappa coefficient is always lower than the agreement score in the same situation. The cohen's kappa takes into account the chance agreement, which lessens the entire value compared to the agreement score. Meanwhile, we observe that the range of samples does not influence a lot for the majority of annotators.

In general, the agreement ratio between annotators is less than a half, which means that human annotating work has a rather high individual relevance, even though they have the same knowledge background. To be simplified, in our case, we use the annotation work from only one rater.

TABLE 3.3 – An example to calculate cohen's kappa.

	yes	**no**
yes	a	b
no	c	d

TABLE 3.4 – The agreement degree between human annotators. A1, A2 and A3 is the identity for different annotators.

	entire samples			**partial samples***		
	A1	A2	A3	A1	A2	A3
kappa_score	0.34	0.31	0.29	0.36	0.16	0.22
agreement_score	0.47	0.41	0.46	0.46	0.26	0.44
DiagnalSum	95	82	91	26	14	8
AllSum	200	200	200	57	54	18

The difference between Gold Standard and Keywords

This section introduces the statistics of the Gold Standard and the extraction of Keywords for each corpus. The Gold Standard is the name of a list of terms that are examined by the domain experts, to guarantee that each term of Gold Standard is semantically close to one core concept. The Keywords are the terms attached to the documents that we acquire before dealing with the related corpus, which are regarded as the prior knowledge for the corpus.

There are two aspects to concern about in the formation of Gold Standard : 1) the terms of Gold Standard should conform to the pre-processed format. 2) a term is labeled by its core concept if they are closely related unless this term is discarded. As shown in Table 3.5, in the column of 'Gold Standard', we listed the number of terms for each core concept, where we have 1629 terms of Reuter's Gold Standard and 2150 terms of CS's Gold Standard. They are labeled based on the terms of the entire corpus, not only the sampled corpus.

We have different strategies to extract Keywords for the different corpus, because of the variance of the given resource. For Reuter corpus, the Keywords are initially extracted from the 90 topics [6], in which each topic contains several Keywords. After the pre-processing procedure over these Keywords, we only retain the Keywords that also appear in the final corpus. As for the Computer Science corpus, the Keywords, which are attached to the documents, are gathered together regarding the different

6. http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/a16-rbb-topic/topics.rbb

TABLE 3.5 – The Gold Standard and keywords of the whole corpus.

Corpus	Core Concepts	Gold Standard	Keywords
Reuter	company	603	62
	government	483	38
	economic	346	26
	commodity	197	8
		1629	**134**
Computer Science	Algorithm design	113	61
	Bioinformatics	92	84
	Computer graphics	378	81
	Computer programming	131	48
	Cryptography	179	70
	Data structures	258	67
	Distributed computing	172	75
	Machine learning	256	44
	network security	247	64
	Operating systems	139	54
	Software engineering	185	86
		2150	**734**

core concepts. We select the top-100 frequent Keywords for each core concept and transform them into a unified format. Likewise, the last step is to guarantee that the extracted Keywords also appear in the final corpus.

Even though the terms of Gold Standard and of Keywords seem to have similar semantics because of their relatedness to the core concepts, they should not be mixed up. In fact, they have big differences in the correctness and the utilities : 1) the terms of Gold Standard are evaluated by domain experts, while the terms of Keywords do not have this guarantee, however, at least they contain the terms that are related to the content of documents. 2) the Gold Standard is only used for evaluation at the end of the experiments, while the Keywords could be used as prior knowledge during the

experiments in Chapter 5.

3.3.3 Experiment Settings

This section introduces the details before executing the different experiments. We start by discussing how to control the variables of different clustering strategies, which includes the dimension of NPs representations and the number of clusters. Then we explore all the possible random settings in every step of the clustering procedure and determine the random scenario to run for experiments. Finally, we compare the size of clusters between different clustering strategies.

Practical Parameters

In our clustering framework, the combination of NPs representations and the clustering algorithms jointly contributes to clustering the terms, where both the dimension of NP representations and the number of clusters are the key variables. For the LDA-based clustering strategy, similarly, the size of the topic feature and the number of clusters is also the main variables.

In the practice, we prone to restrict the variables within a reasonable range, or even the equivalent settings as best as we can. For the NPs representation, the proposed co-occurrence representations and weighted co-occurrence representations use the number of VPCs as the size of their feature (please check the *#uniqueVPCs* column of Table 3.1), which is hard to be altered in the post-processing. However, for the NMF co-occurrence representation, word2vec embedding representation, and topic embedding representation, the dimension of NPs' features could be easily controlled by the prior setting. Therefore, we set the size of the feature to 100. To maintain consistency with the LDA-based clustering strategy, we set the number of topics to 100.

As for the clustering algorithm, we go through the number of clusters from 5 to 50 by step of 5. The practical library parameters of clustering experiments are shown in Table 3.6. It is worth mentioning that, affinity propagation and DBscan clustering algorithms are designed to find the optimal number of clusters, therefore, they are not required to go through different numbers of clusters as others.

TABLE 3.6 – The practical library parameters of the typical clustering algorithms.

algorithms	python library	function	k selection	other parameters
k-means	sklearn	cluster.KMeans()	5-50	default
k-medoids	sklearn_extra	cluster.KMedoids()	5-50	default
affinity propagation	sklearn	cluster.AffinityPropagation()	-	default
DBscan	sklearn	cluster.DBSCAN()	-	$eps = 0.2, min\_samples = 3$
co-clustering	sklearn	cluster.SpectralCoclustering()	5-50	$n\_init = 1$

Randomness of Clustering

For the same parameter setting, it is famous to train with multiple repetitions, so as to get convincing results for evaluation. Due to the existence of random effects among the repetitions, we need to ensure that different repetitions have different random states, so as to avoid totally identical repetitions.

In the classic clustering framework, the NMF co-occurrence representation, word2vec embedding representation, and topic embedding representation provide the possibilities to set random states, at the same time, all of the five clustering algorithms are also able to fix random states. We set 3 random states of NPs representations and 3 randoms states of clustering algorithms. In brief, we manage to experiment with 3*3 different random states for one parameter setting of clustering. In the same way, the clustering strategy based on LDA also needs to examine the 9 different random states for each experiment.

Size of Clusters

In our proposed term clustering framework, we present the clustering strategies of typical algorithms. Also, we propose the LDA-based clustering strategy. Due to the application of term cluster's thinning metrics in the latter work, the obvious difference lies in the size of clusters.

We present the averaged statistics of multiple repetitions here. As shown in Table 3.7, the total number of NPs in clusters (from 50 NPs to 500 NPS) is around 5 times less in the LDA-based clustering strategy than that of the classic clustering strategy. The relevant NPs are the terms that we can find in the Gold standard ; the irrelevant

TABLE 3.7 – The size of clusters in different clustering strategies.

Corpus	Clustering strategies	#NPs in Each Cluster	#clusters	#NPs in Clusters	% relevant NPs	% irrelevant NPs
Reuter	LDA-based strategy	<=10	from 5 to 50	from 50 to 500	77.47%	22.53%
	classic algorithms	No limitation	from 5 to 50	5043	32.30%	67.70%
Computer Science	LDA-based strategy	<=10	from 5 to 50	from 50 to 500	64.70%	35.30%
	classic algorithms	No limitation	from 5 to 50	5139	41.84%	58.16%

NPs are the terms that we cannot find in the Gold standard (the amount of GS is listed in Table 3.5). The percentage here is the averaged value when the number of clusters varies. The comparatively high percentage of relevant NPs indicates that the LDA-based clustering strategy is able to extract more domain-related NPs than other clustering strategies.

3.4 Evaluation of clustering strategies

In this section, the two major term clustering strategies (the classic clustering algorithms strategy and the LDA-based clustering strategy) were examined concurrently from various aspects.

Regarding the issue of choosing the number of clusters, we present the performance of these two clustering strategies along with the increasing number of clusters. Moreover, we also explore to select the optimal number of clusters of the classic strategy, under the consideration of the complexity arising from the various combinations of NPs representations and clustering algorithms.

Subsequently, we compare the optimal performance of these two clustering strategies under the same metrics on the same corpora. As for the framework of classic clustering algorithms, we provide a detailed analysis of the existing preference for the combination of the different NPs representations and the diverse clustering algorithms. Eventually, we summarize the comparison and highlight the most outperforming clustering strategy, which turns to be helpful for the ongoing research.

3.4.1 Influence of clusters numbers

The number of clusters is always an important factor for clustering performance. However, due to the complexity of the different combinations in the classic clustering strategy, it brings chaos if we analyze the influences on account of the entire framework of the classic term clustering. Given this, we distinguish two aspects from the classic clustering framework and study the influence of cluster numbers from these two aspects : NPs representations and the clustering algorithms.

We examine the performance of clusters regarding an important metric : micro precision. In a term cluster, we assume that the partial terms that belong to the most frequent label of this cluster, are regarded as the correct separation. The portion of correct separation is also called the purity ratio of a term cluster, which also is identical to micro precision (please see Section 2.4 for more information).

Along with the growing cluster numbers, the influence of the different clustering strategies are shown in Figure 3.9 and Figure 3.10 in the different corpora. In the sub-figures about the aspect of feature representations, except for the blue line, each other line presents the averaged micro precision of the five clustering algorithms and one specific NPs representation technique. In the sub-figures about the aspect of clustering algorithms, except for the blue line, each other line presents the averaged micro precision of the five NPs representation techniques and one specific clustering algorithm. It is worth mentioning that if the number of clusters could not be chosen within the range, the corresponding clustering algorithms would not be presented, i.e. Affinity Propagation and DBscan. The blue lines denote the averaged micro precision of the LDA-based clustering strategy.

In Figure 3.9, based on the Reuter corpus, the performance of clustering strategies is enhanced with the increasing cluster numbers. From the upper sub-figure (the aspect of feature representations), we notice that the performance of NPs representations aspect of the classic clustering strategy is close to each other. But we also remark that the topic embedding representation (NPs_lda) has better performance than others. The bottom sub-figure (the aspect of clustering algorithms) draws the same overwhelming line on the clustering based on LDA and it shows that, in general, the co-clustering algorithm is slightly better than other clustering algorithms. Notably, affinity propagation and DBscan clustering algorithms tend to fail in this range of clusters from 5 to 50. It is because these two algorithms are capable to choose the number of the cluster by their own computation.

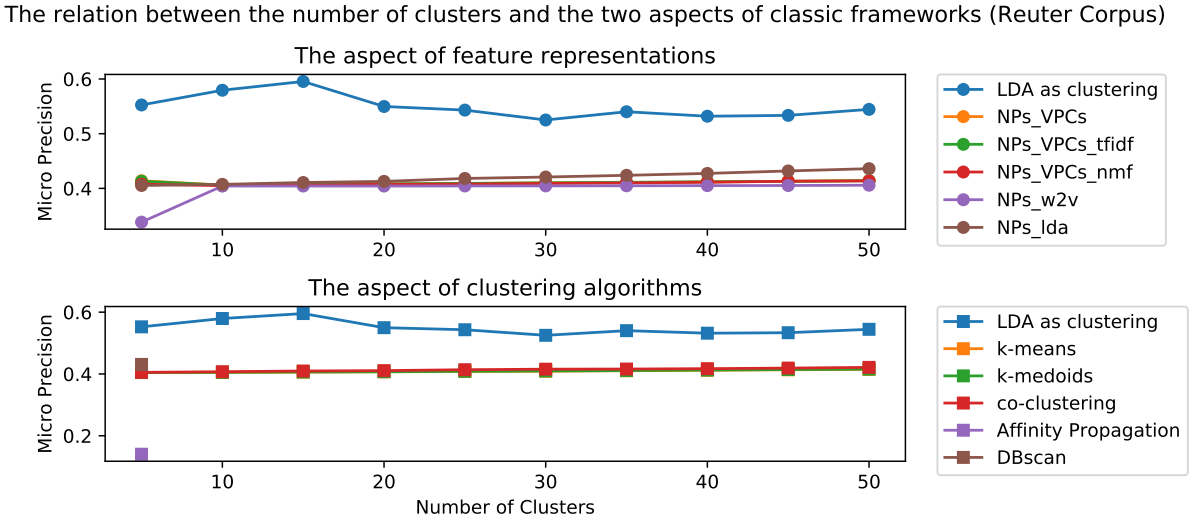


FIGURE 3.9 – The performance of clustering strategies with the increasing of cluster numbers (Reuter corpus).

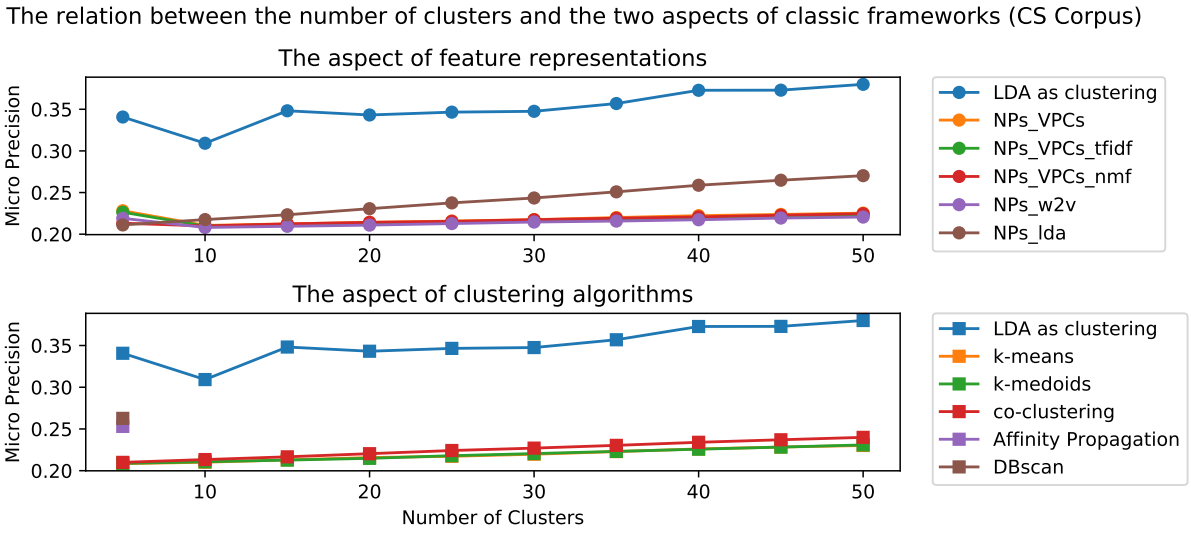


FIGURE 3.10 – The performance of clustering strategies with the increasing of cluster numbers (CS corpus).

In Figure 3.10, based on the Computer Science corpus, even though the general precision is lower than that of the Reuter corpus, but we could still notice the outstanding lines of the clustering strategy based on LDA. In addition to this, the other performances conform to that of the Reuter corpus.

In brief, even though the two clustering strategies achieve different precisions in the different corpus, their performances are improving with the rising cluster numbers. The LDA-based term clustering strategy shows an overwhelming precision over any of the two aspects of the framework of a classic clustering strategy. Meanwhile, the topic embedding representation (NPs_lda) turns to be the best NPs representation for term clustering.

3.4.2 The optimal cluster numbers

In the last section, we have analyzed the performance of the classic clustering framework into two aspects. For each separate aspect, it shows better performance with higher cluster numbers. However, we cannot conclude that the combined word representation and clustering algorithms will still have enhanced performance with a large number of clusters. In this part, we aim to explore the optimal cluster numbers of the entire classic clustering framework, with the combination of the two aspects.

In the experiments, to examine the performance of every possibility of the classic clustering framework, we go through the entire 5×5 combinations of the classic clustering framework. Each combination will be executed with all the number of clusters, from 5 to 50 with the step of 5. For each execution, we will study the resulting term clusters by micro precision, the same as that in the last section. Based on those abundant experiments, eventually, we can select the optimal cluster number of each combination, once the highest micro precision is achieved across all possible cluster numbers.

In Tabel 3.8, we summarize the optimal number of clusters regarding the combined NPs representations and clustering algorithms. Notably, it exists some failed combinations, which is signified with '$-$'. For the co-clustering algorithm, it fails to execute over the condensed co-occurrence representation. For the affinity propagation algorithm, only the word embedding representation has succeeded for execution. For the DBscan algorithm, word2vec embedding representation and topic embedding representation are failed. The failed combinations indicate that the calculation of the clustering algorithm cannot converge within a reasonable computation time, without human tuning

TABLE 3.8 – The optimal number of clusters regarding to the combined word representations and clustering algorithms

		NPs_VPCs	NPs_VPCs_tfidf	NPs_VPCs_nmf	NPs_w2v	NPs_lda
Reuter	k-means	35	40	20	25	25
	k-medoids	30	30	20	25	30
	co-clustering	40	30	-	25	35
	affinity propagation	-	-	-	10	-
	DBscan	186	163	225	-	-
Computer Science	k-means	35	35	35	25	25
	k-medoids	45	35	20	20	30
	co-clustering	40	30	-	25	35
	affinity propagation	-	-	-	168	-
	DBscan	220	207	42	-	-

and intervention. Hence we do not take into account those failed combinations.

As we have discussed, the affinity propagation and DBscan algorithm are capable to choose their own optimal numbers, and their choices are presented in Table 3.8. They prefer a larger amount of clusters than the pre-defined range of clusters. Besides, we notice that the optimal cluster numbers of other clustering algorithms fluctuate a lot, but still far from the ceiling of range (i.e. 50 clusters).

In conclusion, it is not true that the larger the number of clusters the better performance of the entire classic clustering framework. Still, the choice of optimal cluster number varies a lot, depending on the combination of word representation and clustering algorithms and on the corpus.

3.4.3 The comparison between the classic strategy of clustering framework and the LDA-based clustering strategy

Based on the optimal number of clusters that we have explored over the classic clustering framework, we would like to compare these two clustering strategies in the different evaluation metrics : the internal indices (i.e. silhouette width and Dunn index) are applied to measure the compactness and separateness of term clusters in their

geometric feature space ; the external indices (i.e. Macro precision, Micro precision, and Asymmetric rand index) are involved to evaluate the agreements between term clusters and the Gold Standard, please see details in Section 2.4.

Except for the evaluation of term clusters, it is also interesting to learn about the robustness of these two clustering strategies. Concisely, we prefer a more stable clustering strategy, who only brings little difference in various random states. Thus we also present the standard deviation of these clustering strategies.

The overall performance of the classic clustering framework

To simplify the presentation of these abundant experiment results, we draw the five figures regarding different metrics, from Figure 3.11 to Figure 3.15 on Reuter corpus. As for another corpus about Computer Science, their results are presented in Appendix 7.2 to save space in context. The data, that we applied in the figures, is calculated as the average value of multiple random states of one combination of the framework of classic clustering strategy, by using the corresponding optimal cluster number. The dashed cyan lines in the figures present the averaged value of the LDA-based clustering strategy with the number of clusters as 50.

Regarding the silhouette width score, -1 indicates the worst clustering, while 1 indicates the best clustering situation. The higher the score, the clusters are more prone to reach the high compactness and separateness. From Figure 3.11, we notice that k-means and k-medoids have better performance, whereas the k-means clustering is slightly better than that of k-medoids ; and the performance of co-clustering algorithm fluctuates a lot with different NPs representation, which implies that the performance of co-clustering algorithm relies more on the combined NPs representations. As for the NPs representations, the word2vec embedding representation has the worst performance in compactness and separateness.

For the Dunn score, I computed in the Davies-Bouldin's algorithms [7], where 0 indicates the minimum score. The lower the score, the clusters are more prone to reach the high compactness and separateness. From Figure 3.12, we can induce a similar analysis as that of silhouette width.

The macro precision denotes the portion of pure clusters, in which 0 means all clusters contain the terms with mixed labels and 1 means all clusters are pure clusters.

7. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score.html

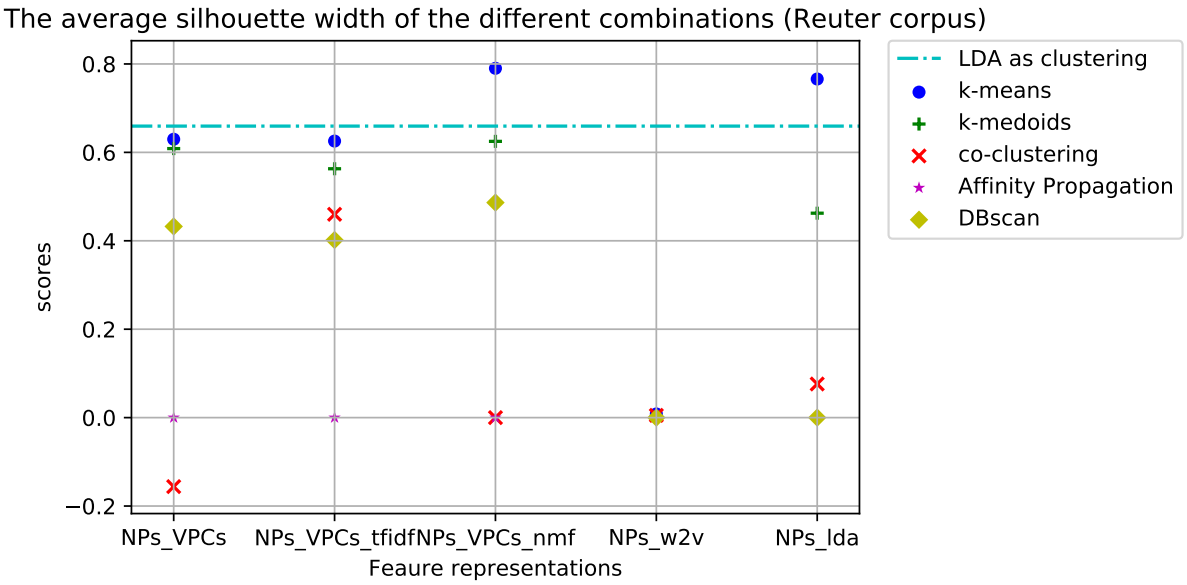


FIGURE 3.11 – The silhouette width of the two clustering strategies(Reuter corpus)

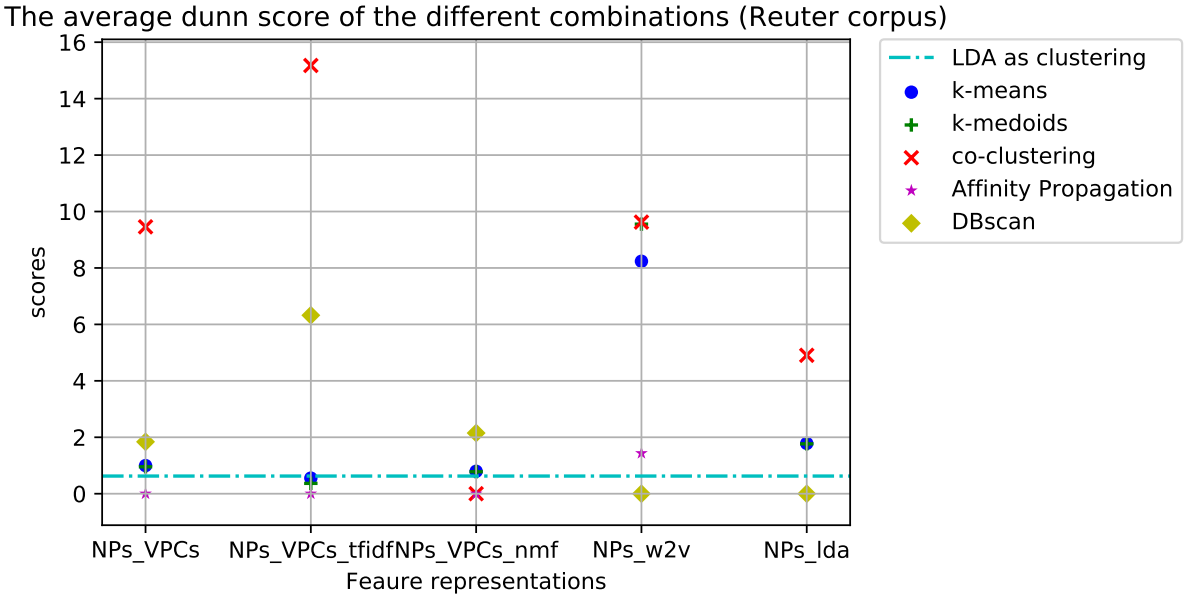


FIGURE 3.12 – The dunn score of the two clustering strategies(Reuter corpus)

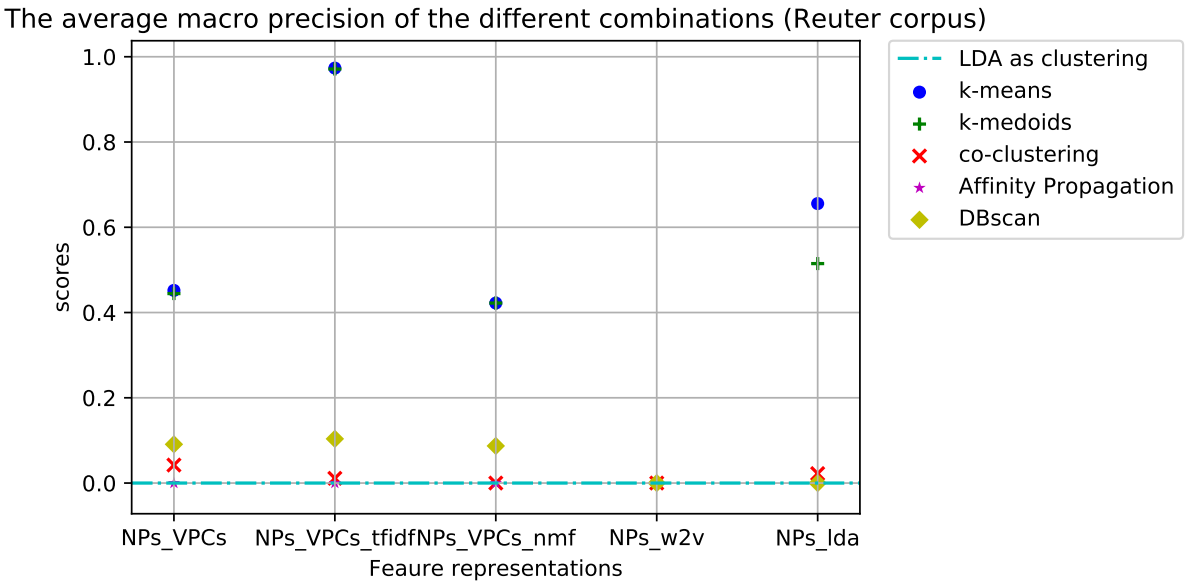


FIGURE 3.13 – The macro precision of the two clustering strategies(Reuter corpus)

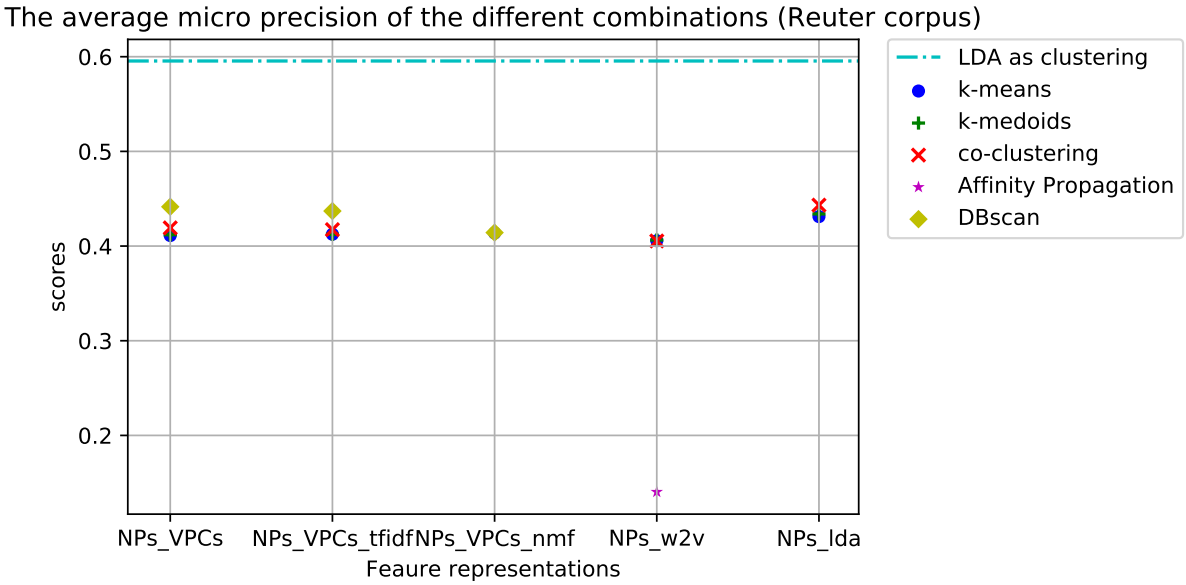


FIGURE 3.14 – The micro precision of the two clustering strategies(Reuter corpus)

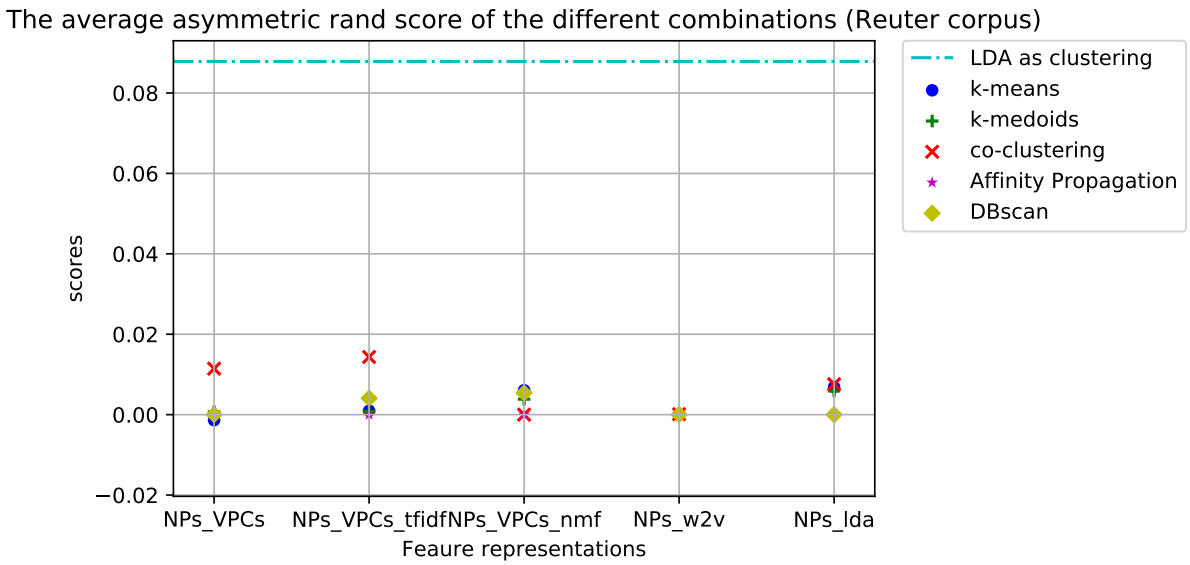


FIGURE 3.15 – The asymmetric rand score of the two clustering strategies(Reuter corpus)

The higher value the more pure clusters. In Figure 3.13, most of the clustering algorithms approach 0, except for k-means and k-medoids clustering algorithms. In fact, it exists an exceptionally high score for k-means algorithms. This phenomenon could be caused by abundant scattered pure clusters.

The micro precision denotes the averaged purity of all clusters, in which 0 means all terms of a cluster are not from the Gold Standard, and 1 means all terms of a cluster belong to one label. The higher the more agreement between clusters and Gold Standard. In Figure 3.14, the co-clustering and DBscan algorithms show better performance than others. However, the higher micro precision score is mainly due to a large number of clusters of DBscan algorithms. In general, the topic embedding representation (i.e. NPs_lda) reaches higher scores than other representations. Furthermore, the proposed co-occurrence representation and the weighted co-occurrence representation show better performance than the NMF co-occurrence representation, or than the word2vec embedding representations. Particularly, the combination of topic embedding representation and the co-clustering algorithm achieves the highest micro precision than other combinations of the framework of a classic clustering strategy.

The asymmetric rand score denotes the agreement between clusters and Gold Standard, in which 1 means the equivalence between clusters and the partitions of Gold Standard. The higher the more agreement between clusters and Gold Standard.

In Figure 3.15, the analysis accords to that of the micro precision. Briefly, even though the co-clustering algorithm achieves the poor compactness and separateness of clusters, its clusters reach a rather high agreement to the Gold Standard.

The benefit of the LDA-based term clustering strategy

Based on the same evaluation metrics, we would like to provide an obvious comparison between the classic term clustering strategy and the LDA-based term clustering strategy. Before the contrast, it is worth mentioning the difference in the parameter settings of these two clustering strategies. On the one hand, we select to present the performance of a classic clustering strategy with their corresponding optimal cluster numbers. On the other hand, for the LDA-based strategy, we apply the fixed ceiling number of clusters (i.e. 50 clusters) to be compared with. In addition, we utilize the standard deviation to measure the robustness of these two clustering strategies.

In the figures of last part, e.g from Figure 3.11 to Figure 3.15, we remark the dashed blue line appeared in the evaluation of each metric. These lines show the corresponding value of different metrics for the term clustering strategy based on LDA. We notice that the LDA-based strategy achieves a rather better performance in most of the metrics, except for the macro precision score. The 0 value of macro precision implies that it does not exist any pure clusters from the generated 50 clusters.

As for the robustness, in Table 3.9, we present the standard deviation regarding each aspect of clustering strategies. Due to the failure of multiple experiments on affinity propagation and DBscan, their standard deviation will not be taken into account. In this way, the LDA-based term clustering strategy is proved to be a more robust clustering strategy than others.

We could conclude that the LDA-based strategy could produce clusters with a similar degree of compactness and separateness as that of k-means clustering algorithms. However, the LDA-based clustering strategy can generate the term clusters with a higher agreement degree to the Gold Standard.

3.5 Summary

We have presented the framework of classic term clustering strategy in order to obtain the desired term clusters, where the clusters could be considered as concepts

TABLE 3.9 – The deviation for different clustering strategies

		NPs_VPCs	NPs_VPCs_tfidf	NPs_VPCs_nmf	NPs_w2v	NPs_lda
classic term clustering	Reuter corpus	0.0332	0.0823	0.0281	0.1538	0.0655
	CS corpus	0.0316	0.0776	0.0349	0.0130	0.0454
		k-means	**k-medoids**	**co-clustering**	**AP**	**DBscan**
	Reuter corpus	0.0427	0.0529	0.0502	0.5575	0.0061
	CS corpus	0.0373	0.0530	0.0387	0.0082	0.0060
LDA-based term clustering strategy	Reuter corpus			0.0273		
	CS corpus			0.0276		

of the ontology to build. During this step, we examine their performances with the rising cluster numbers. We discovered it is not true that the larger the number of clusters the better performance of the entire classic clustering framework. Still, the choice of optimal cluster number varies a lot, depending on the combination of word representation and clustering algorithms and on the corpus.

In parallel to the classic term clustering strategy, we proposed new NPs representation techniques in linguistic perspective (i.e. the proposed co-occurrence representations and their variants) and in statistic perspective compared to word2vec embedding representations and the topic embedding representations. Then we explored the preferences of those NPs representations for their suitable clustering algorithms. We found that even though the co-clustering algorithm achieves the poor compactness and separateness of clusters, their clusters reach a rather high agreement to the Gold Standard. Also, the topic embedding representation (i.e. NPs_lda) reaches a higher micro precision than other representations. Furthermore, the proposed co-occurrence representation and the weighted co-occurrence representation show better performance than the NMF co-occurrence representation, other than the word2vec embedding representations. Moreover, the combination of topic embedding representation and the co-clustering algorithm is considered as the optimal pair of the framework of classic term clustering strategy, which satisfies the need to possess the high precision modules of modular ontologies.

Comparatively, we also introduced a clustering strategy of applying the topic model LDA directly for term clustering. We found that the LDA-based clustering strategy

shows an overwhelming precision than any of the two aspects of the classic term clustering strategy. Still, it could produce clusters with a similar degree of compactness and separateness as that of k-means clustering algorithms (which is in the best situation of classic clustering framework). Particularly, the LDA-based term clustering strategy can generate the term clusters with a higher agreement degree to the Gold Standard. As for the robustness, it is proved to be a more robust clustering strategy than any aspects of the framework of the classic term clustering strategy.

From the results of this chapter, we notice that LDA could assist to achieve better performance in term clustering, not only as of the feature presentations of the classic clustering framework but also as the LDA-based clustering strategy. Therefore, we continue to optimize the LDA-based clustering strategy, which allows us to build a modular ontology where the label of each core concept will be the label of a module of the ontology to build.

It should be noted that the partial work in this chapter was first published at KEOD conference [140]. The conference paper was then extended into a chapter book to be published in Springer [148].

DEUXIÈME PARTIE

Contribution : Semi-supervised Modular Ontology Learning with Topic Modeling Driven by Core Concepts

CHAPITRE 4

RELATED WORKS ABOUT LDA

As we have mentioned in the last chapter, applying LDA has been proved to be profitable as a technique of term clustering towards ontology learning. Now we plan to discuss the concrete algorithms of LDA from the aspect of topic models. This chapter describes the different phases of topic models and provides the different metrics to evaluate their performance. More interestingly, we would like to explore the extensive LDA models, which are assisted by some prior knowledge to improve the topics towards the concrete concepts.

It starts with the basic hypothesis and the corresponding notations for topic models in Section 4.1. We discuss the interesting surveys and tools for topic models in Section 4.2. Then we specify the typical topic models in Section 4.3, from the simple statistical models, i.e. Latent Semantic Indexing algorithms(LSI), to the simple latent variable models, i.e. probabilistic Latent Semantic Indexing (pLSI), and finish with the Latent Dirichlet Allocation (LDA). Section 4.4 then details the extensive LDA models which make use of the seed information as prior knowledge and provides the summary of the mentioned models. After that, we investigate the LDA-based approaches for ontology learning in Section 4.5. Finally, in Section 4.6, we present the evaluation methods in the topic model aspect and the term clustering aspect to answer that : what are the language model evaluation metrics and how to apply the statistical evaluation metrics.

4.1 Specific Notation and Terminology

In the background of text mining techniques, we assume that :

- a document is represented as the bag of its terms, regardless of grammar and word order, but keeping counts. It is well known as **bag-of-words assumption** by Zellig Harris [4] ;
- An essential unit in a text document, e.g., a single term, is regarded as a **token**.

TABLE 4.1 – The notions of topic models.

Models	Symbol	Meaning
pLSI	w_i	the i-th word in vocabulary V
	d_j	the j-th document in the corpus D
	t	a topic
	V	the vocabulary of corpus
	D	the corpus including documents
	T	the total topics
LDA	α	dirichlet prior for document-topic distribution
	β	dirichlet prior for topic-word distribution
	θ_d	document-topic distribution to generate document d
	ϕ_t	topic-word distribution to generate words respecting to topic t
	z	the latent topic of a word
z-labels	C_i	the set of possible topicIDs for the word w_i
	η	soft constraint value of applying C_i
Seeded LDA	β^s	dirichlet prior for topic seed distribution ϕ^s
	β^r	dirichlet prior for topic word distribution ϕ^r
	ϕ^s	topic seed distribution, where seed represent seed words
	ϕ^r	topic word distribution
	π_t	a parameter to control the portions of ϕ^s and ϕ^r contributing to a topic t
	s	the seed set
	ψ_s	group-topic distribution of seed set s
	ζ^d	the document-topic multinomial parameter of document d

According to the bag-of-words assumption, the text could be parsed into different units, where the unit could comprise a contiguous sequence of n tokens, denoted as *n-grams*, and store the term frequency of n-grams as before. For example, the noun phrases (NPs) and verbs preposition compositions (VPCs) belong to the category of n-grams

- The classic input for various text models is **word-document matrix**, which records the frequency of words that occur in a collection of documents.

Let us consider the topic models :

- a $word$ of a document is the basic unit of discrete data, from the vocabulary indexed by {1,...,V}. A vocabulary is a dictionary constituted by a set of unique words or other strings extracted from text, which are arranged in order. The v-th word in the vocabulary is represented by V-vector w such that $w^v = 1$ and $w^u = 0$ for $u \neq v$;
- a $document$ is a sequence of N words, denoted by $\boldsymbol{d} = (w_1, w_2, ..., w_N)$;
- a $corpus$ is a collection of M documents denoted by $\boldsymbol{D} = (\boldsymbol{d_1}, \boldsymbol{d_2}, ..., \boldsymbol{d_M})$;
- a topic model, e.g. Latent Dirichlet Allocation (LDA) [87], considers a semantic representation that documents are represented as random mixture over $latent\ topics$ among $\{1, ..., \mathcal{Z}\}$, where each topic z is characterized by a distribution over $words$. More parameters could be found in Table 4.1

4.2 The overview of topic models

The main intention of topic modeling is to learn a model explaining the co-occurrence of terms in the documents. To undercover the connection between terms and documents, the topic is introduced as the intermediate bridge. Thus, in a topic model, terms can be working with documents and these documents are mixtures of topics, where a topic is a probability distribution over words. To be specific, a topic model executes a simple probabilistic procedure, by which documents can be generated.

There are many interesting surveys to summarize the development of topic modeling algorithms. In a systemic sort of view, *Alghamdi et Alfalqi 2015* [149] divided the area of methods into two sections : Bag-Of-Words topic Model and Topic Evolution Model. The former kind of model includes the typical topic models based on the Bag-Of-Words assumption, e.g. LSI [52], pLSI [150] and LDA [9], while the latter kind of model considers an important factor time, allows identifying topics with the appea-

rance of time and checks their evolution with time. For example, it has introduced Topic Over Time(TOT) [151], Dynamic Topic Models(DTM) [152], multiscale topic tomography [153], et al.

In a normal way, many surveys of topic models focus on discussing the application over text corpus, however, *Chen, Thomas et Hassan 2016* [154] extended with an in-depth discussion on the reliable application of the software un-structural data over the topic model, which is generated during software engineering. For instance, the un-structural data raised by the software includes the source code, emails, requirements or design documents, execution logs, bug reports, et al. Their work not only presented a broad study on the different topic models (i.e. many diverse variations of LDA) but also summarized the different facets concerning the practical usages of those models (i.e. target tasks and evaluation directions).

In the following sections, we concentrate on discussing the typical topic models and the variations of LDA with seed information, without the consideration of the time factor.

4.3 The typical topic models

4.3.1 Simple Statistical Models

This section introduces the development to uncover the semantic meaning of documents from the algebraic manipulation with the simple statistical models.

There are usually many literal terms to express one given conceptual topic ; however, one single term is deemed to be unreliable to represent a conceptual topic or meaning of a document [52]. To overcome the problem of retrieving conceptual content from documents, *Latent Semantic Indexing* (LSI) [51], [52] was proposed to reform documents representation into low dimension latent semantic space to capture the implicit association between terms and documents. Gradually, it is also well known as a more general terminology, *Latent Semantic Analysis* (LSA).

Based on the distributional hypothesis, words that are used and occur in the same contexts tend to have similar meanings [4]. LSA embarks on establishing associations between those terms that occur in similar contexts. Then it uses linear algebra and singular value decomposition (SVD) [155] to identify a linear subspace of the TF-IDF feature that captures the most of meanings in documents collection.

Formally let $\boldsymbol{A}$ be the $i \times j$ term-document matrix of a collection of documents with

TF-IDF as a weighing scheme, then

$$\boldsymbol{A}[i,j] = tf(i,j) * idf(i,D) = \begin{cases} f_{i,j} \cdot \log \frac{|D|}{|\{d_j \in D : w_i \in d_j\}|}, & f_{i,j} \neq 0 \\ 0, & f_{i,j} = 0 \end{cases} \tag{4.1}$$

,where the first component(term frequency $tf(i,j)$) computes how frequently a term occurs in a document and $f_{i,j}$ denotes the raw count of term i occurs in document j ; the second component(inverse document frequency $idf(i,D)$) computes how much a term is common or rare across all documents. Within the second component, the $\mid D \mid$ is the total number of documents in the corpus and $\mid \{d_j \in D : w_i \in d_j\} \mid$ is the number of documents where the term i appears [156].

The matrix $\boldsymbol{A}$ is usually large and sparse, the dimension reduction technique turns to be essential. The SVD could be performed by approximating term-document matrix $\boldsymbol{A}$ into three other matrices — an i by r term-concept matrix $\boldsymbol{T}$, an r by r singular values matrix $\boldsymbol{S}$, and a j by r concept-document matrix $\boldsymbol{D}$, which satisfy the following relations :

$$\mathbf{A} = \mathbf{USV^T} \tag{4.2}$$

In LSI, we assume that the singular value is too small to be negligible, thus replaced by 0. Let us say that we only keep the first-k singular values in $\boldsymbol{S}$. As such, we can reduce matrix $\boldsymbol{S}$ into $\boldsymbol{S_k}$ which is an $k \times k$ matrix containing only the k singular values that we keep, and also reduce $\boldsymbol{U}$ and $\boldsymbol{V}$ into $\boldsymbol{U_k}$ and $\boldsymbol{V_k}$, to have k columns and rows, respectively.

$$\mathbf{A} \approx \mathbf{A_k} = \mathbf{U_k S_k V_k^T} \tag{4.3}$$

Intuitively, the k remaining ingredients of the eigenvectors in $\boldsymbol{U}$ and $\boldsymbol{V}$ correspond to k "hidden topics," in which it preserves the most important semantic information in the text.

In the perspective of the dimension reduction, the large collections carry out significant compression over the term document matrix through $\boldsymbol{V_k}$ and it maps document into a low dimensional space, but it alleges difficulty in determining the optimal number of dimensions to use for performing the SVD. The number cannot be chosen to arbitrary numbers, but it depends on the rank of the matrix and cannot go beyond that. Furthermore, once the new documents appear, the calculation of SVD requires intensive efforts of computation, and eventually the model is hard to update.

From the perspective of semantic meaning, LSI can retrieve synonyms (i.e. a term means nearly the same to another) of the query terms from $\boldsymbol{U}_k$. However, it raises the difficulties of interpreting the dimensions.

4.3.2 Simple latent variable models

From the last section, the algebraic approaches aim to uncover the semantic relationship between terms and documents, whereas the previous statistical models are not preparing to handle the large collections of documents. To organize, understand, search, and summarise those text documents, the hidden themes of documents could be explored to annotate documents and use those annotations to manage amounts of documents. Technically, according to *Blei, Ng et Jordan 2003* [9], we can begin with generating text documents from the given topics, expecting that the outcomes fit the observed documents. Thereby the topic assignments turn to be apparent to those original collections.

Initially, as the simplest generative process, a document could be constituted by words drawn independently from a single multinomial distribution, denotes as $w \sim Mult(p)$. The probability of this document is :

$$p(d) = \prod_{n=1}^{N} p(w_i) \tag{4.4}$$

Secondly, before generating a document, choose a topic from $z \sim Mult(p)$, then its component words will follow the conditional multinomial distribution independently, $w \sim Mult_z(p)$ [157] . While it has the limitation that each document exhibits exactly one topic.

$$p(d) = \sum_{z} p(z) \prod_{i=1}^{V} p(w_i \mid z) \tag{4.5}$$

In the third place, the *probabilistic Latent Semantic Indexing* (pLSI)proposed by *Hofmann 1999* [150], extends LSI assuming that each document is a probability distribution over topics and each topic is a probability distribution over words. The pLSI model uses the statistical latent variant for factor analysis of the raw count in the term-document matrix, where Figure 4.1 displays the matches between matrix resolving and

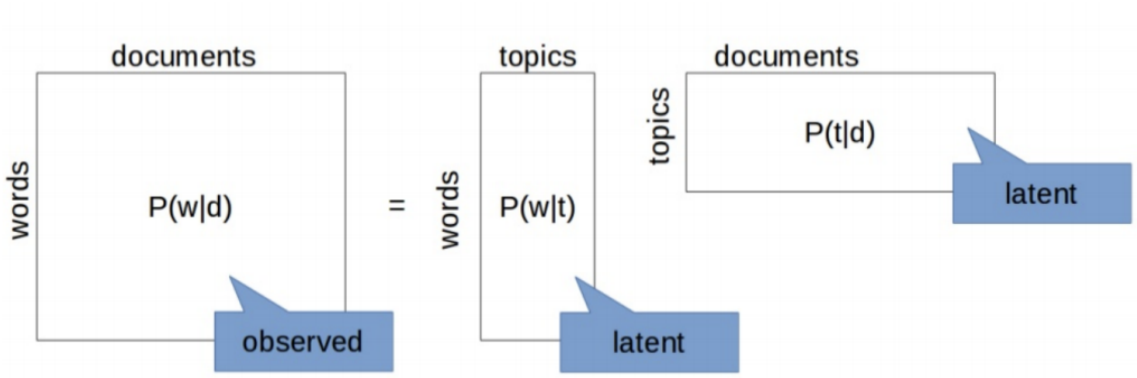


FIGURE 4.1 – The matrix resolving of word document distribution.

the probabilities expression ; it could also be written as :

$$p(w \mid d) = \sum_{z \in \mathcal{Z}} p(w \mid z) p(z \mid d) \tag{4.6}$$

it is worthy noting that the equivalent symmetric version can be obtained by inverting the conditional possibility $p(z \mid d)$ with the help of Bayes' rule, in which a document and a word are conditionally independent given a latent topic z :

$$p(w, d) = p(d) p(w \mid d) = \sum_{z \in \mathcal{Z}} p(z) p(w \mid z) p(d \mid z) \tag{4.7}$$

The application of pLSI handles the problem of polysemy. However, different from the previous generative process, pLSI treats topics as term distribution and uses probabilistic methods instead of matrices. It captures the possibility that a document may contain multiple topics. It must be pointed out that this model is prone to overfitting (being corpus specific) and a linearly increasing number of parameters that need to be estimated, with the inclusion of extra training documents [87].

4.3.3 Latent Dirichlet Allocation

The basic ideas behind *Latent Dirichlet Allocation* (LDA) are to discover the topics from a collection of documents automatically. It is most easily described as a statistical model by the intuition of the words' occurrence within documents. Contrary to this foresight, this model assumes that the topics are generated first, and then documents are estimated from the generative process that includes *hidden variables* [6].

LDA assumes that each word w_i is associated with a latent topic t. Each of these topics $t = 1, ..., T$ is associated with a multinomial ϕ_t over the V-word vocabulary, each ϕ is drawn from a Dirichlet prior with parameter β. Likewise, each document $j = 1...D$ is associated with a multinomial θ_j over topics, drawn from a Dirichlet prior with parameter α. The full generative procedure is then

I For each topic $t = 1, 2, ..., T$

(a) Sample topic-word multinomial $\phi_t \sim Dirichlet(\beta)$

II For each document $j = 1, 2, ..., D$

(a) Sample document-topic multinomial $\theta_j \sim Dirichlet(\alpha)$

(b) For each word w_i in document j :

i. Sample topic $z_i \sim Multinomial(\theta_j)$

ii. Sample word $w_i \sim Multinomial(\phi_{z_i})$

The procedure implies a joint probability distribution over the random variables $(\mathbf{w}, \mathbf{z}, \phi, \theta)$, which is given by

$$P(\mathbf{w}, \mathbf{z}, \phi, \theta \mid \alpha, \beta, \mathbf{D}) \propto (\prod_{\mathbf{t}}^{\mathbf{T}} \mathbf{p}(\phi_{\mathbf{t}} \mid \beta))(\prod_{\mathbf{j}}^{\mathbf{D}} \mathbf{p}(\theta_{\mathbf{j}} \mid \alpha))(\prod_{\mathbf{i}}^{\mathbf{V}} \phi_{\mathbf{z_i}}(\mathbf{w_i})\theta_{\mathbf{d_i}}(\mathbf{z_i})) \tag{4.8}$$

, where $\mathbf{w}$ and $\mathbf{z}$ are two coupled vectors of terms respecting to topics of the same size ; $\mathbf{w}$ represents a document as a vector of words w_i, and $\mathbf{z}$ represents the vector of topics z_i of words w_i at the same position/index i in document j ; z_i is the latent topic associated with the i-th term in the corpus, $\phi_{z_i}(w_i)$ is the w_i-th element in vector ϕ_{z_i} , and $\theta_{d_i}(z_i)$ is the z_i-th element in vector θ_{d_i} . The conditional dependencies implied by this distribution can be represented by the directed graphical model shown in Figure 4.2.

There is a step-by-step example of the generative process designed by *Andrzejewski, Craven et Zhu 2010* [158], as shown in Figure 4.3. Subfigure (a) presents an elementary vocabulary with three terms inside, and subfigure (b) indicates that we set to 3 topics with the concrete value for each hyper-parameter α and β. Following the generative process that we discussed above, we first sample a topic-word multinomial ϕ_z for each topic in subfigure (c), then we sample document-topic multinomial θ_d in subfigure (d). To present the probabilities more straightforwardly, these ϕ vectors and β vectors could be plotted graphically on a simplex. In the subfigure (e) and (f), the

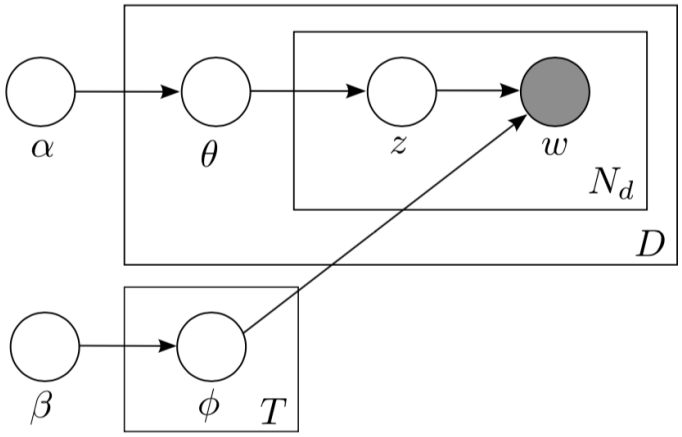


FIGURE 4.2 – The directed graphical model of LDA.

nearness of each ϕ_i to each corner is proportional to the probability that ϕ_i places on their corresponding word. For more detail, in subfigure (e), the ϕ_1 possesses a rather high probability on $word = 3$ and the location of ϕ_1 is close to $P(w = 3)$ comparing to the other words. The same idea suit for subfigure (f) as well, in which θ_1 approaches to $P(z = 3)$ and θ_2 approaches to $P(z = 2)$.

From the generative process of LDA, we observe that the LDA uses dirichlet priors for the document-topic and topic-word distributions. The inclusion of dirichlet priors prevents over-fitting effects. However, it seems to be unable to model further relations among topics.

Given the observed words w, the major goal for calculation is the inference of the hidden topics z. However, computing the conditional distribution of the topic structure given the observed documents (also called the *posterior*) is intractable. Various inference techniques have been developed to approximate the posterior. The topic modeling algorithms generally fall into two categories– sampling-based algorithms and variational algorithms.

The sampling-based algorithms attempt to collect samples from the posterior to approximate it with an empirical distribution [6]. We resort to a Markov Chain Monte Carlo (MCMC) sampling scheme, specifically Collapsed Gibbs Sampling [159].

Alternatively, the variational algorithms are a deterministic methodology for approximating likelihood and posteriors [6], [160]. It reformulates the problem of computing the posterior distribution as an optimization problem. Variational inference algorithms generally perform worse but run faster than sampling-based algorithms. Thus, variational

	1	Dog
w	2	Run
	3	Cat

(a) Example vocabulary.

T	3
α	0.5
β	0.1

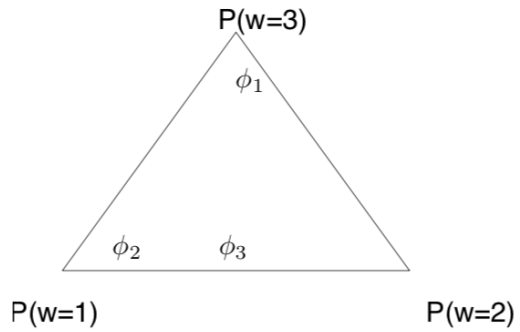
(b) Example parameters.

ϕ		w		
		1	2	3
	1	.1	.1	.8
z	2	.8	.1	.1
	3	.5	.4	.1

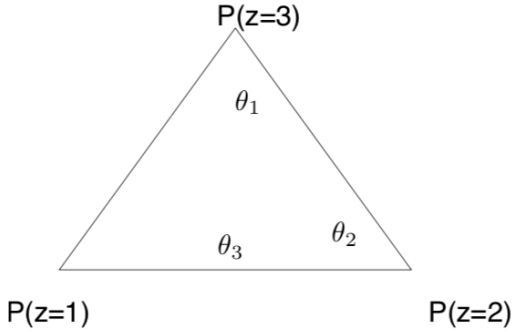
(c) $\phi_z \sim Dirichlet(\beta)$

θ		z		
		1	2	3
	1	.15	.15	.7
d	2	.15	.7	.15
	3	.5	.4	.1

(d) $\theta_d \sim Dirichlet(\alpha)$



(e) Simplex representation of ϕ.



(f) Simplex representation of θ.

FIGURE 4.3 – An example of the LDA generative process extracted from *Andrzejewski, Craven et Zhu 2010* [158].

inference suit well particular for large-scale documents.

Roughly speaking, both inference algorithms perform a search work over the topic structure. The choice of the inference algorithms depends on the size of the corpus and the expectation of the searching performance.

4.4 The extensive LDA models with seed information

Topic models, e.g. LDA, have emerged as a popular tool to analyze document collections in an unsupervised way. The LDA model's practical objective is to maximize the probability of the most obvious observed data but sacrifice the performance on rare topics. Intuitively, this results in a skewed topical impression of the corpus. However, this skewed topic is not following the underlying topical structure for human interpretation in most cases. We address this problem by guiding topic models to learn topics of specific interest to a user.

Here we consider the involvement of 'prior knowledge' to improve the topic relevant in LDA. And there are many different ways to introduce some prior knowledge, which include the non symmetric Dirichlet meta-parameters, must link and cannot link constraints, wordnet lexical properties, list of pre-labeled words (seeds), etc.

For instance, during the generative process of topics, in order to involve users' supervision, an interactive topic model [161] brings the user into the loop by allowing him/her to make suggestions on how to improve the quality of the topics at each iteration. Their approach uses the Dirichlet Forest method to incorporate the user's preferences.

Many other works also use external knowledge to operate at the level of tokens, documents, and pairwise constraints. In the level of tokens, *Andrzejewski et Zhu 2009* [10] proposed to apply the topic-in-set knowledge on LDA, but the seed information is provided manually. The word-level seed information can be converted into token-level information, but this prevents their model from distinguishing them based on the word senses. In the level of pairwise constraints, *Andrzejewski, Zhu et Craven 2009* [162] incorporated domain knowledge into topic modeling via Dirichlet forest priors, but the seed information is still required manually. Interestingly, he proposed to use Dirichlet forest priors to incorporate Must Link and Cannot Link constraints into the topic models so as to distinguish the similar and opposite clusters, which has the analogous concept with the constrained K-means clustering algorithm [163]. At the documents' level, many

works aimed to predict the category labels for the input documents based on document labeled data, i.e., the supervised topic models [164] and DiscLDA [165].

In the perspective of the seed information usage, *Thelen et Riloff 2002* [166] proposed to learn semantic lexicons using extraction pattern contexts. However, semantic information only focuses on specific notions (i.e., entities) but not general concepts. Apart from this, *Li, Roth et Sporleder 2010* [167] used sets of words for the word sense disambiguation task ; this model assumes that a topic is a distribution over synsets and relies on the Wordnet to obtain the synsets. The labeled LDA [168] can operate on a multi-class labeled corpus. Besides the operation in the document level, the z-labels LDA [10] deals with the multi-topics for each word so as to raise the accordance from words to topics. Besides, generating document topic distribution of the labeled LDA [168] is similar to generating group distribution to another model Seeded LDA [11]. As for Seeded LDA, it provides sets of seed words that a user believes in representing the underlying topics in a corpus. Those seeds are then used to improve both topic-word distributions by biasing topics to produce appropriate seed words and improve document-topic distributions by biasing documents to select topics related to the seed words they contain.

4.4.1 z-labels LDA

The target of z-labels LDA [10] is to use the topic assignments of seed words to guide the learning of topics in the corpus. Initially, it assumes that some seed words are already known and assigned to one or several topic IDs ; this is the prior knowledge provided by the domain expert according to the corpus's content. Let C_i be the set of possible z-labels for the word w_i. For instance, if we wish to restrict word w_i to a single value (e.g., topicID_of_w_i is equal to 5), this can now be accomplished by setting $C_i = \{5\}$. Likewise, we can restrict word w_i to a subset of topicIDs {1, 2, 3} by setting $C_i = \{1, 2, 3\}$. Otherwise, for the unconstrained word w_i , we simply set $C_i = \{1, 2, ..., T\}$.

To insert this prior knowledge into the learning process, it induces a boolean indicator function $\delta(v \in C_i)$, which takes value as 1 when $v \in C_i$, otherwise 0. This function will be used as a hard constraint by modifying the Gibbs sampling equation. The full conditional Gibbs sampling equation used for sampling individual z_i values from the

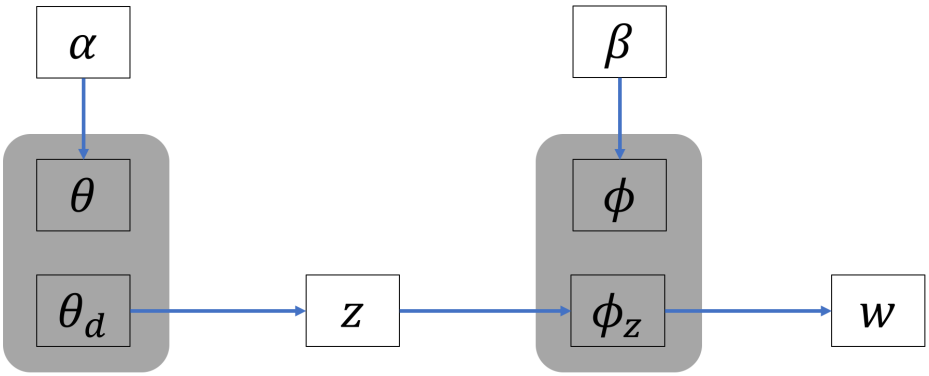


FIGURE 4.4 – The directed graphical model of LDA.

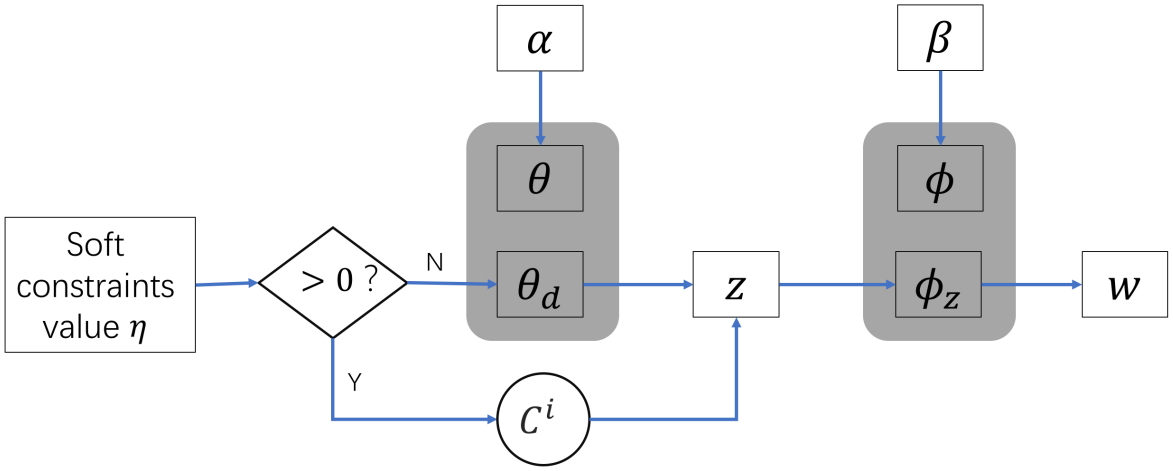


FIGURE 4.5 – The graphical model of z-labels LDA.

posterior is given by :

$$P(z_i = v|z_{-i}, w, \alpha, \beta) \propto (\frac{n^{(d)}_{-i,v} + \alpha}{\sum_u^T (n^{(d)}_{-i,u} + \alpha)})(\frac{n^{(w_i)}_{-i,v} + \beta}{\sum_{w'}^W (n^{(w')}_{-i,v} + \beta)}) \tag{4.9}$$

,where $n^{(d)}_{-i,v}$ is the number of times topic z is used in document d, and $n^{(w_i)}_{-i,v}$ is the number of times word w generated by topic z. The $-i$ notation signifies that the counts are taken omitting the value of z_i. In detail, if we consider the right part of Equation 4.9 as q_{iv}, then the modified Gibbs sampling equation could be :

$$P(z_i = v \mid z_{-i}, w, \alpha, \beta) \propto q_{iv} \cdot \delta(v \in C_i) \tag{4.10}$$

This modification makes the inference of latent topics become flexible, regarding the prior knowledge. Meanwhile, the hard constraints could be relaxed into the soft constraints. Let $0 \leq \eta \leq 1$ be the strength of our constraint, where $\eta = 1$ equals to the hard constraint Equation 4.10 and $\eta = 0$ equals to the original sampling Equation 4.9. Then this soft constraint equation will be :

$$P(z_i = v \mid z_{-i}, w, \alpha, \beta) \propto q_{iv} \cdot (\eta \cdot \delta(v \in C_i) + 1 - \eta) \tag{4.11}$$

The generative process of z-labels LDA is listed below :

I For each topic $t = 1, 2, ..., T$

  (a) Sample topic-word multinomial $\phi_t \sim Dirichlet(\beta)$

II For each document $j = 1, 2, ..., D$

  (a) Sample document-topic multinomial $\theta_j \sim Dirichlet(\alpha)$

  (b) For each word w_i in document j :

    i. If w_i is contained by seed words set :

      if $\eta \neq 0$:

        choose topic assignment z_i from the seeded topic set C_i

    ii. Sample topic $z_i \sim Multinomial(\theta_j)$

    iii. Sample word $w_i \sim Multinomial(\phi_{z_i})$

As a comparison, we draw the generative graph of basic LDA into Figure 4.4, on this basis, the generative graph of z-labels LDA could be drawn as Figure 4.5. It is obvious

that the z-labels LDA uses the topic assignments of seed words to guide the learning of topics in the corpus and it introduces the constraints of z-labels LDA to control the contribution of seed topic distribution. The results of z-labels LDA [10] shows that the performance is greatly affected by the topic assignments of seed words.

4.4.2 Seeded LDA

Similar to the z-labels LDA model, the Seeded LDA [11] aims to handle the learning of topics by applying the topic alignments of seed words. However, Seeded LDA has a rather sophisticated mechanism.

On the one hand, it benefits from two different **topic-term distributions** : one for *topic-seed distribution* ϕ^s (with the shape of $T \times S$, where S is the number of seed words), another for *topic-word distribution* ϕ^r (with the shape of $T \times V$, where V is the size of vocabulary). Each topic is a mixture of these two distributions, and the parameter π_t controls their portions contributing to a topic. For the part of topic-term distribution, the generative process of Seeded LDA is listed below (see Figure 4.6) :

I For each topic $t = 1, 2, ..., T$

(a) Choose regular topic $\phi_t^r \sim Dirichelet(\beta_r)$.

(b) Choose seed topic $\phi_t^s \sim Dirichelet(\beta_s)$.

(c) Choose $\pi_t \sim Beta(1, 1)$.

II For each document $j = 1, 2, ..., D$

(a) Sample document-topic multinomial $\theta_j \sim Dirichlet(\alpha)$

(b) For each word w_i in document j :

i. Select a topic $z_i \sim Multinomial(\theta_j)$.

ii. Select an indicator $x_i \sim Bernoulli(\pi_{z_i})$.

iii. If x_i is 0 :
Select a word $w_i \sim Multinomial(\phi_{z_i}^r)$.
//choose from regular topic

iv. If x_i is 1 :
Select a word $w_i \sim Multinomial(\phi_{z_i}^s)$.
//choose from seed topic

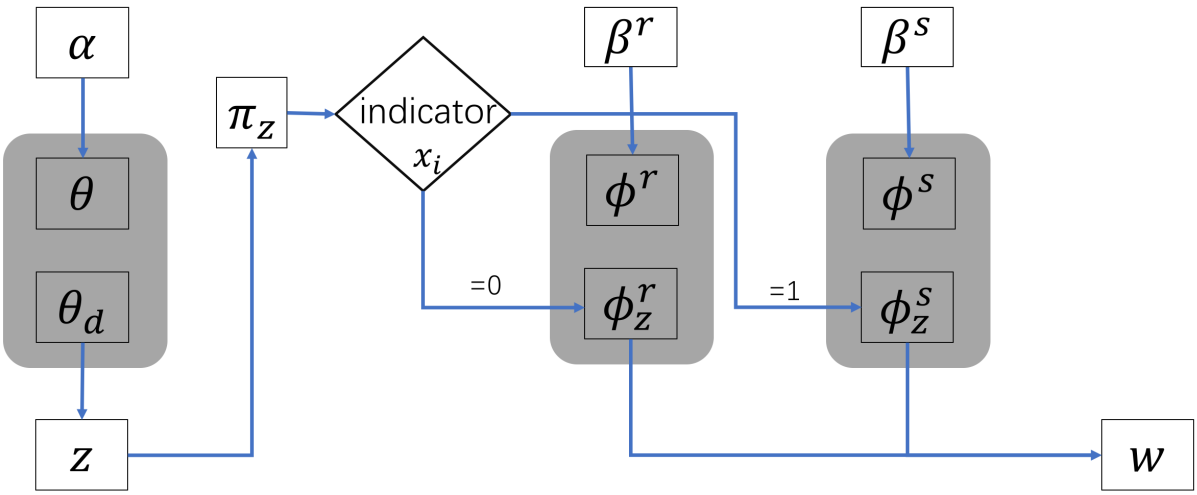


FIGURE 4.6 – The graphical model of topic-term distribution part of Seeded LDA.

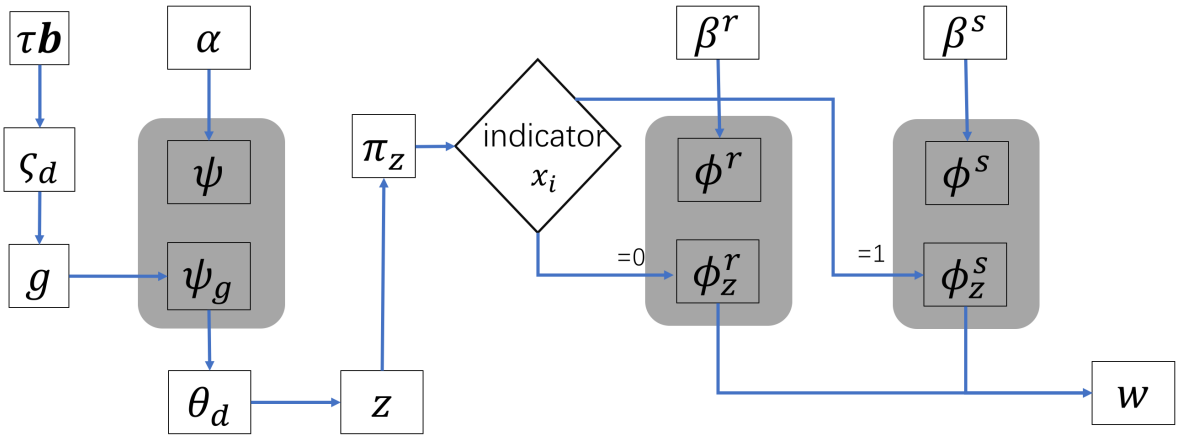


FIGURE 4.7 – The graphical model of document-topic distribution part of Seeded LDA.

In the previous process, we notice that the indicator x_i decides to choose to apply the topic-seed distribution or the topic-word distribution for a specific word. The binary decision is similar to the output of hard constraints of z-labels LDA [10] ; however, the major difference is that this model tried to consider the topic preference of its related document by using the Beta distribution in (I.c) and the Bernoulli distribution in (II.b.ii). While for the hard constraints, the binary decision only considers whether a word is a seed word. If so, it will apply all the topic distributions of this seed word. Additionally, the soft constraints of z-labels LDA take into account a certain portion of seed topic distribution, which could be considered a mixture of distribution, but this scenario never happens for this model.

On the other hand, this model also improves the **document-topic distributions** by biasing documents to select topics related to the seed words they contain. In the general case, the number of seed topics is not equal to regular topics. Hence, this model associate each seed set (here refer as g for *group*) with a multinomial distribution over the regular topics, which we call *group-topic distribution* ψ_g. In the next step, it will be considered prior to draw the document-topic distribution θ_d. In this way, this model allows a flexible number of seed and regular topics and connects the topic distributions of all the documents within a group.

For the integrated model of both topic-term distribution and document-topic distribution, the generative process of Seeded LDA is listed below (corresponding to Figure 4.7) :

I For each topic $t = 1, 2, ..., T$

  (a) Choose regular topic $\phi_t^r \sim Dirichelet(\beta_r)$.

  (b) Choose seed topic $\phi_t^s \sim Dirichelet(\beta_s)$.

  (c) Choose $\pi_t \sim Beta(1, 1)$.

II For each seed set $s = 1, , , S$,

  (a) Choose group-topic distribution $\psi_s \sim Dirichlet(\alpha)$.

III For each document $j = 1, 2, ..., D$

  (a) Choose a binary vector $\overrightarrow{b}$ of length S.

  (b) Choose a document-topic multinomial $\zeta^j \sim Dirichlet(\tau \overrightarrow{b})$.

  (c) Choose a group variable $g \sim Multinomial(\zeta^j)$.

  (d) Choose $\theta_j \sim Dirichlet(\psi_g)$.

(e) For each word w_i in document j :

i. Select a topic $z_i \sim Multinomial(\theta_j)$.

ii. Select an indicator $x_i \sim Bernoulli(\pi_{z_i})$.

iii. If x_i is 0 :

Select a word $w_i \sim Multinomial(\phi^r_{z_i})$.

//choose from regular topic

iv. If x_i is 1 :

Select a word $w_i \sim Multinomial(\phi^s_{z_i})$.

//choose from seed topic

The integrated Seeded LDA also uses the topic assignments of seed words to guide the learning of topics in the corpus. The results show that it has the ability to handle a situation with an unequal number of seeds and regular topics. However, they also found that allowing a seed word to be shared by multiple sets of seed words degrades term clustering performance. The computation complexity is considerably high.

After the discussion of the typical topic models and the variations of LDA accompanying with seed information, we present the comparison of these mentioned models with their strength and weakness in Table 4.2.

4.5 LDA based approaches for ontology learning

In the last two decades, topic modeling has been explored in various domains, i.e. text mining and information retrieval, however, applying topic modeling for ontology learning still needs more exploration.

Many works were interested in making use of the connection between terms, topics, and documents from LDA. There are two main directions to benefit from LDA for ontology learning : 1). to apply the knowledge of ontology to LDA for better term representations and optimize the ontology with the semantic-close terms ; 2). regardless of the existing ontology, to combine the word embedding representations and term topic representations in order to tackle the specific tasks of ontology construction, such as term clustering and term classification.

In the first direction, we would like to explain four typical works in detail, which are notated as 'LDA_WN' [170], 'LDA_Probase' [171], 'LDA_Probase2' [172] and 'LDA_CTM' [173] in Table 4.3.

TABLE 4.2 – The comparison table of topic models, adapted from *Barde et Bainwad 2017* [169]

Model	Strengths	Weaknesses
	Typical topic models	
LSI [52]	1). It can handle the problem of synonymy to some extent. 2). It maps documents to a low-dimensional space.	1). It depends heavily on SVD which is computationally intensive and hard to update as new documents appear. 2). The latent topic dimension can not be chosen to arbitrary numbers. It depends on the rank of the matrix, can not go beyond that.
pLSI [150]	1). It handles the problem of polysemy 2). It treats topics as term distribution and uses probabilistic methods instead of matrices	1). It is prone to overfitting, so the number of parameters increases linearly with the number of documents.
LDA [9]	1)It uses dirichlet priors for the document-topic and topic-word distributions for semantics discovery. 2)It prevents over-fitting.	1). It becomes unable to model relations among topics.
	Variations of LDA (with seed information)	
z-labels LDA [10]	1). It uses the topic assignments of seed words to guide the learning of topics in the corpus. 2). It introduces the constraints of z-labels LDA to control the contribution of seed topic distribution	1). the performance is greatly affected by the topic assignments of seed words.
Seeded LDA [11]	1). It uses the topic assignments of seed words to guide the learning of topics in the corpus. 2). It has the ability to handle a situation with an unequal number of seeds and regular topics.	1). the computation complexity is considerably high. 2). allowing a seed word to be shared by multiple sets of seed words degrades term clustering performance.

TABLE 4.3 – The summary of LDA based works for ontology learning

Models	Targets	Deployments of LDA	Deployments of ontology	Clustering algorithms	Results	Utilities
LDA_WN [170]	to distinguish the polysemy of terms in the WordNet taxonomy (WORDNET-WALK)	provide the probabilities between topics and the terms or concepts from wordnet-walk	use the WordNet taxonomy to calculate the closeness probabilities between concepts and embed these probabilities to LDA model	-	the closest taxonomic path of a term from the WordNet taxonomy is found, so as to deal with polysemy problems in interpretation	semantic labeling for terms
LDA_Probase [171]	to map terms to the concept from ontology	represent the topic by terms from text p(topic| term) and link the topic with the concepts from Probase p(concept|topic)	specify the probability of each term belonging to a concept from Probase ontology p(term|concept)	-	in the given context, the relatedness between a term and a concept from Probase could be measured respecting to the context	semantic labeling for terms; term similarity measurement
LDA_Probase2 [172]	to distinguish the polysemy of terms by the explicit concepts from ontology and the implicit concepts from topics, regarding context	explore the implicit concepts of terms	specify the probability of each term belonging to an explicit concept from Probase ontology p(term|concept)	k-Medoids	the explicit concepts has higher accuracy but low ârecallsâ, and the implicit concepts provide a complement, which could include the words (such as verbs and adjectives) that not exists in the ontology.	similarity measure between terms
LDA_CTM [173]	to automatically tag Web pages with concepts from an ontology without any need for labeled documents.	modify the basic LDA, and use the dirichlet prior to estimate the p(concept|term) and p(document|concept)	provide a list of concepts for labeling task from the CIDE ontology	-	the probabilities between the concepts from ontology and term or documents could be measured and analyzed.	document labeling; semantic labeling for terms
LDA_SongUser [174]	to find the closeness between a cluster of songs with other clusters of songs, and the closeness between a cluster of songs with a cluster of frequent users	get the topic-song probabilities p(topic|song) and the user-topic probabilities p(user|topic).	-	k-means	the song clusters and user clusters have their topic significance; the song clusters and user clusters can be associated with each other in the different clusters maps.	music knowledge map construction; songs clustering
LDA_word embedding [175]	to discriminate the polysemy and classify similar terms with concrete meanings	to learn words representations separately by the skip-gram algorithms and by the features of the topic of LDA, and join them together	-	-	the terms are classified to the closest concept, in which the polysemy terms could also be a concept to label the class and the terms in that class could help to distinguish the meaning of the polysemy terms.	term classification

Boyd-Graber, Blei et Zhu 2007 [170] proposed the 'LDA_WN' to distinguish the polysemy of terms in the WordNet taxonomy (WORDNET-WALK). For a term with different meanings (known as polysemy), it has different paths in the taxonomic structure of wordNet. Respecting the context of a polyseme, the 'LDA_WN' model will locate the taxonomic path, so that the sense of polyseme could be precised by the taxonomic terms of WordNet. To achieve this goal, they extract the concepts from the taxonomy, calculate the closeness probabilities between concepts, and embed these probabilities to LDA model. The new term representations of LDA could be used to measure the semantic similarity between the query terms and concepts in taxonomy, so as to locate the query terms with the closest concepts in taxonomy.

Except to acquire knowledge from WordNet, Probase [176] is also a good resource for capturing the relationships between terms. It contains a number of closely related terms and every relationship has a typicality score indicating the importance of one term to another term. *Kim, Wang et Oh 2013* [171] aimed to map terms to the concepts of Probase, i.e. 'apple' maps to 'fruit' or 'firm'. In the proposal 'LDA_Probase', they managed to represent the topic information by terms from text $p(topic \mid term)$ and link the topic information with the concepts from Probase $p(concept \mid topic)$. In this way, in the given context, the relatedness between a term and a concept from Probase could be examined, and terms could be mapped and semantically labeled by the concepts. Following the same idea, *Cheng, Wang, Wen et al. 2015* [172] proposed the 'LDA_Probase2' to distinguish the polysemy of terms by combining the explicit concept of Probase and the implicit concepts of topics regarding to context. They applied the k-medoids clustering on the raw terms of Probase to obtain the explicit conceptual clusters. The centroids of clusters are worded as the explicit concept embedding. Additionally, they explored the implicit concepts embedding by performing LDA over the corpus. Finally, they join these two embeddings together to disambiguate the meanings of terms respecting their context. They found that the explicit concepts have higher accuracy but low recall, and the implicit concepts provide a complement, which could include the words (such as verbs and adjectives) that do not exist in Probase ontology. And the joint effects outperform the single output.

The previous works focus on the semantic labeling for terms, in a different way, the 'LDA_CTM' model [173] is interested in the semantic labeling for Web pages. *Chemudugunta, Holloway, Smyth et al. 2008* [173] proposed the concept-topic model (CTM) to automatically tag Web pages with concepts from an ontology without any need for the

labeled documents. They modified the basic LDA and estimated the $p(concept \mid term)$ and $p(document \mid concept)$, where the original topics in the estimation are instantiated as the concepts from the Cambridge International Dictionary of English (CIDE) ontology. Then the probabilities between the concepts of ontology and term or documents could be measured and analyzed.

In the second direction, we would like to explain two typical works in detail, which are notated as 'LDA_SongUser' [174] and 'LDA_word embedding' [171] in Table 4.3.

Apart from applying LDA over the plain text, it could also be used in the structural text, i.e. the string names of songs [174] or the software log files [154]. Here we will detail the structural text with songs. *Zhou, Fan et Zhang 2019* [174] proposed 'LDA_SongUser' to use LDA to find the closeness between a cluster of songs with other clusters of songs and the closeness between a cluster of songs with a cluster of frequent users, in order to build up the knowledge maps of songs and users. To accomplish this goal, they started by building up the constituted corpus, which includes a number of separate text documents for each user. In each user's document, it includes a list of string names of songs that this user listen frequently. Based on this corpus, they acquired the topic-song probabilities $p(topic \mid song)$ and the user(document)-topic probabilities $p(user \mid topic)$ from LDA. Then they applied the k-means clustering on both probabilities representations to cluster the songs and to cluster the users. Finally, they matched the clusters in two clustering maps according to the similarity between clustering centers of songs and users. They observed that the song clusters and user clusters have their topic significance ; the song clusters and user clusters can be associated with each other in the different clusters maps.

In recent works, more and more researchers [175], [177]-[179] focused on combining together the word embedding techniques and LDA techniques for the concrete terms representations for specific tasks, i.e. polysemy discrimination, conceptualization, and term classification. Here we introduce one work from *Liu, Liu, Chua et al. 2015* [175], denoted as 'LDA_word embedding'. They aimed to discriminate the terms of polysemy and classify similar terms with concrete meanings. In the beginning, they learned word representations separately by the skip-gram algorithms [90] and by the features of the topic of LDA, joined them together in the trade-off manner. Based on the classification rules, the terms could be classified to a close concept. For more details, the polysemy terms could also be a concept as the label of a class, while the terms in a class reversely assist to distinguish the meaning of this polysemy term.

4.6 Evaluation Metrics for Topic Models

Except for the statistical evaluation metrics, topic models could be evaluated thanks to their intrinsic features as a language model. Without any post-processing interpretation, a language model could be examined for its predictive power in terms of its ability to predict words in unseen documents. There are two main metrics to evaluate the production of a language model, i.e. language model perplexity and topic coherence score. The perplexity can be interpreted as being proportional to the distance between the word distribution learned by the model and the distribution of words in an unseen document [173] ; and the topic coherence is used to judge how good a topic model on qualitative understanding of the semantic nature of the learned topics. In addition, expect for the evaluation on models, we also mentioned the evaluation metrics for topical terms, e.g. saliency and relevance.

4.6.1 Model perplexity

Let's think about this question, what makes a good language model ? Intuitively, we expect a language model to understand the rules of language but decrease the probabilities of chaotic language expression. In other words, a good language model is to pursue the high probability and lower perplexity while generating the language expressions. We could also interpret the value of perplexity in a simple way. For instance, if we have a perplexity of 100, it means that a language model is as confused as if it had to pick 1 word between 100 words. Thus, lower scores are better.

It is straightforward that, the longer a sentence the more uncertainty is introduced. To be independent of the size of text, model perplexity is calculated in a normalized manner by the total number of word probabilities. As shown in Equation 4.12, the perplexity could be interpreted as the inverse joint probability of a text, normalized by the number of words in that text, in which the W contains the sequence of words of all sentences, denoted by V words as $w_1, w_2 ... w_V$.

$$perplexity(W) = \frac{1}{P(w_1, w_2, ..., w_V)^{\frac{1}{V}}} \tag{4.12}$$

The joint probability assigned by a language model is simply expressed in Equation 4.13, once we take a unigram model as example. Under the assumption of unigram model, words are assigned with probability independently, then the joint probability

could be transformed into the products of the individual probability of words.

$$P(w_1, w_2, ..., w_V) = P(w_1)P(w_2)...P(w_V) = \prod_{i=1}^{V} P(w_i) \tag{4.13}$$

If we apply the perplexity in the text document for practice, the perplexity of the test documents is calculated by the individual probability of words given the train documents [173] :

$$perplexity(w_{test} \mid D_{train}) = exp(-\frac{\sum_{j=1}^{D_{test}} \sum_{i=0}^{N_{d_j}} \log p(w_i \mid D_{train})}{\sum_{j=1}^{D_{test}} N_{d_j}}) \tag{4.14}$$

, where D_{train} is the documents in training set, D_{test} is the documents in test set, w_{test} is the words in test set, w_i are words in document d_j, and N_{d_j} is the number of words in document d_j.

In the experiments of a topic model, the documents of the training set are used to train the model and the remaining test documents are for computing the perplexity. For each test document d_j, a randomly selected subset of the words in the document are assumed to be observed and used to estimate the document-specific parameters $p(z \mid d_j)$ via Gibbs sampling. Perplexity is then computed on the remaining words in the document.

4.6.2 Topic coherence

We have talked that the topic model can be evaluated intrinsically in terms of model perplexity, but there has been less effort on qualitative understanding of the semantic nature of the learned topics [180]. Topic models learn topics by representing as sets of important words, which are automatically extracted from massive documents. In this manner, we treat words as fact, compare word pairs and rate their topics as the coherence score for the qualitative measure.

To calculate topic coherence for topic models, there are four different stages to be concerned in the framework of coherence measure [181] : segmentation, probability estimation, confirmation measure, and aggregation. The coherence of a set of words measures the hanging and fitting together of single words or subsets of the words. In the beginning, we could segment a word set into pieces of pairs of word subsets. Secondly, the individual word probabilities are computed based on a given reference

corpus. Then, we choose a certain confirmation measure to score the agreement of a given pair. Last, those values are aggregated to a single coherence value for evaluation. In general, the higher the coherence score the better a topic model is.

As we have talked, the calculation of topic coherence is relevant to a reference corpus. Originally, it was the human task to evaluate the semantic relevance between words of a topic [182].

Now a variety of the automated coherence methodologies are proposed. On the one side, based on the statistic information, *Mimno, Wallach, Talley et al. 2011* [183] believed that standard topic models do not fully utilize available co-occurrence information and a held-out reference corpus is not required. Therefore they defined the coherence metric, relying only upon word co-occurrence statistics gathered from the corpus. The results proved that the low-quality topics can be detected by their metrics but not by the existing word-intrusion tests. *Newman, Lau, Grieser et al. 2010* [180] applied a range of topic scoring models for topic coherence evaluation, drawing on WordNet, Wikipedia and the Google search engine, and existing research on lexical similarity/relatedness. In comparison with the human evaluation methodologies, they found a simple co-occurrence measure based on point-wise mutual information over Wikipedia data is able to achieve results for the task at or nearing the level of inter-annotator correlation.

On the other side, by utilising word embedding algorithms, such as word2vec [89] and fastText [184]. *Belford et Greene 2019* [185] evaluated the different impact on coherence scores between these two popular word embedding algorithms and their variants, using two distinct external reference corpora. It is clear that the choice of these three factors has a large impact on coherence values, which might affect the interpretations made from the topics ultimately.

4.6.3 Metrics for Topical Terms

Apart from the evaluation metrics of the entire topic models, we summarize some metrics based on the closeness of terms to their affiliated topics. Here we would like to introduce the *saliency* and *relevance*, who are capable to examine the significant terms for a certain topic.

$$relevance(w, t \mid \lambda) = \lambda \log p(w \mid t) + (1 - \lambda) \log \frac{p(w \mid t)}{p(w)} \tag{4.15}$$

The **relevance** is defined by *Sievert et Shirley 2014* [186] to rank important terms within topics to aid for topic interpretation. As shown in Equation 4.15, the *relevance* depends on the $p(w \mid t)$, the $p(w)$ and a weight parameter λ, where $p(w)$ presents the raw counts of words and λ ($0 \leq \lambda \leq 1$) determines the weight given to the probability of term w under topic t relative to its lift. Setting $\lambda = 1$ results in the familiar ranking with the topic-word probability $p(w \mid t)$, and $\lambda = 0$ ranks terms solely by their lift. According to the analysis of *Sievert et Shirley 2014* [186], it is said that setting $\lambda = 0.6$ could be the optimal value to improve topic interpretation.

$$saliency(w_i) = p(w_i) \times \sum_{t}^{T} p(t \mid w_i) log \frac{p(t \mid w_i)}{p(t)} \tag{4.16}$$

$$p(t \mid w_i) = \frac{p(w_i \mid t) \times \ p(t)}{p(w_i)} \tag{4.17}$$

The **saliency** is proposed by *Chuang, Manning et Heer 2012* [187] to filter out terms, so as to assist for the rapid disambiguation of topics. In Equation 4.16, it is composed by $p(t \mid w)$, $p(w)$ and $p(t)$, where the $p(t \mid w)$ is the probability about the observed words generated by a latent topic t. It could be induced by the topic-word probability $p(w \mid t)$, marginal distribution of words $p(w)$ and marginal distribution of topics $p(t)$, as shown in Equation 4.17. In the right side of Equation 4.16, it describes how informative the specific term w_i is for determining the generating topic versus a randomly-selected term to determine such a topic. There are more generic terms than distinctive terms for the topics, *saliency* could differentiate among the significant topics and potential junk topics by measuring the salient terms inside those topics.

However, in brief, even though these metrics of language model and of topical terms provide us a useful way of quickly comparing models, it does not take into account the specific tasks, i.e. the evaluation of term classification or clustering. These initial evaluations of language model are not as "good" as the statistical evaluation metrics, who are dedicated to achieve a specific final task. Therefore, to have an integrated evaluation for the performance of topic models, we could employ both kinds of these evaluation metrics : language model evaluation metrics and the statistical evaluation metrics.

CHAPITRE 5

LDA DRIVEN BY CORE CONCEPTS FOR TERM CLUSTERING

From the experiments and results of Chapter 3, we learned that LDA could assist to achieve better performance in term clustering, not only as the topic model to learn a probabilistic embedding of terms and documents into a topic feature representation for the classic clustering strategy but also as the foundation of the LDA-based clustering strategy. Among them, we noticed that the LDA-based clustering strategy has an overwhelming clustering performance. However, it is difficult to correlate the resulting clusters to the meanings of core concepts in that corpus.

Here we propose to enhance the strategy of using LDA as a clustering strategy, to cluster terms over core concepts as a support for learning a modular ontology. It allows us to build a modular ontology where the clusters of each core concept constitute a module of the ontology to build. In Section 5.1, we introduce the approaches using core concepts and their hyponyms as prior knowledge to guide LDA to cluster over core concepts. Then, we discuss the techniques to discover taxonomic relations by noun modifier relationships and knowledge bases. In Section 5.2, we analyze the key factors that could affect the performance of our proposal, e.g. the syntactic roles of NPs, the inclusion of verbs occurring with NPs, the inclusion of NPs which only exists in GS, and the number of LDA training times. In Section 5.3, we experiment with our approach by studying the effect of these factors and other LDA parameters on its performance. In section 5.4, we analyze the results of these experiments and according to that, we highlight the best manner to embed prior knowledge to our strategy of using LDA for term clustering towards modular ontology learning. Then we compare it with other LDA-based approaches using prior knowledge, i.e. z-label LDA and seeded LDA. Then we finish by summarizing the various employments of LDA for modular ontology construction in Section 5.5. It should be noted that the partial work in this chapter about twice trained LDA was first published in KES conference [188].

5.1 Our proposal : LDA guided by core concepts and their hyponyms

In this section, we propose to improve the LDA-based clustering strategy for modular ontology building in two sides : 1) In the LDA training procedure, we apply the **prior knowledge embedding** techniques to guide LDA model within the clustering strategy, in order to obtain the term clusters close to core concepts ; 2) In the ontology construction procedure, we employ the **taxonomy relation discovery** techniques to construct the taxonomic hierarchy of each module, i.e. noun modifier relationships methods and knowledge bases methods. Eventually, benefiting from the results of the optimizations on two sides, we present the workflow to construct the modular ontology with the term clusters of topic semantics.

Starting from the re-constituted corpus, the prior knowledge embedding techniques are composed of two main methods relating to the core concepts : 1) applying the **core concept replacement** method i.e. if a NP is a hyponym of a core concept (in the following we call it a CC-subconcept), this NP will be replaced by this core concept ; 2) implementing the **subdomain knowledge supplementation** method i.e. If a document has the prior knowledge relating to several CC-subconcepts, this document will be extended with these CC-subconcepts. The CC-subconcepts could belong to one CC or several CCs, respecting the presence of prior knowledge.

We believe that the two prior knowledge embedding techniques bring many benefits for the LDA-based clustering strategy in terms of guiding the training of LDA. The first method reduces the sparseness of topics, for which situation the meaning of topics is close to the sparse CC-subconcepts but not the integrated core concepts. In response, this method enforces the topics with a stronger connection to a core concept. The second method assists LDA to capture more CC-subconcept-related context in a document. Also, they bring benefits in terms of forming clusters close to core concepts. The first method facilitates clustering performance from gathering CC-subconcept-related terms to gathering the core concept-related terms. And the second method helps gather the related terms with the assistance of the supplemented context. Overall, the proposed techniques could guide LDA to gather terms related to the CC-subconcepts and the core concepts.

5.1.1 Prior knowledge embedding

The corpus pre-processing procedure makes use of its linguistic features to simplify the corpus. In this procedure, we would like to enrich this kind of simplified corpus with the related prior knowledge of topics. For the core concept replacement method, we would discuss what kind of CC-subconcepts are interesting to be replaced, where to find those subconcepts and what the replacement benefits are. For the subdomain knowledge supplementation method, we would distinguish the difference from the previous method and decide how many CC-subconcepts would be added in the reconstituted corpus.

Core concept replacement

The precondition of core concept replacement is that the partial taxonomy relations of core concepts are known, which will be detailed later in Section 5.1.2. Once a core concept possesses the containment relation to their subconcepts, it would present much more general meaning than their descendants. Generally, the slight difference between a core concept and its subconcept brings little impact to the topics or the subdomains that they belong to. Therefore, the replacement from subconcepts to core concepts in a new corpus brings little difference. For this reason, a new corpus could be constructed without affecting the validity of the topic representation of terms [189]. Additionally, core concepts play an essential role in topic modeling to match topics to subdomains. Intuitively, if we replace CC-subconcepts with their core concept, the meaning of topics obtained from LDA tends to be close to the core concept.

For instance, in a computer science domain, we can consider two subdomains : the 'machine learning subdomain' and the 'data structure subdomain.' In this way, the two associated core concepts are 'machine learning' and 'data structure'. After the terms are replaced by the core concepts, for example, *'supervised classification'* is replaced by *'machine learning'*, the topics would have a higher tendency to correspond to the subdomains of the corpus. Notably, not every term is acceptable to be replaced by a core concept ; it will induce a semantic drift if the replaced terms are far from the core concepts that replace them. Therefore, the recognition of core concept-subconcept pairs turns to be the key parts. In Section 3.3.2, the terms appearing as the "keywords" of documents are feasible to express the critical meanings of the subdomains ; thus some subdomain-related terms appeared in the "keywords" prone to be selected as

TABLE 5.1 – The diverse approaches to employ prior knowledge in different cases.

Approaches	Functions	Examples
Approach1	without any prior knowledge	new image analysis algorithm', 'layer neural network', 'support vector machine'
Approach2	core concept replacement upon Approach1	new image analysis algorithm', 'layer neural network', **'Machine learning'**
Approach3	keywords supplementation upon Approach1	new image analysis algorithm', 'layer neural network', 'support vector machine', **'computer vision', 'yield prediction'**
Approach4	core concept replacement and supplementation upon Approach1	new image analysis algorithm', 'layer neural network', **'Machine learning', 'Computer graphics', 'Computer graphics'**

subconcepts. Accordingly, the pair can be denoted as $<$*CoreConcept, Keyword*$>$. In Table 5.1, the *'support vector machine'* is recognized as the 'Keyword' in 'Approach2' and is replaced by its 'CoreConcept' *Machine learning*. In the following content, we will utilize *keywords* as a unified name for the subconcepts.

Concerning the bag-of-word assumption, the core concept replacement technique brings many benefits in supporting LDA training :

- The size of the vocabulary is dramatically reduced, which leads to the reduction of computational complexity.
- The frequency of hypernym terms has a significant increase, such that their statistic importance is highlighted during training.
- The neighbor terms of original hyponyms (words being replaced) could be captured easily because they co-occur with the more frequent hypernyms after replacement, to implicitly reinforce the connection between neighbor terms and the substitute hypernyms.

Subdomain knowledge supplementation

The previous method tends to adjust corpus to help topics get close to subdomains, comparatively, the subdomain knowledge supplementation method preserves original terms but adds supportive information to increase the prominence of supportive information for topic modeling.

From the bag-of-world assumption, LDA uses the statistics of term and document frequency to generate the probabilistic topic model ; therefore, the order of terms in a document is not important for the LDA algorithm. As a new method, the supportive

information could be appended at the tail of each document, which extends the content with a higher occurrence of these appended words. In this context, the supportive information should present with the most specific and expressive terms to accentuate the dependence between documents and their subdomains(i.e., keywords in each document). As the keywords discussed in Section 3.3.2, we would append from 0 up to 5 keywords for each document. Overall, the supportive information for each document can be the keywords list or the list of the corresponding core concepts of these keywords, corresponding to 'Approach3' and 'Approach4' of Table 5.1 respectively.

5.1.2 Taxonomy relation discovery

Here we are interested in building a partial taxonomy deriving sub-concepts from the core concepts. There are two resources to acquire taxonomic relation between terms : 1) from the text's linguistic features ; 2) from the published taxonomy of common knowledge bases. We apply the noun modifier relationships techniques in the former approach to identify the taxonomic relations between noun phrases, especially between NPs and the core concepts.

Taxonomic discovery from text by noun modifier relationships

A head noun along with a noun pre-modifier is often called a noun compound, which is one kind of noun phrase. *Levi 1978* [190] argued that the word formation make noun-noun compounds a heterogeneous class.

For the purpose of taxonomic relation discovery, we would like to focus on the semantics of a noun compound that are *transparent* and *endocentric* according to *Barker et Szpakowicz 1998* [7]. The meaning of a *transparent* compound is close to its elements. For example, 'query language' is transparent (the language specified for the usage of a query) ; in the opposite, 'Guinea pig' has no obvious relationship to guinea or to pig. The meaning of an *endocentric* compound is considered as a hyponym of its head. For example, 'relational database' is a kind of 'database' ; in the opposite, 'bird brain' does not refer to a kind of 'brain', but rather to a kind of person (whose brain resembles that of a bird).

Furthermore, the multiple nouns compounds is also a kind of noun phrase, who includes more than two words. The taxonomic relation is located between the head noun compound and the pre-modifier noun compound. For example, 'formal query language'

TABLE 5.2 – The discovered hyponyms of head terms

head terms	query language	relational database	knowledge base
NPs with prefix (hyponyms)	formal query language	probabilistic relational database	behavior knowledge base
	database query language	non relational database	current knowledge base
	keyword query language	relational database system	heterogeneous lexical knowledge base
	generic temporal query language	multi relational database	ontological knowledge base

is a kind of 'query language'. *Liberman et Sproat 1992* [191] has investigated to bracket those head noun compounds and pre-modifier noun compounds to specify the relation between them. In fact, a head noun compound contains a more general concept than its combined noun phrases. We supposed that the shorter head noun compound (namely hypernym) holds a taxonomic relation with its longer combinations(namely hyponym).

However, to guarantee the reliability of the discovered relations, it is vital to judging whether a head noun itself is meaningful as domain knowledge. If a head noun is significant for a domain, it is convinced that its noun phrase is still meaningful because it becomes a more concrete concept. Therefore, we consider the core concepts themselves and their corresponding keywords as the head noun compound.

To find the hyponyms of the selected head terms, it is also to find the noun phrases in which a head term is combined with prefixes. We can benefit from a pattern defined by the regular expression :

$$.*HeadTerms\$$$

This pattern is used in string searching algorithms to find the target string as its corresponding hyponym.

For instance, in Table 5.2, here are several examples of the resulted hyponyms derived from the selected head terms. It is worth noting that the head terms themselves could be noun phrases, which is even more preferable than the single term.

Consequently, this pattern could lead to rather high precision with the cost of the low recall. The probable damage over precision is given by adding the negative prefix to a head term. i.e. *structured storage* and *non-structured storage*.

TABLE 5.3 – An example for hyponym acquisition in DBpedia.

Step	Logical Implication	SPARQL Query
i	$\exists dbr{:}Machine\_learning$	-
ii	$\exists dbr{:}Machine\_learning.$ $dct{:}subject.\ \{dbc{:}res\}$	**SELECT** ?x WHERE {dbr:Machine_learning dct:subject ?x. FILTER regex(?x, "http://dbpedia.org/resource/Category:")}
iii	$\exists\{dbc{:}res\}.skos{:}broader.$ $dbc{:}Machine\_learning$	**SELECT** ?y WHERE { ?y skos:broader dbc:Machine_learning. FILTER regex(?y, "http://dbpedia.org/resource/Category:")}
iv	$\exists\{dbc{:}res1\}.skos{:}broader.$ $dbc{:}res$ $\exists\{dbc{:}res2\}.skos{:}broader.$ $dbc{:}res1$ ...	**SELECT** ?z WHERE { ?z skos:broader{1,5} dbc:Machine_learning. FILTER regex(?z, "http://dbpedia.org/resource/Category:")}

Taxonomic discovery by knowledge bases

The knowledge bases tend to blanket extensive information to be encyclopedic and serviceable for users. It is beneficial to connect the local ontology to the global knowledge bases to obtain a befitting and expert domain ontology. Especially, we focus on extracting the taxonomic relations between core concepts and their hyponyms that occurred in the corpus.

However, it is not simple to canonicalize the noun phrases to the available entities in the knowledge bases. The canonicalization process is, given the synonyms of NPs, to select a representative NP that will replace the other NPs in the canonicalized KB [118]. To simplify the canonicalization process, we apply the *API :Opensearch* [192] from MediaWiki project [1] to match the NPs to the existing items in the related knowledge bases. We draw attention to the two widely used knowledge bases, i.e., DBpedia [193] and Wikidata [194], and anticipate retrieving the hyponyms of the proposed core concepts. For instance, we follow four steps by applying SPARQL queries in DBpedia to obtain the descendants of core concepts :

i). match a NP to one existing entity in DBpedia

ii). match the existing entity(i.e. 'Machine_learning') to the category items (i.e. 'Cate-

1. https://www.mediawiki.org/wiki/API:Opensearch

gory :Machine_learning') in DBpedia

iii). find the subcategories of the obtained category item

iv). find the subcategories in the depth of 5 layers of the obtained category item

For each step, we provide the corresponding logical implication and the SPARQL query with "Machine Learning" as an example in Table 5.3. The first step implies that the noun phrase "Machine Learning" is successfully canonicalized by an identifier of DBpedia by using the *API :Opensearch* [192], i.e., $dbr{:}Machine\_learning$. In the column of logical implication, the query results are denoted as $\{dbc{:}res\}$, and the relations stand out as $dct{:}subject$ and $skos{:}broader$. We carefully select the former relation to explore the category items and apply the latter relation regarding the obtained category. The final results are listed in the Appendix as Figure 7.18.

By contrast, the taxonomy extraction process from Wikidata is more convenient with the assistance of a tool *wikidata-taxonomy*[2]. This tool makes use of the property "subclass of" or "subproperty of" to obtain the descendants of a canonicalized noun phrase, and one example of the 'Machine Learning' taxonomy is listed in the appendix of Figure 7.19.

After the hyponyms extraction from the two knowledge bases, it is important to check whether the extracted NPs have also occurred in our source corpus. To maximize the hyponym outcomes, for each core concept, we merge its hyponyms from the two different sources. As shown in Table 5.4, the first group columns present the statistics of the extracted hyponyms from two KBs. Then we will filter out the duplicate hyponyms of each core concept and delete the chaotic hyponyms that affiliate to multiple core concepts, denotes as the two deletion Phrases 'Del1 and Del2'. In the end, those extracted hyponyms will be cleaned in the same manner as that of other NPs in the corpus. From the second group columns, we can observe that the corpus only contains 215 NPs over the entire extraction with 1129 NPs, and the occurrence of these 215 NPs takes place 1308 times in the corpus.

Interestingly, we also summarize the effects of taxonomic phrase discovery by applying the techniques of noun modifier relationships. In the last group of Table 5.4, all of the hyponyms of core concepts are taken into account in column 'Without Del'. Then the hyponyms that only appear once in the corpus are deleted to avoid storing the rare hyponyms. It shows that we could acquire 164 unique hyponyms of core concepts, and

2. https://github.com/nichtich/wikidata-taxonomy

TABLE 5.4 – The extracted hyponyms and its appearance in corpus

Core Concepts	#hyponyms in the knowledge bases				#KB hyponyms in Corpus		#hyponyms from the noun modifier relationships		
	Two KBs	Del1	Del2	Clean	#unique	#Occu	#without Del	#unique	#Occu
Algorithm_design	0	0	0	0	0	0	112	14	38
Bioinformatics	12	12	12	13	8	24	86	1	2
Computer_graphics	36	36	36	34	32	141	52	6	14
Computer_programming	115	114	113	120	20	43	57	5	18
Cryptography	21	21	21	20	21	21	75	15	98
Data_structure	211	200	197	202	11	113	290	54	186
Distributed_computing	50	43	43	42	8	140	0	0	0
Machine_learning	75	75	75	75	9	236	92	6	18
Operating_system	635	565	562	518	25	112	172	38	188
Software_engineering	72	72	71	69	44	293	100	15	40
Network_security	37	37	37	36	37	185	73	10	42
total	**1,264**	**1,175**	**1,167**	**1,129**	**215**	**1,308**	**1,109**	**164**	**644**

Note : 'Del1' denotes the operation to filter out the duplicate hyponyms of each core concept ; 'Del2' is to delete the chaotic hyponyms that affiliate to multiple core concepts ; '#unique' represents the number of unique hyponyms ; '#Occu' denotes the occurrence of the unique NPs in the corpus ; '#withoutDel' means all of the hyponyms of core concepts are taken into account.

they appear 644 times in the corpus. All in all, we observe that it has no overlapping between the hyponyms from KB and the hyponyms from the noun modifier relationships. Therefore, the total number of hyponyms we found is the summation of the two kinds of hyponyms.

From the comparison of the hyponym discovery performance between the two approaches, it is clear that the occurrence of extracted hyponyms by the noun modifier relationships approach is rather lower than that by the KB approach. It implies that the KB approach could extract the hyponyms that are significant in corpus occurrences. However, the noun modifier relationships approach is much easier and faster to execute and acquire taxonomic results.

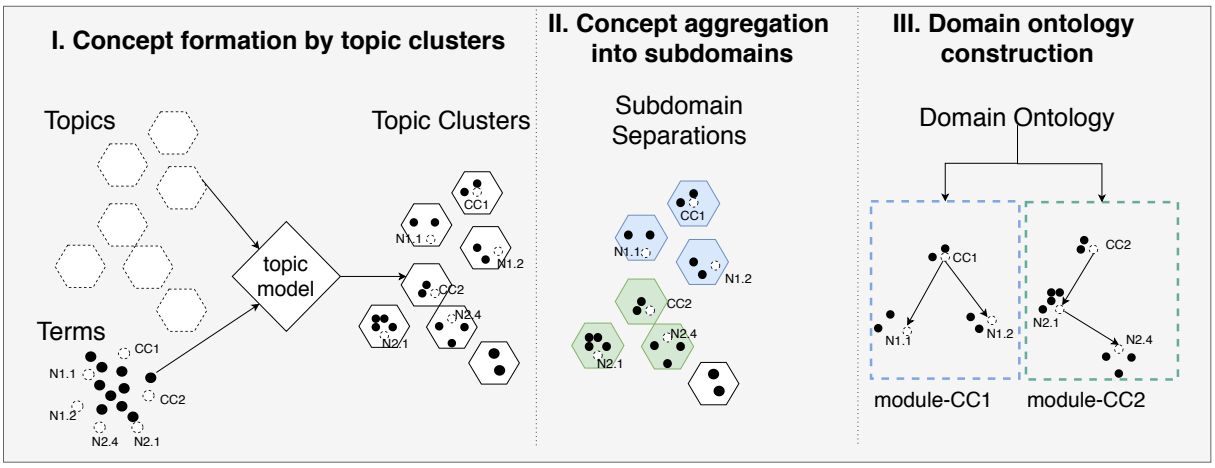


FIGURE 5.1 – The modular ontology construction with topic clusters.
Notes : The black dots in bottom-left are the NPs extracted from the corpus, where the 'CC#' denotes the core concepts and 'N#.#' denotes their corresponding hyponyms.

5.1.3 Towards modular ontology

After the formations of clusters and their aggregation by subdomain (i.e. by core concept), we could benefit from the recognition of the taxonomic relations among NPs, to construct a modular ontology and a taxonomy inside each module. The workflow is presented in Figure 5.1. In step I, all terms could be assigned to topic clusters according to the LDA training results. In step II, these topic clusters are tagged into the different separations (i.e. the blue clusters and green clusters), which guarantees that the cluster including one core concept and the clusters including the hyponyms of this core concept are aggregated together. It is worth mentioning that the white cluster here is not assigned to any separations because it does not contain any core concepts or their hyponyms, which would be eliminated in the ontology construction step.

In the final step, each module corresponds to one sub-domain separation that includes the related topic clusters. And the discovered taxonomic relations will be considered the backbone of a hierarchy. Except for the backbone terms, the terms, who do not hold taxonomic relations but in the same topic cluster with a core concept, possess the topic-related relations with this core concept. Because we assume that all the terms within a topic cluster hold the topic-related relations to each other.

These relations set up the strong inner connections for a module. For the modules of an ontology, they are loosely coupled and alternative to be replaced. This flexible construction facilitates knowledge reuse and provides users the knowledge with the

self-adapted scope.

5.2 The Influencing Factors of our Proposal

5.2.1 Composition of the reconstituted Corpus

The idea of the reconstituted corpus has been introduced in Section 3.1.1, here we would like to detail this idea using more the notion of probability.

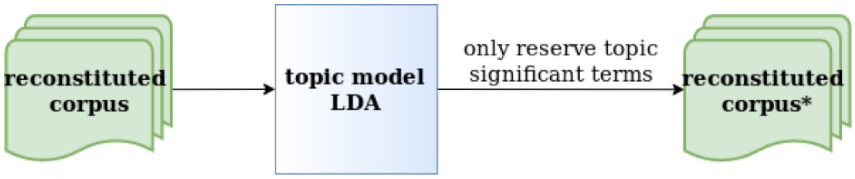
The general process of the corpus reconstitution is to extract the 'focusing terms' from a document and store them into a string list as a reconstituted document. In fact, the types of 'focusing terms' work as one of the main influencing factors in our approach. We consider four kinds of 'focusing terms' : 1) NPs with syntactic roles as subject and object ; 2) NPs and their co-occurred verbs ; 3) NPs that only exist in Gold Standard ; 4) NPs that have high topical significance.

NPs as subject and object

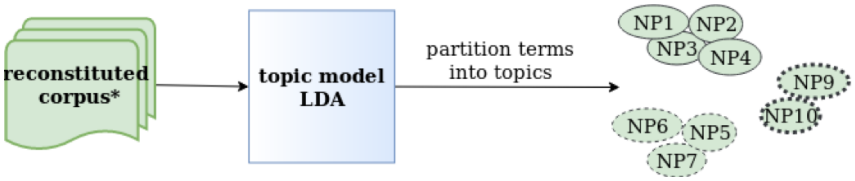
Regarding the utility of syntactic roles, a sentence's skeleton comprises the subject, the object, and their related verb. In Section 3.1.1, we believe that terms with important syntactic roles are assumed to cover the most descriptive information in a sentence. Despite the other term constituents of a sentence, we would like to examine the difference between the NPs as subject and object and the NPs with any syntactic roles.

NPs with verbs

The NPs, acting as subject or object, are worth to be highlighted in concept extraction. On the one hand, their contextual components, i.e., verbs or VPCs, could present the concrete connection between NPs. On the other hand, verbs or VPCs usually serve as functions in sentences [195]. We would like to designate the reconstituted corpus to only include NPs with the syntactic role as subject and as an object and/or include their co-occurred verbs as well.



(a) A first trained LDA



(b) A second trained LDA

FIGURE 5.2 – The procedures to extract NPs with high topical significance in the modle of twice trained LDA [188].

NPs from GS

Imagine that there is the only domain-related term (i.e. terms from Gold Standard) in the corpus, it is interesting to discover how the topic model will perform on this 'pure corpus'. The comparison between this corpus and other reconstituted corpus will help to judge whether the inclusion of domain unrelated term contributes to the topic model.

NPs with high topical significance

Except for the terms from Gold Standard, it is noteworthy that many other terms are not interesting to be engaged in the partitions of domains. For instance, if some terms are not significantly related to a certain topic, these terms' involvement will bring some fuzzy meanings into topic clusters and even give rise to semantic drift of topics. To address the problem, we propose to employ twice trained topic modeling of LDA as an approach to sorely concentrate on the topic-significant terms, which is detailed in my published paper [188]. In the drawing of Figure 5.2, from the first training of LDA, we can obtain the term probabilities of topics and term significance of topics from the corpus. This information can be used as indicators to identify topic-significant terms. Then the residual terms will be kept for the second training of LDA. It is anticipated that the second trained LDA can cluster terms into topics regarding the significance of terms.

Overall, the different operations correspond to the different cases of the manipulated

TABLE 5.5 – The different cases of corpus reconstitution.

	Nps as Subj or Obj	NPs with Verbs	NPs From GS	NPs with high topical significance
Case1	yes		yes	
Case2	yes	yes	yes	
Case3	yes			
Case4	yes	yes		
Case5				
Case6		yes		
Case3-T	yes			yes
Case4-T	yes	yes		yes
Case5-T				yes
Case6-T		yes		yes

corpus. To have a clear overview, in Table 5.5, the columns present these four kinds of "focusing terms". A row denotes a case of a specific reconstituted corpus associated with a combination of "focusing terms" kinds. Cases ending with '-T' correspond to the fourth factor with the model of twice trained LDA. We study the influence of the kind of "focusing terms" and find out the suitable cases to support LDA as a clustering strategy.

5.2.2 Corpus Filtering

As discussed above, the corpus is re-constructed in different manners respecting different phases, but the primary purpose stays the same : it is to facilitate the training of LDA for better clustering results. Still, the most common and rarest words' existence decreases the efficiency to learn LDA, where the common words mix up the separate topics, and the rare words increase the complexity of calculation but contribute little to the distinction of topics. To handle this problem, the idea of the TF-IDF (Term Frequency - Inverse Document Frequency) statistic model could help to filter out the most common terms, and their term frequency could simply separate the rare terms.

A high value in TF-IDF is reached by a high term frequency and a low document frequency of this term. The value hence is used for filtering. However, the integrated TF-IDF value also tends to filter out topic-related terms under a certain threshold. Based on

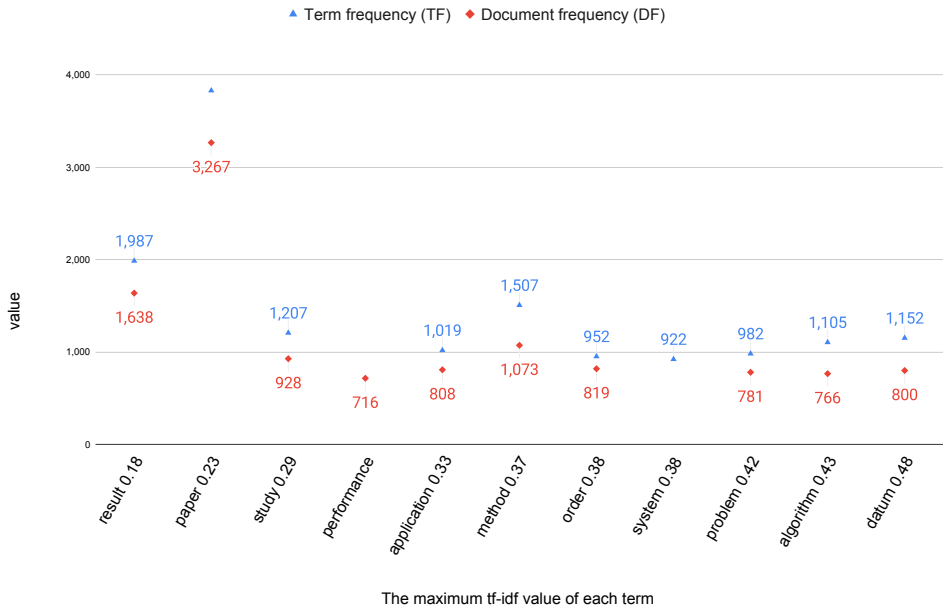


FIGURE 5.3 – The comparison of the top-10 terms of TF and DF regarding to their maximum TF-IDF values.

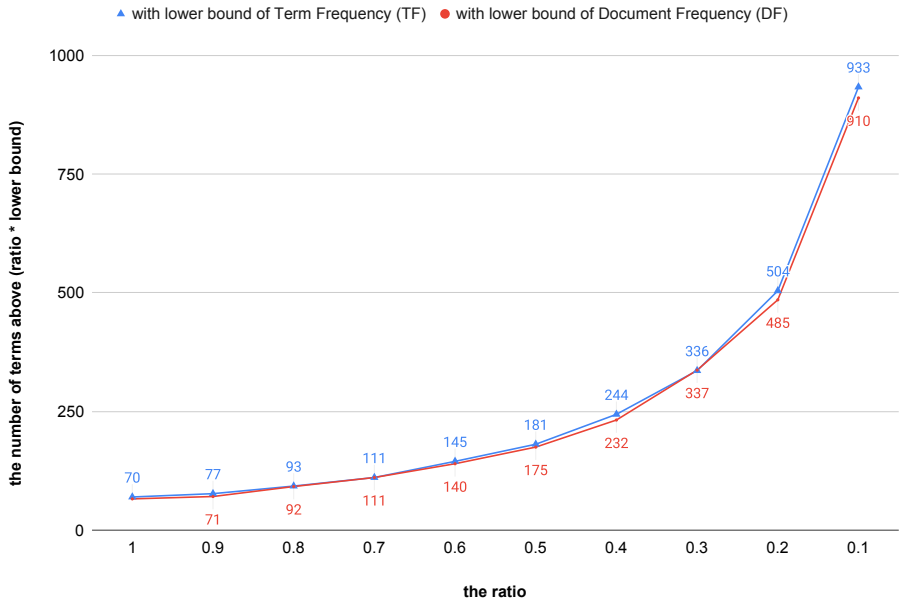


FIGURE 5.4 – The number of 'common terms' that exceeds the different ratios of the lower bound.

this problem, it is better to analyze the filtering impact of term frequency and document frequency individually.

In the beginning, it is interesting to check whether the two individual metrics tend to filter out the same sets of terms within the same range. In Figure 5.3, the top-10 frequent terms of TF and DF are listed in the order of their maximum TF-IDF values. It is evident that even though the two metrics could filter partially the same terms, still they have different preferences to filter out words. And we could conclude that the maximum TF-IDF value of a term could not provide the evident boundary to distinguish those 'common terms'. However, with the combination of TF metric and DF metric, it is sufficient to filter out the common terms by given a certain threshold.

The threshold could be a concrete value, a ratio of the set, or even the appearance of a 'target term'. For this corpus, our 'target term' could be the most interesting, related, and distinctive term ; it is acceptable to consider the most frequent *core concept* term as a 'target term'. For this 'target term', it is easy to acquire its term frequency and document frequency. These two values could be assumed as the 'basic lower bound' ; the number of terms whose frequency exceeds this 'basic lower bound' could be recorded then. In Figure 5.4, it presents an example with concrete numbers above the threshold (ratio lower bound) in the corpus *Computer Science*. In this example, along with the decreasing ratio in X-axis, the number of terms whose frequency exceeds the threshold (ratio lower bound) is increasing dramatically. We could observe a big gap between ratio 0.2 and ratio 0.1 ; To avoid over-filtering, the frequency with ratio 0.2 is specified as the 'final threshold'.

Once the 'final threshold' is selected, the two sets of terms could also be extracted by both term frequency and document frequency. The overlapping terms of two side extraction would be considered as the common terms, which are designed to be filtered out then.

Besides terms, it is also required to eliminate some fragmented documents, which only include a few words as the entire content. In the pre-processing procedure, since each document has been transformed into the designed content with NPs, the extracted content may be extremely limited. On this occasion, the fragmented document would be filtered out because of its tiny content. Then this step is followed by filtering out terms to have the newly constructed corpus. Table 5.6 presents an example of Computer Science corpus with concrete data for the whole filtering procedure. It is evident that step-(2) has four times less content than the original corpus in step-(1). In

TABLE 5.6 – The statistics of filtering of computer science corpus.

Different Phrases of Corpus	Type of Tokens	Number of Tokens occurrence	Number of Unique Tokens	Additional Info.
(1). Source File	Terms	1,180,813	76,459	# files : 6514
(2). All NPs File	NPs	289,564	120,293	# files : 6514
(3-Filtering Out Files) (< 10)	NPs	289,333	120,189	# files : 6486
(4-Filtering Out Common Terms)	NPs			#terms to be deleted : 398
(5-Filtering Out Rare Terms) (< 2)	NPs			#terms to be deleted : 95,310
(6-New NPs file)	NPs	114,807	24,585	

the filtering procedure, the fragmented files are firstly cut out if they contain NPs less than 10 (i.e. step-(3)). Then the common and rare NPs are filtered out in step-(4) and step-(5). Finally, step-(6) ends with the extra two times less based on step-(2).

5.3 Experiments

To improve the LDA model for the term clustering purpose, we focus to examine the three main aspects that significantly influence the LDA model's performance. Firstly, we study to find the optimal parameter setting for LDA models regarding a convinced metric, i.e. silhouette width. Then, along with the same parameter setting, the LDA model's random effects would be examined, in which the stable LDA models are favored. Finally, once we have settled down the steerable parameters of LDA models, we summarize the procedures of model selection to find the models that are capable of providing good performance on clustering terms into sub-domains.

5.3.1 Parameter setting

In order to accelerate the training process of the LDA model in the massive document collections, it is prominent to use the online variational Bayes algorithm for Latent Dirichlet Allocation(LDA) [196], owing to the quick convergence of variational Bayes objective function. Now the concrete model has been decided, it is important to learn

about the impact of the intrinsic parameters of the algorithm on the performance of the LDA model :

- **Topic number K**. The number of topics has the most straightforward influence on the probabilities between topics and words. The fewer topics are prone to mix up the comprehension of the sub-domains knowledge, while the more topics apt to over-interpret the concepts into the scattered term clusters. In contrast with these two extreme cases, we anticipate that a term cluster holds a detailed concept rather than a mixed concept. Therefore, the number of topics is worth to be measured in the increasing trend, e.g., from 10 topics to 200 topics.
- **Alpha** α. The parameter α presents the prior belief for each topics' probability. In most cases, it is set as the symmetric distribution, where all the topics have the same probability in the beginning.
- **Beta** β. The parameter β presents the prior belief for the topic terms probability. In parallel, it is set as the symmetric distribution as default usually.
- **Chunksize**. The learning parameter chunksize indicates the number of documents to be used in each training chunk. It is the customized parameter for online LDA [196]. The higher the value, the shorter the computation time, but the performance suffers.
- **Repetition**. The times to repeat the training procedures without changing any parameters above. Here all of the experiments are repeated 10 times and they offer the averaged results for evaluation.

After learning about the parameters' utilities, we expect to discover the optimal parameter setting for LDA models. For this reason, we could benefit from the evaluation metrics. The target of selecting the LDA model parameters is to obtain the optimal topic-term probabilities, where the terms show the distinctive likelihood for a certain topic. In the geometric space, the goal is reflected by the preferred phenomenon, in which the terms gather together for one topic and the distance between topic clusters is far from each other.

It conforms to the idea about the compactness and separateness of the mentioned metric, **silhouette width**.

The benefit is that, by using the silhouette width, the parameter selection procedure be done without using the Gold Standard. The silhouette width of the topic term probability will be considered as the criterion for selecting the parameters. We alter the number of topics and the number of chunksize as the two main LDA model's variables,

assuming others to be the default setting. Worth mentioning that, because of the application of forming terms' clusters, the topic clusters in geometric space are instantiated into terms' clusters. However, not all the topic clusters possess their topic-significant terms (see Section 3.2.2), which leads to some empty topic clusters. The empty cluster means that the topic feature is not able to be instantiated by terms. All of the variables and their resulting silhouette widths are presented in Table 5.7. The results are acquired based on the Computer Science corpus ('case5' in Table 5.5) by applying the filtering ratio of lower bound to 0.2.

On the one hand, we assume that the high silhouette width reflects the better aggregation of terms. On the other hand, we also need to take into account the number of topics. As shown in Table 5.7, the highest silhouette width appears in the fourth record from the end. However, in this record, the number of non-empty clusters is too low to provide sufficient and clear clusters. We would like to choose the parameter (50, 50,50, 0.949) (which is underlined in the table) because there are no empty clusters and the silhouette score is nearly high. Therefore, we select the corresponding parameters to train an LDA model.

However, the optimal parameters are not fixed for the different corpora. If a new corpus is introduced, we need to go through all of the candidate parameters to find the optimal option. Once the number of topics has been settled down, selecting parameters could be facilitated by only choosing the chunksize.

5.3.2 Random effect

Because of the existence of random effects in the topic model, we usually always get different results with the same parameter setting. After considering this, it is necessary to learn about the influence of random effects on the term clusters' final results. The final term clusters are instantiated by a group of terms, however, the silhouette width metric cannot be used to measure the semantic closeness of terms and the size of terms in clusters. At this point, we choose the *macro, micro, and pairwise metric* as direct evaluation measurements. In this manner, we could learn about the random effect respecting applying LDA as a clustering strategy.

For instance, we directly apply the selected parameter setting (the underlined record in Table 5.7) to the LDA model and estimate the cluster results with the assistance of the Gold Standard. The results are acquired based on the Computer Science corpus

TABLE 5.7 – The silhouette width on topic term clusters with different parameters of LDA in Computer Science corpus.

# topics	# chunksize	# non-empty clusters	silhouette width
10	10	10	0.658
10	50	10	0.685
10	100	10	0.729
10	500	10	0.500
10	2000	10	0.290
50	10	24	0.927
50	50	50	0.949
50	100	50	0.937
50	500	50	0.716
50	2000	50	0.558
100	10	28	0.966
100	50	81	0.925
100	100	97	0.917
100	500	100	0.821
100	2000	100	0.626
200	10	4	0.984
200	50	28	0.977
200	100	73	0.961
200	500	175	0.890
200	2000	200	0.673

TABLE 5.8 – The random effect of LDA model.

	LDA model			average	standard deviation
	RANDOM 0	**RANDOM 1**	**RANDOM 2**		
Avg size of cluster	7.6	7.0	7.5	7.4	0.2658
macro_prec	19.4%	14.3%	15.8%	16.5%	2.16%
macro_recall	0.0%	0.0%	0.0%	0.0%	0
macro_f1	0.0%	0.0%	0.0%	0.0%	0
micro_prec	57.7%	59.9%	54.0%	57.2%	2.40%
micro_recall	25.6%	25.2%	25.6%	25.4%	0.19%
micro_f1	35.4%	35.4%	34.8%	35.2%	0.32%
pair_prec	36.6%	39.1%	30.2%	35.3%	3.76%
pair_recall	11.1%	10.2%	10.9%	10.7%	0.40%
pair_f1	17.1%	16.2%	16.1%	16.4%	0.44%

Notes : The LDA models are trained with 50 topics, 50 chunksize and 50 non-empty clusters. It is applied on the Computer Science corpus by applying the filtering ratio of lower bound to 0.2.

('case5' in Table 5.5) by applying the filtering ratio of lower bound to 0.2.

As shown in Table 5.8, the evaluation results are generated from the same LDA model but with three different random states. From the comparison, we observe a little difference between the averaged clusters' size, which suggests that the resulting terms keep stable in quantity for different random states. It conforms to the small variations on the recall values ; that is to say, the amount of retrieved terms has a small change. Besides, we notice that the difference in the three precision metrics changes significantly than other metrics. This scenario reveals that some terms alter their affiliations of clusters in random states.

In brief, the LDA model's random effect does not have a strong impact on the resulting range of terms, but it brings a slight variance in the affiliation of the clustered terms.

5.3.3 Model selection

Up to now, we have discussed the method of optimal parameter setting for the LDA model. We also measured the random effects of LDA models when the parameter set-

ting is the same. However, we still have no idea which model satisfies the anticipation of term clusters. In this section, we would like to present the strategy of selecting LDA models based on the previous training methods. To be generalized, we will go through the whole procedures of filtering datasets, clustering terms, and evaluating the LDA models. In this framework, we display each step's optimization strategies to get the most optimal LDA model in different steps.

All in all, the entire step of training LDA model is summarized below :

i pre-process corpus
–select the different re-constituted corpus and the prior knowledge embedding techniques along with it (Section 5.1.1)
–delete the rare terms and common terms of corpus (Section 5.2.2)

ii select LDA model (Section 5.3.1)
–set the parameters of LDA model, e.g. chunksize and number of topics
–select the most outperforming LDA model by the silhouette width metric regardless of the random state

iii cluster terms
–form term clusters (Section 3.2.1)
–thin term clusters (Section 3.2.2)

iv evaluate term clusters
–evaluate the term clusters with different metrics (Section 5.4.1)
–compare the optimal LDA model with other LDA models with seed information (Section 5.4.2)

5.4 Evaluation : Results, Analysis and Comparison

Given the term clusters of LDA training, two main factors could influence the evaluation step : the corpus field and the evaluation field. The corpus field has 10 different cases of the re-constructed corpus and 4 different approaches of embedding prior

TABLE 5.9 – The evaluation of five different metrics on the re-constituted corpus.

	silhouette width	**purity**	**ARI**	**MCC**	**AMI**
Case-GS preference	yes	yes	yes	yes	yes
Case-syntactic roles preference	yes	not clear	not clear	not clear	
Case-verbs preference	not clear	not clear	not clear	not clear	not clear
Case-training times preference	once	not clear	not clear	not clear	not clear
Approaches preference	Approach2 or approach4	approach4	approach4	approach4	approach4

knowledge. The evaluation field divided the evaluation metrics into 5 different measurements by their different focuses.

5.4.1 Results and Analysis

Regardless of the evaluation metrics, it is indispensable to proceed from the corpus field to judge the term clusters' results.

On the one hand, we intend to examine the clustering effects of LDA trained in the different cases of the corpus field. On the other hand, we bear in mind that embedding prior knowledge also influences the results respecting the Gold Standard.

To provide a clear view of those influence factors, we presented the resulting figures of the individual evaluation metrics in Appendix 7.4. Based on those metrics' performance, we summarize and display the outcomes of term clusters on account of the corpus fields, see Table 5.9. For the implicit meaning of 'C#A#', we could look back to Table 5.5 and Table 5.1 for more details, where the 'C#' corresponds to the different identifiers of cases and the 'A#' corresponds to different identifiers of approaches.

As discussed above, the corpus field is divided into two parts : 10 different cases of re-constructing corpus and 4 different approaches of embedding prior knowledge. The different cases refer to the different property categories. For instance, in Table 5.9, the category *Case-GS preference* indicates that whether the performance is preferable when all the input terms belong to the Gold Standard, and it is signified by the comparison of {case 1 ; case2} and the rest of cases ; the category *Case-syntactic roles preference* denotes that whether it is preferred when only the terms with important syntactic roles are contained as input, which is marked by comparing {case3} against {case5} ; the category *Case-verbs preference* means whether it is outperformed when

the corpus is enlarged with the co-occurred verbs of the remaining NPs, by comparing {case1} to {case2}, {case3} to {case4} and {case5} to {case6}; the category *Case-training times preference* examines whether the twice re-constructed corpus exceeds the others, by contrasting {case#} to {case#-T}. Besides, the category *Approaches preference* expresses whether it is encouraging to apply a certain approach to embed prior knowledge on the corpus by comparing the overall performance in all cases.

The results of **silhouette width** are shown in Figure 7.13 of Appendix 7.4. The negative value or even -1 implies that there are too many or too few clusters,

while the value 1 means the clusters are well separated from each other. Fortunately, all of the results are positive values in this figure, which reveals that the clusters are separated in a balanced way. The high value of case1 and case2 denotes the best compactness and separateness than other cases, which conforms to the fact that the cases with gold standard terms as input are preferred. The decreasing trend from case3 to case5 indicates the syntactic role preference of NPs. Comparing the twice trained cases to the original cases from case3 to case6, we can notice the main decreasing trends, which means more training times do not bring better performance. In general, we observe that approach 2 and approach 4 surpass other approaches for most of the cases.

The results of **purity score** are shown in Figure 7.14 of Appendix 7.4. The value 0 implies no relationship between the groups of terms and the Gold Standard's separations. On the opposite, the value 1 means the perfect matching between them. The resulting value of the purity score ranges from 20% to 55%, which reveals the degree of connection between the resulting groups with the Gold Standard. It is evident that case1 and case2 have around 3 times higher purity than the other cases; it shows the strong GS preference for groups' purity.

For the other cases, with a value of around 20%, it is difficult to distinguish the difference among them. Overall, approach4 is quite outperforming other approaches in a large part of cases.

For the results of **ARI, MCC and AMI**, their results are shown in Figure 7.15, 7.16 and 7.17 of Appendix 7.4. The value 0 indicates the random groups of terms compared to the Gold Standard, while the value 1 means the perfect matching. The trends of results are similar to that of purity scores. The main difference is that most of the cases, except for case1 and case2, were fluctuated around the base value 0%, which means the groups of terms do not conform to Gold Standard. Even if like this, we could

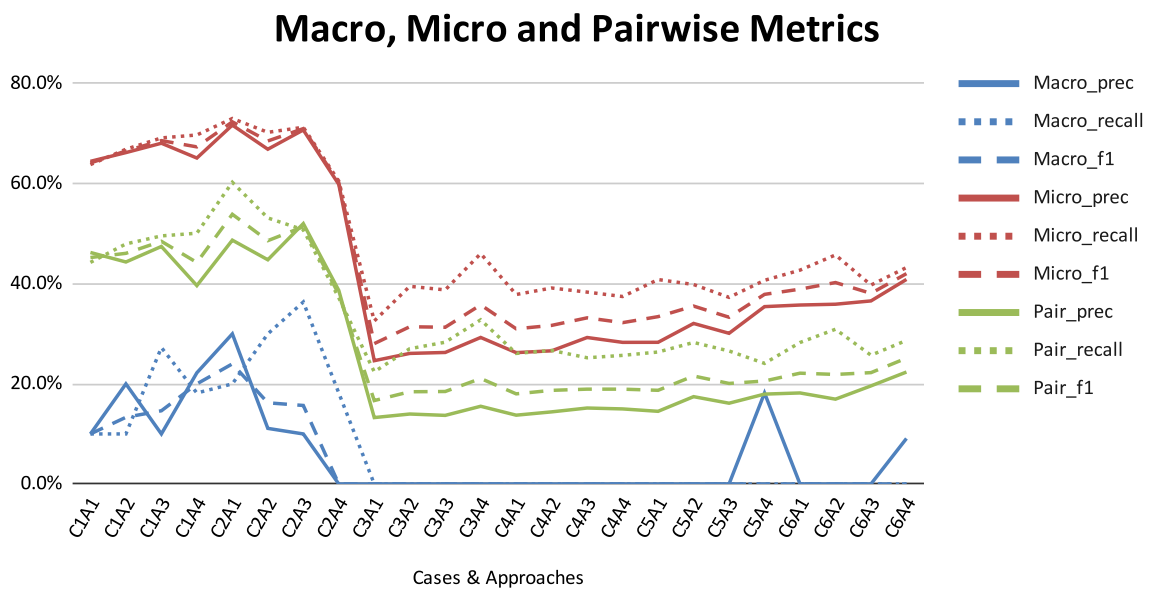


FIGURE 5.5 – The Macro, Micro and Pairwise evaluation results for different cases and approaches.

Notes : The results are averaged for 10 times with parameters : 50 topics and 50 chunksize. It is applied on the Computer Science corpus by applying the filtering ratio of lower bound to 0.2.

observe that approach 4 has the best performance in most cases.

From these facts, we may conclude that the cases with GS corpus are far more outperformed than other cases, which is seen as the upper limit of all the cases. The biggest difference of GS corpus against the other corpus is the lack of "unknown" or "other" terms, which means their corpus only contains domain-specific terms. We can induce that the exclusion of domain-unrelated terms in the input is good for term clustering. Furthermore, it is hard to conclude that the corpus has a preference in terms of syntactic roles of terms, the addition of verbs, or the training times. But it is easier to perceive that approach 4 has a slight distinction from other approaches on the performances of the evaluation metrics.

It is hard to recognize the difference between cases in most of the individual evaluation metrics. To solve this problem, we intend to apply the **macro, micro, and pairwise** evaluation metrics to evaluate the term clusters in a continuous vision. Fortunately, the difference between cases turns to be more significant than the former metrics. In Figure 5.5, the string in horizontal axis 'C#A#' denotes the case# with approach#. The comparison between different cases shows that the GS corpus ('C1A#' and 'C2A#')

achieves around 70% in the micro precision score. On the one hand, we notice a slight increase resulting from verbs' inclusion (i.e. from 'C1A#' to 'C2A#') and a considerable increase by applying the corpus regardless of the syntactic roles(i.e. from 'C3A#' and 'C5A#'). On the other hand, we note that the local peaks of precision and F1 metrics are mostly located at approach 4. Thus the 'C6A4' reaches the highest values except for case1 and case2. To have a clear view of the concrete clustering results of 'C6A4', we provide an example in Figure 7.20 in the Appendix section.

In brief, we are convinced that the GS corpus provides the best term clusters obtained from LDA models. However, this kind of corpus is artificially constructed, respecting the given Gold Standard. Besides, we found that the NPs corpus accompanying with their co-occurred verbs with Approach4 techniques(i.e.'C6A4') provides the best performance in term clusters.

5.4.2 Comparison

Once we have selected the optimal case (i.e.'C6A4') of clustering terms among many different cases, we learn that the re-constructed corpus embedding with prior knowledge could achieve the local optimal results than the normal LDA training. Likewise, some other seed-embedding topic models follow the same idea to guide the topic model to perform in a preferred way. It is necessary to compare our optimal cases with other topic models, e.g., z-label LDA [10] and seeded LDA [11].

From the discussion on the various evaluation metrics, we note that the application of micro precision conforms to our major interests on term clustering, respecting the Gold Standard. Thus we would like to apply only the micro precision to measure the performance of term clusters with the comparison between models, i.e. the selected case on LDA, z-label LDA, and seeded LDA. To thoroughly evaluate different models, we would implement these models in two different corpora like in Section 3.3 : the Computer Science(CS) Corpus[3] and the Reuter Corpus[4]. The information of seed keywords of these two corpora is described in Section 3.3.2 and their statistics are presented in Table 3.5. If you are interested in the methods of seed word assignments regarding the different number of topics, please check the examples in Appendix 7.21.

The **z-label LDA** could modify the inference of latent topics with a flexible degree,

3. https://data.mendeley.com/datasets/9rw3vkcfy4/6
4. Reuters-21578 : http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

ranging from 0 to 1. Here we are interested in the three values as discussed in Section 4.4.1, namely no constraint (value 0), soft constraint (value 0.5), and hard constraint (value 1). In Figure 5.6, these three constraints are signified by different colors. The micro precision of the case 'C6A4' with topic 200 is presented as the fixed value indicating line. It is obvious that the trend of the micro precision on CS corpus decreases steadily, while the trend on Reuter corpus decreases dramatically until reaching the lowest point at 40 and increases a bit then. Overall, the CS corpus reaches higher micro precision than the other corpus for z-label LDA. Also in CS corpus, we can observe that the hard constraint (value 1) outperforms the other constraint values, and all of them are locating above case 'C6A4'. However, in Reuter corpus, there is no clear clue to distinguish the performance of different constraints. Thus we can rank roughly from the best-performed models to the worse ones :

$$zlabel-hardConstraint \gg zlabel-soft \mid noConstraint \gg optimalCase-'C6A4' \quad (5.1)$$

The **seeded LDA** handles the learning of topics by applying the topic alignments of seed words. The M1 model only benefits from the modification over the topic-term distributions, while the seeded LDA benefits from the integrated model of both topic-term distribution and document-topic distribution. At the same time, we present normal LDA as well. In Figure 5.7, it is obvious that the trends of CS corpus and Reuter corpus are decreasing along with the increase of topics and ending with the similar trends below that of case 'C6A4'. We can observe that in the CS corpus the seeded LDA outperforms the M1LDA and the M1LDA outperforms the normal LDA, but all of them are below case 'C6A4'. Thus we can rank them from the best-performed models to the worse ones :

$$optimalCase -' C6A4' \gg seededLDA \gg M1LDA \gg normalLDA \quad (5.2)$$

In the Reuter corpus, even though the micro precision is always lower than that of CS corpus, it ends with similar results once the number of topics reaches 100. The low values in the Reuter corpus are due to the different number of labels (core concepts) for the different domains, e.g., 11 core concepts in CS corpus and 5 core concepts in Reuter corpus. For **z-label LDA**, it is difficult to recognize the best constraint of LDA. In parallel, for **seeded LDA**, we confront the same problem of recognizing the best models

of them. However, in general, the z-label LDA outperforms the variants of seeded LDA. All in all, the selected optimal case of LDA surpasses the Seeded LDA but not z-label LDA.

5.4.3 Ontology Visualization

Based on the clustering results that we acquired from case 'C6A4', we succeeded to build up an modular ontology equipped with the discovered taxonomic relations in Section 5.1.2.

In Figure 5.8, we show the overview of the resulting ontology. The orange nodes indicate the core concepts ; the dark yellow nodes indicate the sub-concepts that link the core concepts with taxonomic relations (according to WikiData knowledge database). For Figure 5.9, the zoom-in resulting ontology is presented. The green nodes indicate the terms from topical clusters. The orange links represent is-a relations and the grey links represent cluster-to relations, where the green nodes gathering together by grey links indicate the terms from the same cluster.

5.5 Summary

Before the variation employment of LDA, we explored the optimal parameter setting during experiments. We observed that the optimal parameters are not fixed for the different corpus. If a new corpus is introduced, we need to go through all of the candidate parameters to find the optimal option, i.e. number of topics and the chunksize value. Also, we found that the random effect of LDA does not influence a lot the quantity of clustered terms, but alters a bit of term' affiliations to the clusters.

In the aspect of the corpus properties, we noticed the overwhelming performance for GS corpus, in which the corpus includes only the domain-related terms. However, the extra training times of LDA do not bring significant performance. It is also hard to conclude that the corpus has a preference in terms of syntactic roles of terms and the addition of verbs. In the aspect of the prior knowledge embedding techniques, we remarked that the combined approaches of core concept replacement and sub-domain knowledge supplementation, i.e. approach4, stand out from other approaches regarding all of the metrics. Besides, we found that the NPs corpus accompanying

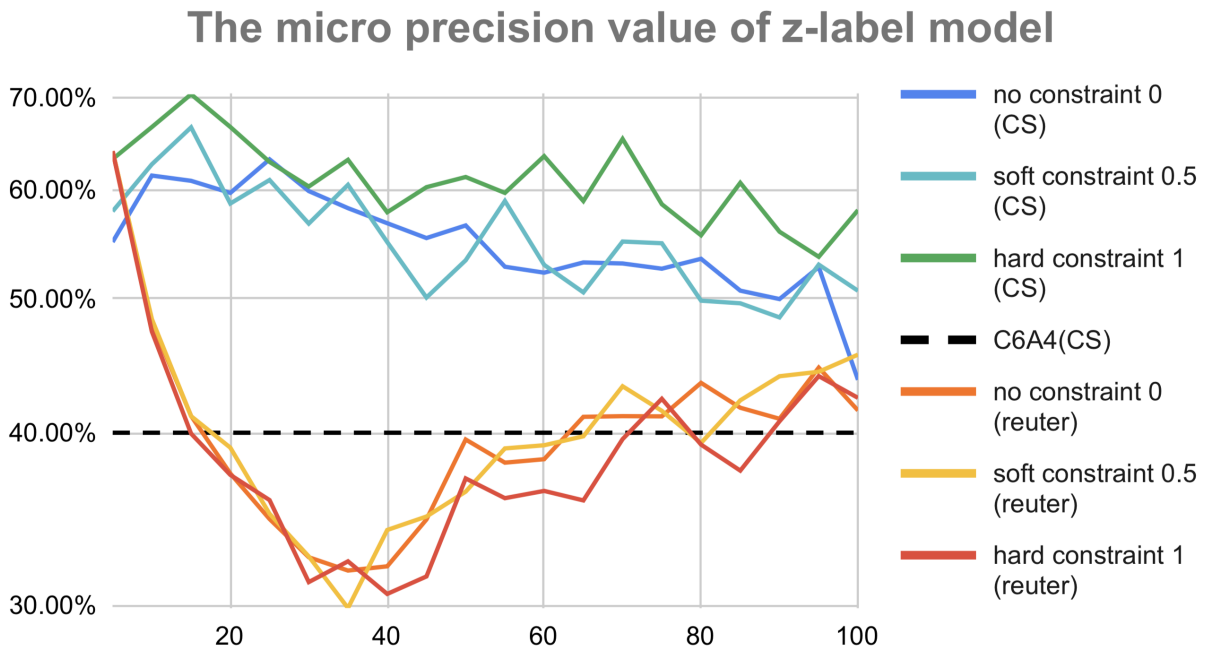


FIGURE 5.6 – The micro precision value of z-label model.
Notes : The X-axis is the value of micro precision ; the Y-axis denotes the number of topics.

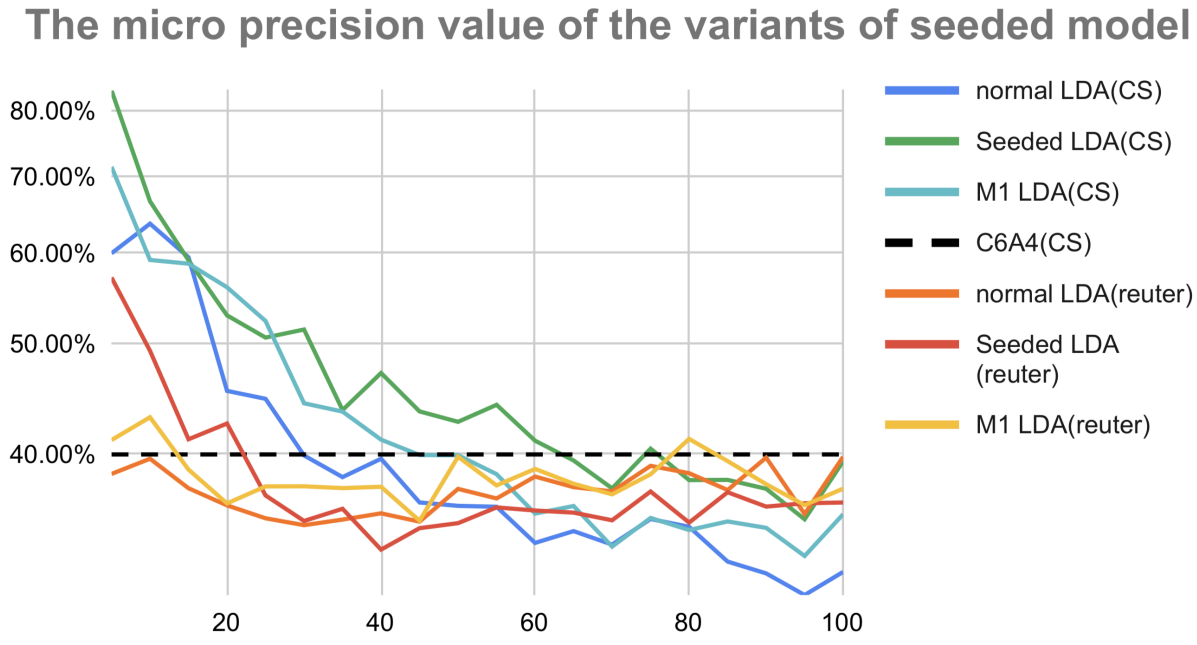


FIGURE 5.7 – The micro precision value of the variants of seeded model.
Notes : The X-axis is the value of micro precision ; the Y-axis denotes the number of topics.

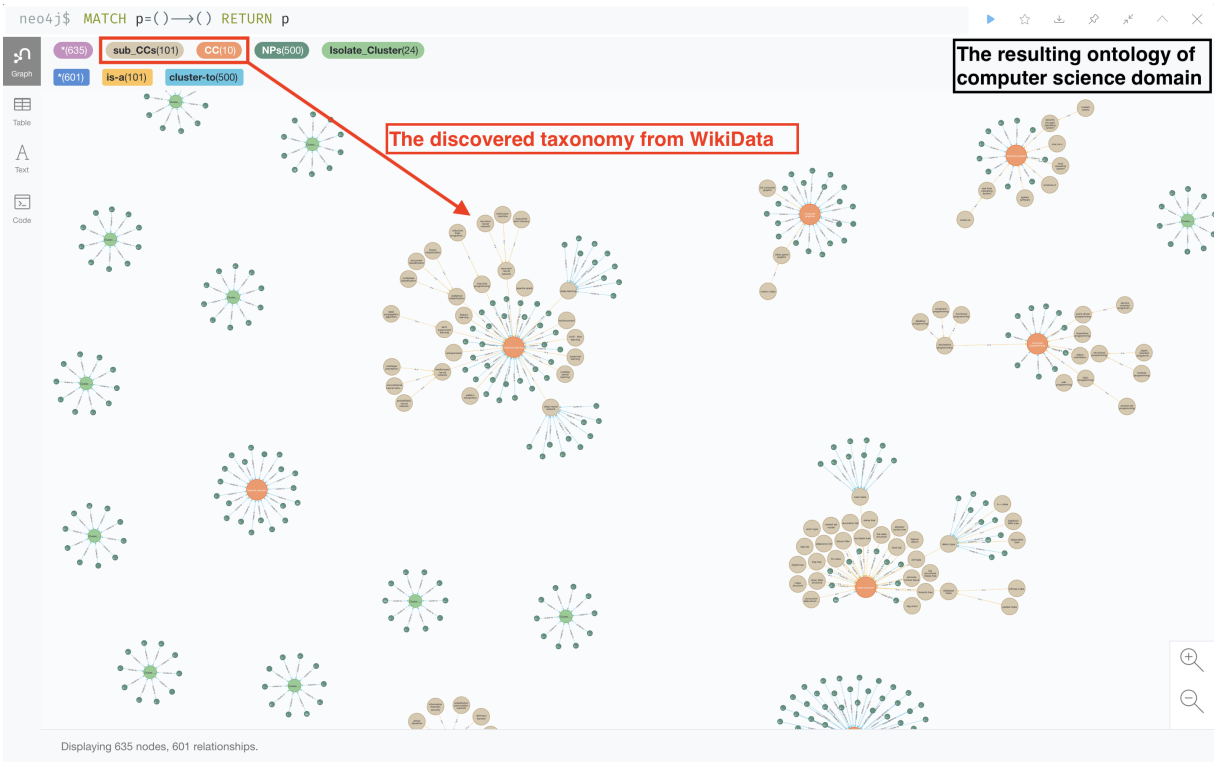


FIGURE 5.8 – The overview of the resulting ontology.

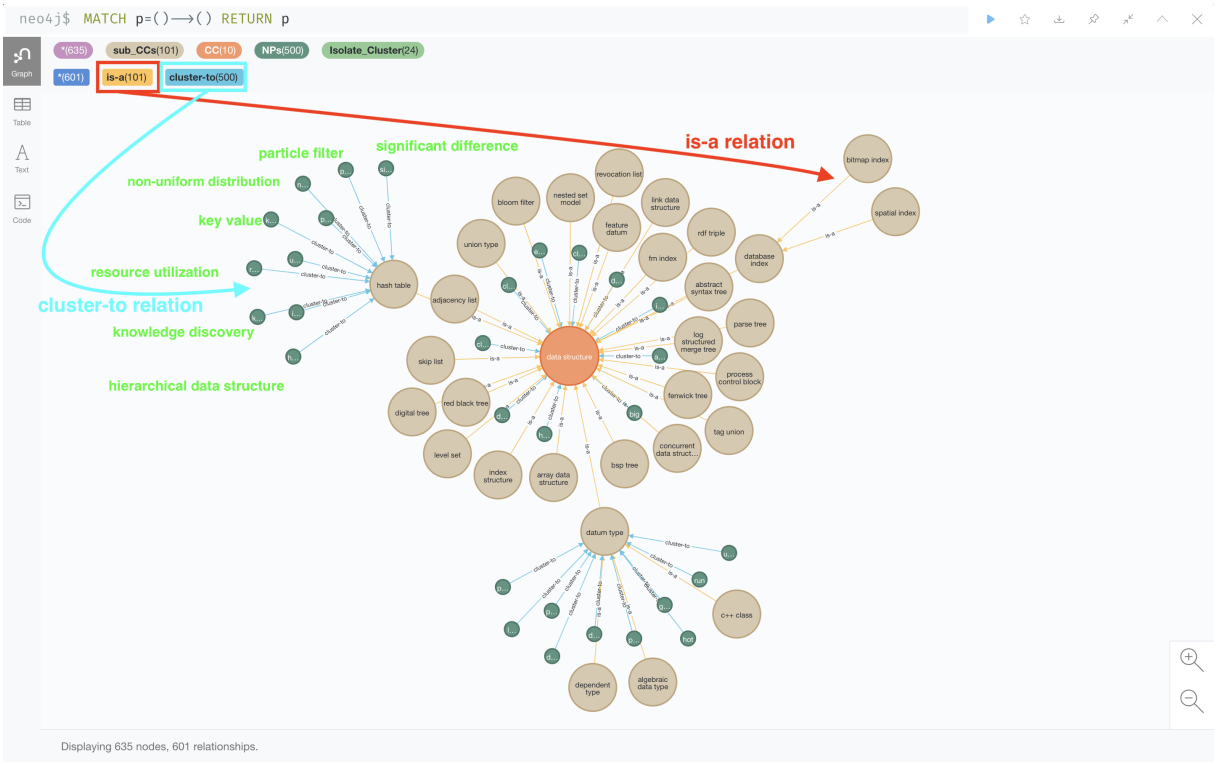


FIGURE 5.9 – The zoom-in of the resulting ontology.

with their co-occurred verbs with Approach4 techniques(i.e.'C6A4') provides the best performance in term clusters.

Based on the best employed LDA, we compared it to other prior knowledge-guided LDA, i.e. z-label LDA, and seeded LDA in two different corpora. As for z-label LDA, we perceived that, in CS corpus, the hard constraint(full usage of seed terms) outperforms the other constraint values, and all of them exceeded the performance of the best case 'C6A4'. However, in Reuter corpus, there is no clear clue to distinguish the performance of different constraints. For the seeded LDA, it is obvious that the trends of CS corpus and Reuter corpus were decreasing along with the increasing of topics and ending with the similar trends below that of case 'C6A4'. In brief, the selected optimal case of LDA surpasses the Seeded LDA but not z-label LDA.

In terms of taxonomic discovery techniques, the statistics of extracted terms implied that the knowledge base approach could extract the hyponyms that are significant in corpus occurrences. However, the noun modifier relationships approach is much easier and faster to execute and acquire taxonomic results.

Briefly, this chapter presents the various concerns to improve the utilities of LDA towards modular ontology building.

CHAPITRE 6

CONCLUSION

The aim of this dissertation *Enhancing LDA for Ontology Learning* is to explore the potential and valuable utilities of LDA to learn an ontology. The LDA models have been widely applied to represent the topics, but to date, no systematic workflows are known to compare and use them to cluster terms, so that the clusters would further compose the modules of an ontology. For this reason, we decided to improve the term clustering performance of LDA models, with the aim of generating those clusters who possess the meaning closing to the meaning of the sub-modules of a domain ontology.

6.1 Contributions

This thesis firstly introduces the fundamental procedures of ontology constructions and presented the ontology learning techniques from both the statistical and the linguistic sides in Chapter 1.

In the first contribution, the framework of the unsupervised term clustering as a task towards ontology learning is presented. Chapter 2 listed and summarized the essential compositions for term clustering algorithms. Then in Chapter 3, the various term representation techniques are proposed and experimented with the different classical clustering algorithms, e.g. K-Means, k-medoids, DBscan, affinity propagation, and co-clustering. The results suggest that the basic and the weighted co-occurrence representation shows better performance than the condensed co-occurrence representation. Moreover, the combination of word topic representation and the co-clustering algorithm is considered as the optimal pair of the classic term clustering strategy.

Comparatively, we also introduce the strategy to apply the topic model LDA directly for term clustering. The results indicate that the LDA-based strategy achieves an overwhelming precision higher than the majority of classic term clustering strategies in a framework. Therefore, the optimization of the LDA-based clustering strategy

is proven to be beneficial for acquiring the term clusters for ontology learning.

The second contribution talks about the semi-supervised modular ontology learning, in which the meaning of term clusters are driven by the core concepts of a domain. It succeeds to expand the topic models' utilities for modular ontology learning. Chapter 4 shows the variations of topic models from a simple statistic model to the basic LDA model, and even to the extensions of the LDA model, which take advantage of seed information to acquire the desired topic features. In Chapter 5, the mechanisms of core concept replacement and subdomain knowledge supplementation are proposed and applied as the knowledge embedding technique over the corpus. Besides, we also study the influencing factors of LDA, e.g., the syntactic roles of NPs, the inclusion of verbs occurring with NPs, and the number of LDA training times. It turned out that the combination of these two knowledge embedding techniques is outperformed,i.e.'C6A4', and the corpus with the inclusion of verbs occurring with NPs shows an outstanding impact on the performance of term clusters.

In terms of taxonomic discovery techniques, there are two resources of acquiring taxonomic relations between terms : from the textâs linguistic features, e.g. noun modifier relationships ; and the published taxonomy of common knowledge bases, e.g. Wikidata and DBpedia. In the comparison of the discovered terms, the results support that the knowledge base approach could extract the hyponyms that are significant in corpus occurrences. However, the approach of noun modifier relationships is much easier and faster to execute and acquire taxonomic results.

In brief, we made broad research on ontology learning procedures, including the extraction of significant terms, the discovery of similar terms, the formation of term clusters, and the taxonomic relation detection from plain text or from knowledge bases. Our thesis started by measuring the effectiveness of the LDA-based clustering strategy and continued to present the various concerns to improve the utilities of LDA towards modular ontology building. We present a detailed and integrated workflow to explore the LDA applications for the term clustering purpose, with respecting to aggregate the meaningful terms for modules of a domain ontology.

In the aspect of programming, the main programming language that I used in the experiments is Python 3.7. From steps to steps, I tried to rewrite the previous codes into a modular and simplified version. Finally, this thesis could be implemented by a few thousand lines of code.

6.2 Perspectives

My future research plans revolve around developing a reusable and knowledgeable ontology/knowledge graph of humans. The extraordinary complexity of humans' understanding makes this a rich topic with many possible avenues of investigation.

In the near future, I would like to employ modular ontology to ameliorate the conceptual structure of knowledge graphs and enrich the knowledge graph with more domain-related terms. To achieve this goal, based on the results that we have acquired, firstly, we would like to remove the non-significant terms in clusters. There are many interesting term weighting techniques that could be used to distinguish the terms by their statistic features. Also for the clusters in the same sub-domain, we believe that the closeness in the feature space between clusters and the valuable concepts of the knowledge graph will help interpret these clusters by the instantiated terms of the closest concept.

To provide the more reliable knowledge between text, I am also interesting on discovering more relations between terms. Following the same idea of Chapter 4, we would make use of seed terms pairs for relation discovery. Despite of applying the knowledge into document levels, the prior knowledge, i.e. seed pairs, would be used for data augmentation to get the acceptable training corpus. In such a semi-supervised manner, we will concentrate on some specific relation extractions, i.e. component-whole relations and cause-effect relations.

In the far future, my agenda is not limited to building more accurate and up-to-date ontology (although this certainly remains a challenge), but encompasses several other problems, including discovering more feasible relations for different end-use needs ; connecting to the other resources, not only text, e.g., video ; extending to some knowledge-intensive domains, e.g., medicine, education and law ; and applying my algorithms to the dataset outside of non-structured data, including semi-structured data, structured data or even labeled data. I anticipate that the applications of this research will be numerous and diverse, for the simple reason that feasible ontology has immediate relevance to anything that involves speeding up searching, facilitating communication, and organizing the newly generated knowledge with the related background.

CHAPITRE 7

APPENDIX

7.1 The Top-N of Partition Size

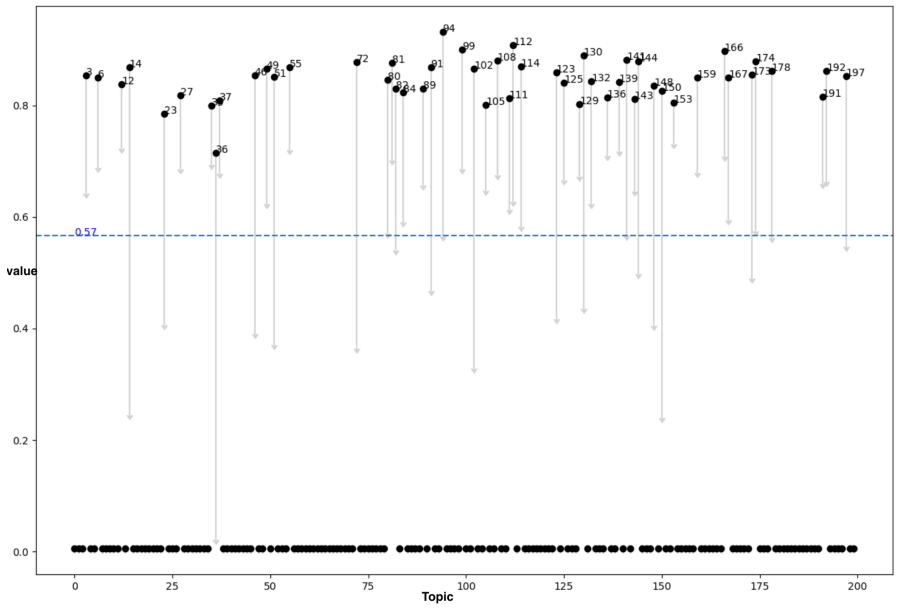


FIGURE 7.1 – The distinction of the **top-10** normalized maximum term probability in each topic partition.

Notes : The Y axis represents the value of $normalized\ p(w|t)$ of each topic in the X axis.

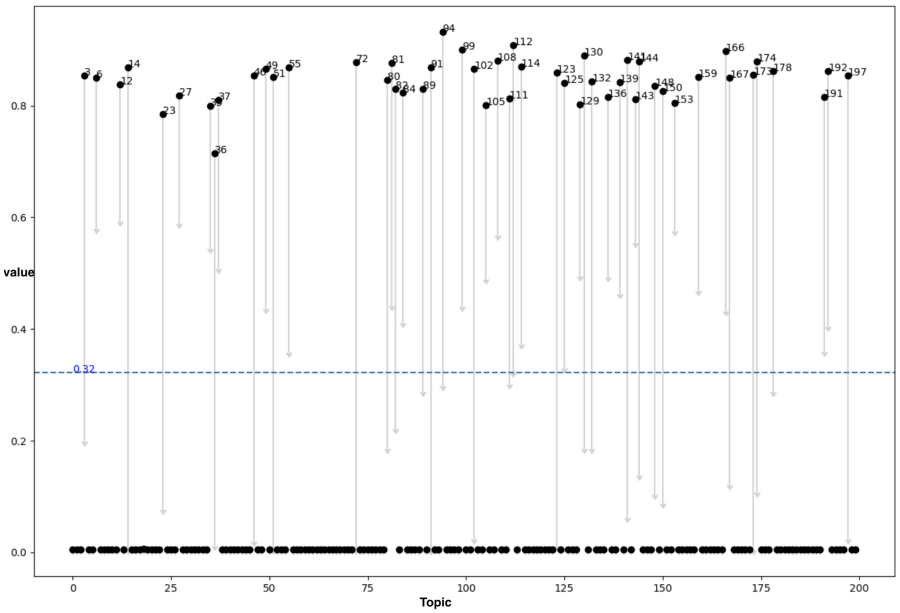


FIGURE 7.2 – The distinction of the **top-20** normalized maximum term probability in each topic partition.

Notes : The Y axis represents the value of $normalized\ p(w|t)$ of each topic in the X axis.

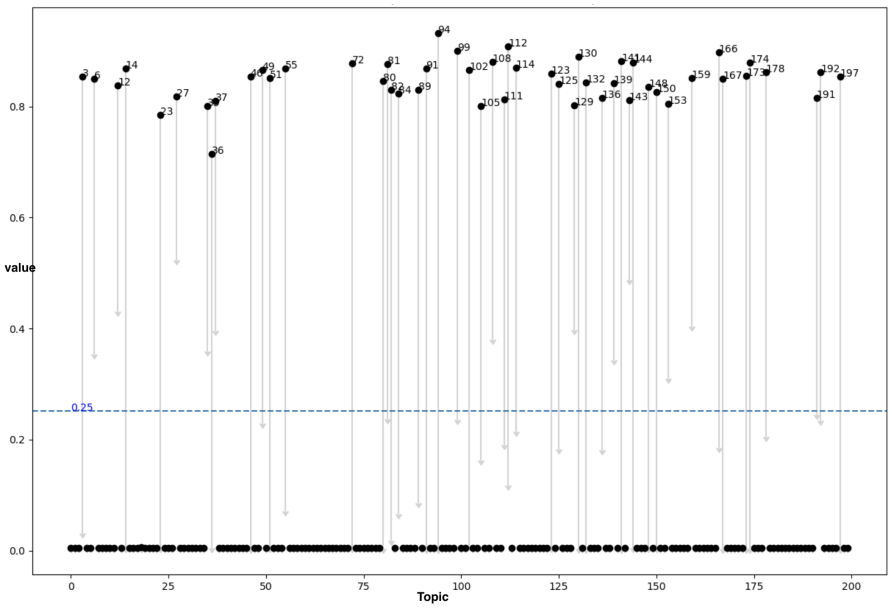


FIGURE 7.3 – The distinction of the **top-30** normalized maximum term probability in each topic partition.

Notes : The Y axis represents the value of $normalized\ p(w|t)$ of each topic in the X axis.

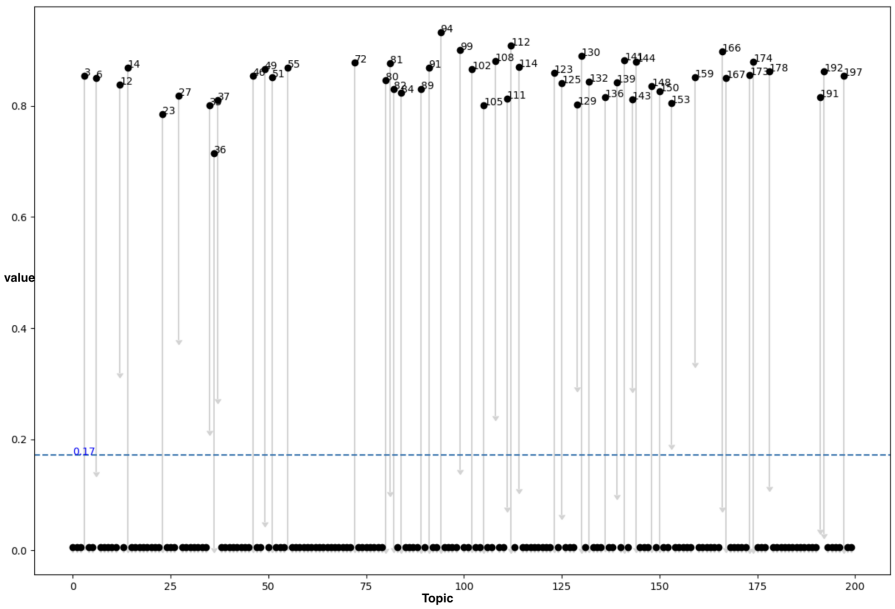


FIGURE 7.4 – The distinction of the **top-40** normalized maximum term probability in each topic partition.

Notes : The Y axis represents the value of $normalized\ p(w|t)$ of each topic in the X axis.

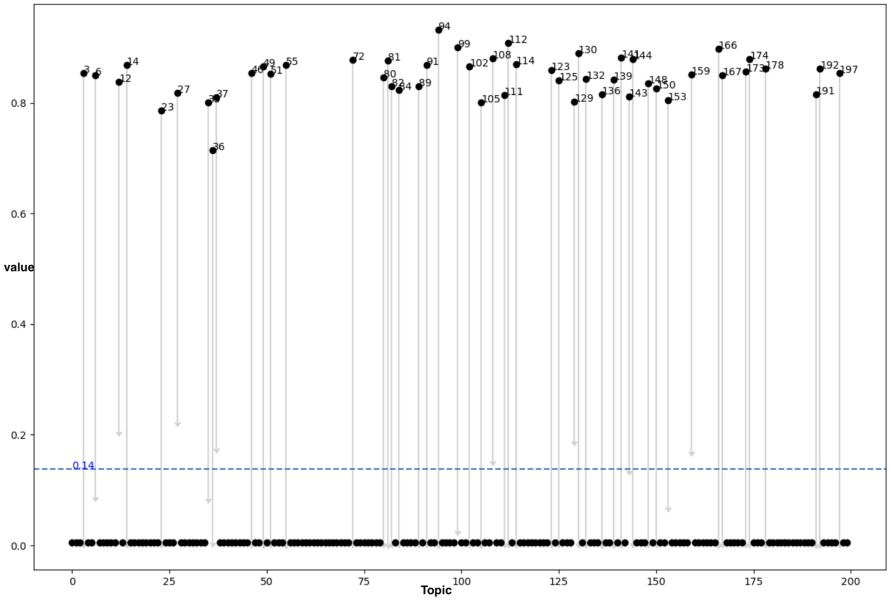


FIGURE 7.5 – The distinction of the **top-50** normalized maximum term probability in each topic partition.

Notes : The Y axis represents the value of $normalized\ p(w|t)$ of each topic in the X axis.

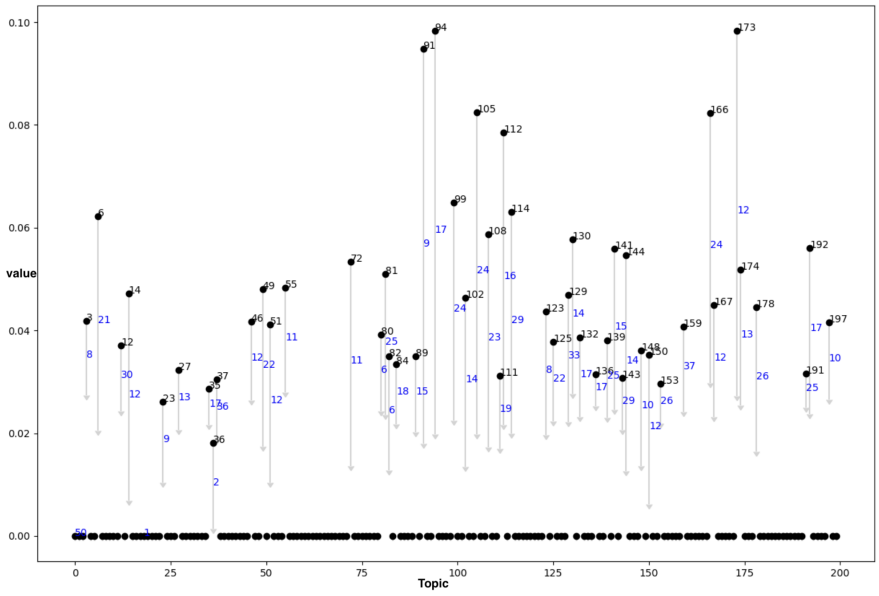


FIGURE 7.6 – The number of terms exceeding the threshold on top-50 probability in each topic partition.

Notes : The Y axis represents the value of $normalized\ p(w|t)$ of each topic in the X axis.

7.2 The Comparison between Two Clustering Strategies on CS corpus

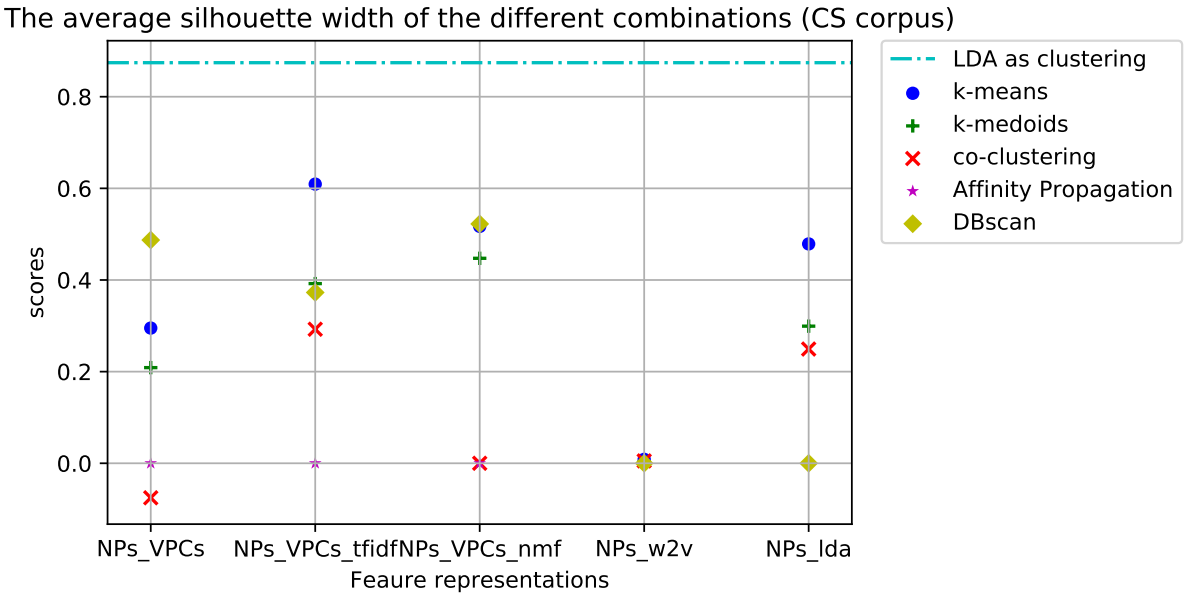


FIGURE 7.7 – The **silhouette width** of the two clustering strategies(CS corpus)

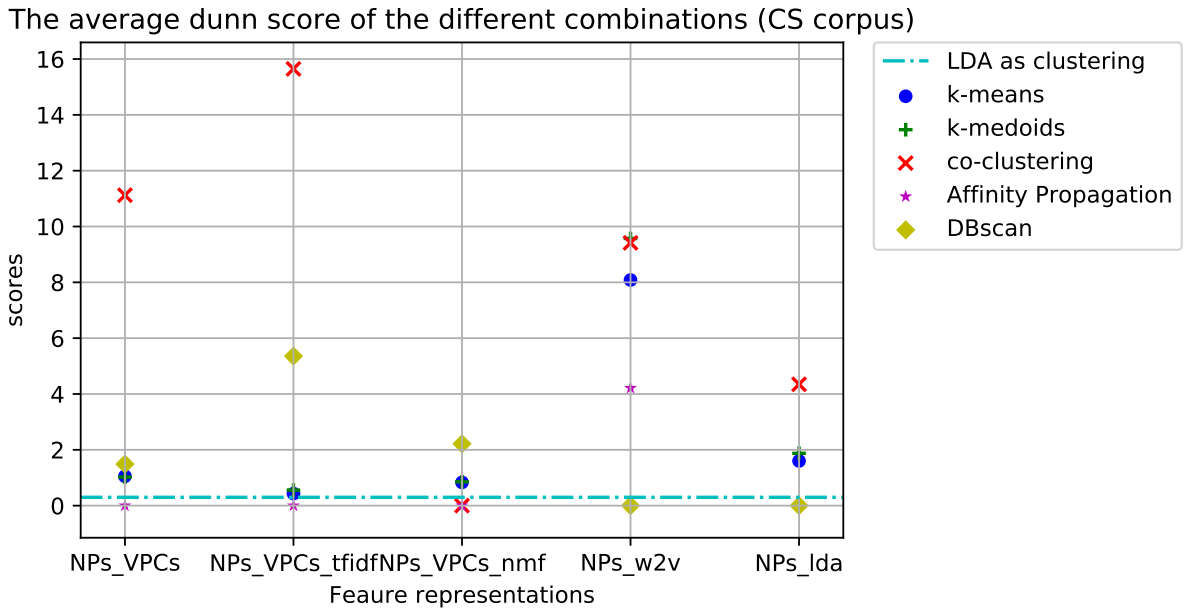


FIGURE 7.8 – The **dunn score** of the two clustering strategies(CS corpus)

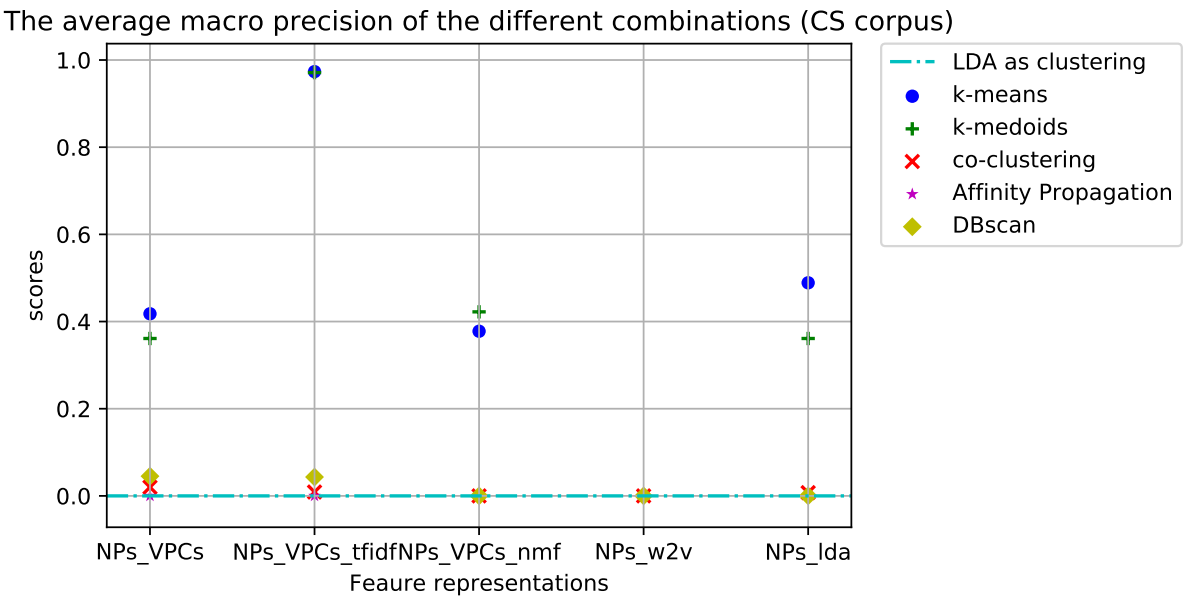


FIGURE 7.9 – The **macro precision** of the two clustering strategies(CS corpus)

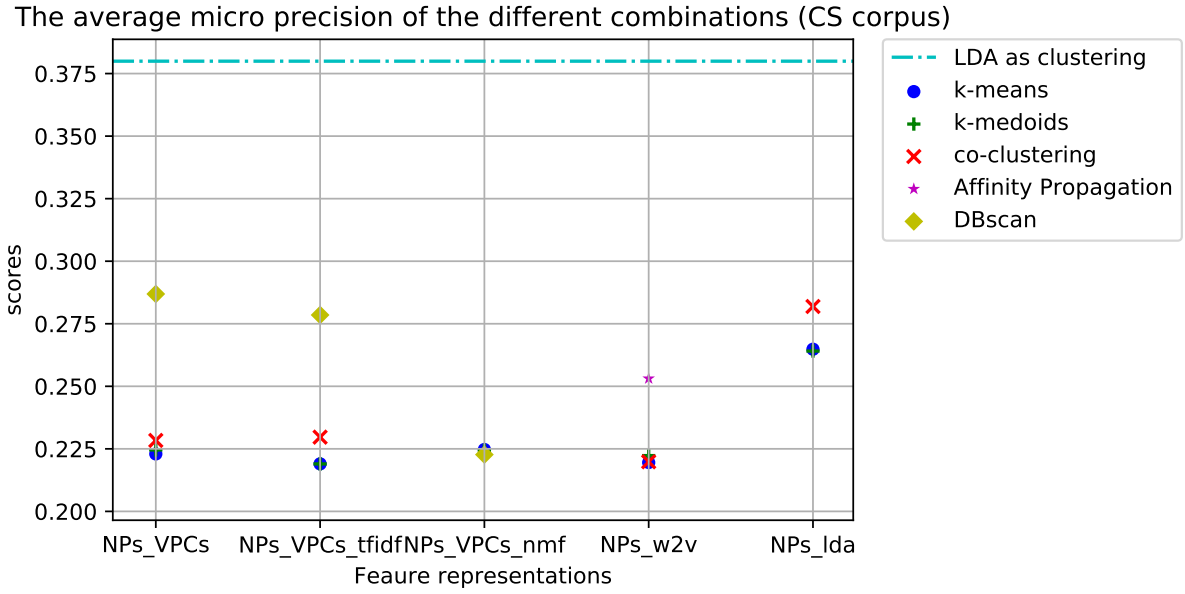


FIGURE 7.10 – The **micro precision** of the two clustering strategies(CS corpus)

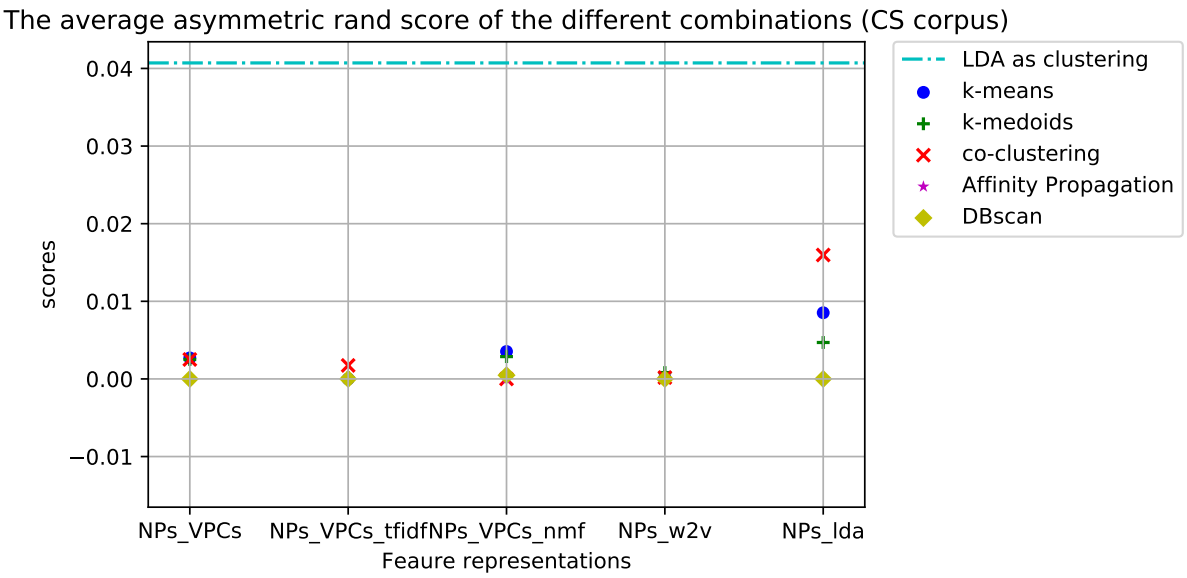


FIGURE 7.11 – The **asymmetric rand score** of the two clustering strategies(CS corpus)

7.3 Human Evaluation

Background Description

Domain	Computer Science
source of corpus	WebOfScience
corpus type	plain text
the size of corpus	6000 documents

Task

* give label to Noun Phrases(NPs) by the predefined 11 subdomains OR 'Unknown' OR 'Others'
* to alleviate the word, you can use abbreviation of those labels

Related and non-related Labels

11 subdomains	Abbreviation
Algorithm design	ad
Bioinformatics	b
Computer graphics	cg
Computer programming	cp
Cryptography	c
Data structures	ds
Distributed computing	dc
Machine learning	ml
Operating systems	os
Software engineering	se
network security	ns

Non related labels	Abbreviation	
Unknown	u	For those NPs, you know it belongs to Computer Science Domain, but you are not sure which subdomains it belongs to
Others	o	For those NPs, you know it does NOT belongs to Computer Science Domain

FIGURE 7.12 – A task explanation document for volunteers, which describes the annotation tasks detailed in Section 3.3.2.

7.4 Term Partition Evaluation

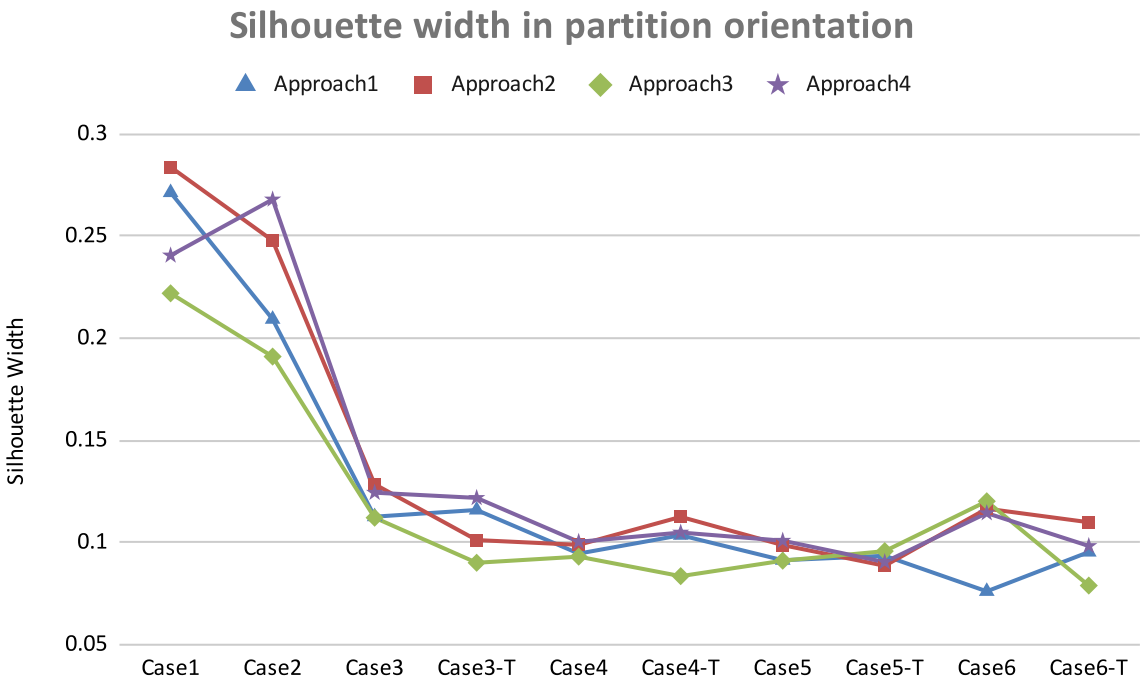


FIGURE 7.13 – The partition-oriented evaluation of 10 cases (including twice trained LDA model) in **silhouette width**.

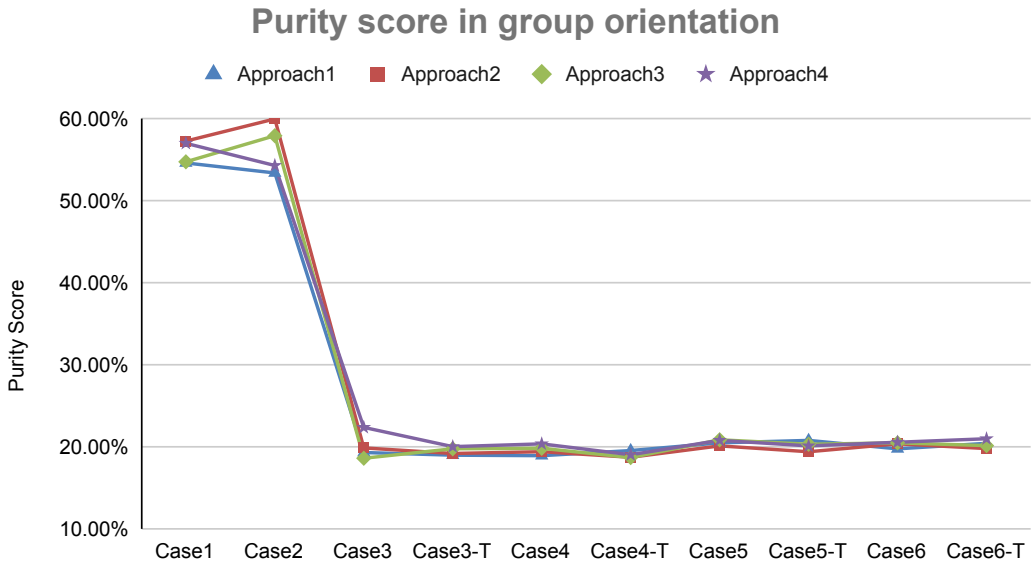


FIGURE 7.14 – The group-oriented evaluation of 10 cases (including twice trained LDA model) in **purity score**.

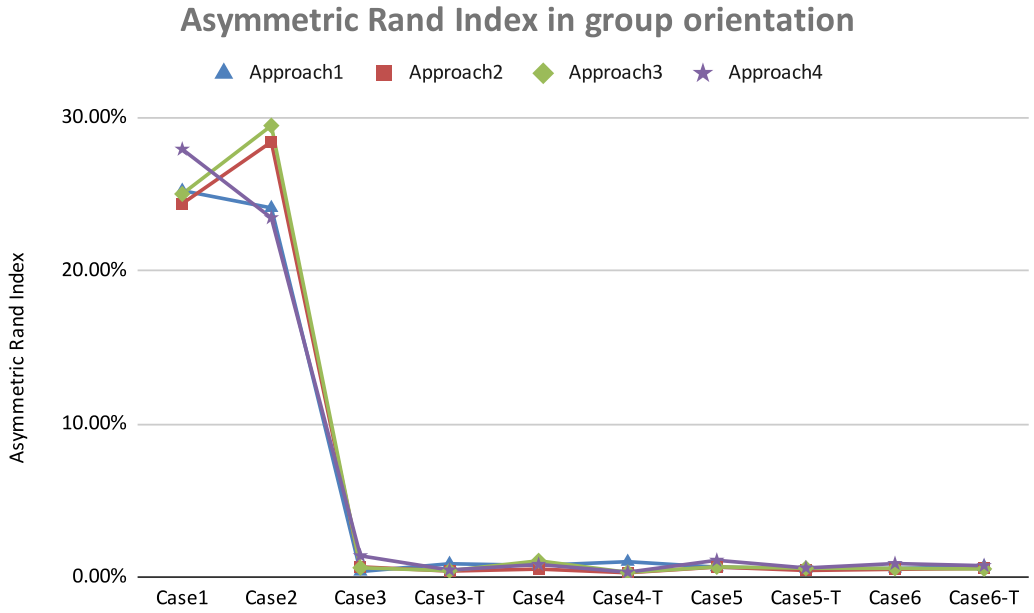


FIGURE 7.15 – The group-oriented evaluation of 10 cases (including twice trained LDA model) in **asymmetric rand index**.

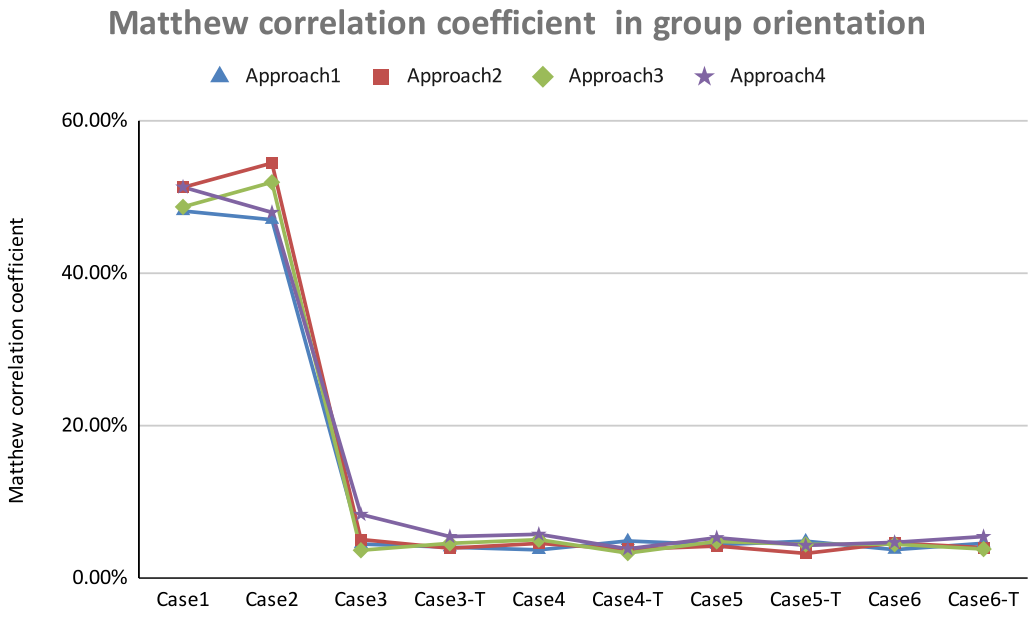


FIGURE 7.16 – The group-oriented evaluation of 10 cases (including twice trained LDA model) in **Matthew correlation coefficient**.

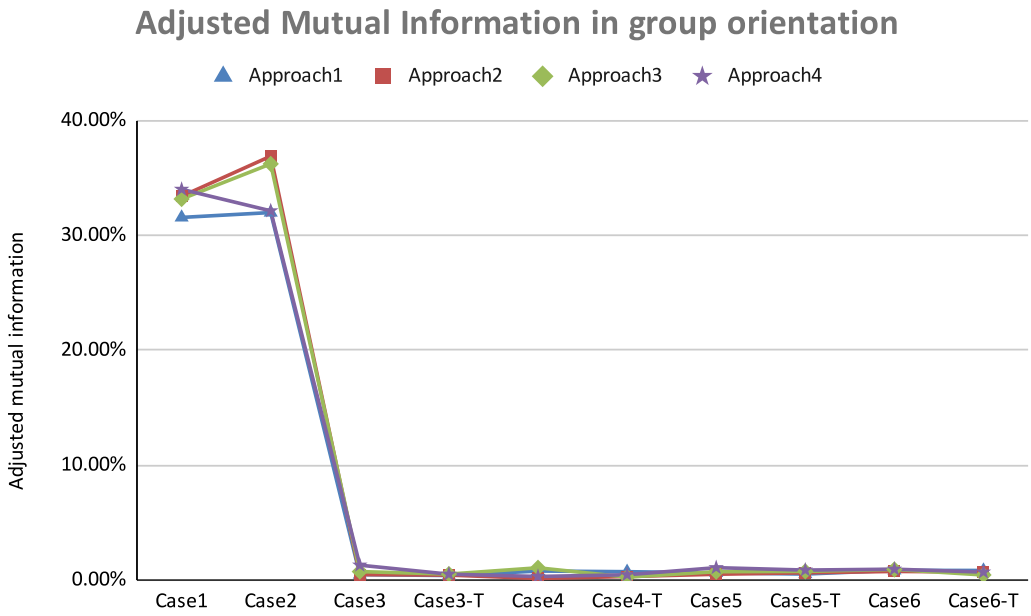


FIGURE 7.17 – The group-oriented evaluation of 10 cases (including twice trained LDA model) in **adjusted mutual information**.

7.5 The results of hyponym acquisition in DBpedia

x (step-ii)	y (step-iii)	z (step-iv)
:Category:Machine_learning	:Category:Inductive_logic_programming	:Category:Ontology_learning_(computer_science)
:Category:Cybernetics	:Category:Unsupervised_learning	:Category:Machine_learning_task
:Category:Learning	:Category:Reinforcement_learning	:Category:Machine_learning_researchers
	:Category:Classification_algorithms	:Category:Loss_functions
	:Category:Cluster_analysis	:Category:Log-linear_models
	:Category:Computational_learning_theory	:Category:Markov_models
	:Category:Structured_prediction	:Category:Evolutionary_algorithms
	:Category:Supervised_learning	:Category:Ensemble_learning
	:Category:Support_vector_machines	:Category:Deep_learning
	:Category:Applied_machine_learning	:Category:Datasets_in_machine_learning
	:Category:Kernel_methods_for_machine_learning	:Category:Data_mining_and_machine_learning_software
	:Category:Latent_variable_models	:Category:Artificial_neural_networks
	:Category:Learning_in_computer_vision	:Category:Artificial_intelligence_conferences
	:Category:Machine_learning_algorithms	:Category:Dimension_reduction
	:Category:Statistical_natural_language_processing	:Category:Bayesian_networks
	:Category:Semisupervised_learning	:Category:Genetic_programming
	:Category:Signal_processing_conferences	:Category:Signal_processing_conferences
	:Category:Genetic_programming	:Category:Semisupervised_learning
	:Category:Bayesian_networks	:Category:Statistical_natural_language_processing
	:Category:Dimension_reduction	:Category:Machine_learning_algorithms
	:Category:Artificial_intelligence_conferences	:Category:Learning_in_computer_vision
	:Category:Artificial_neural_networks	:Category:Latent_variable_models
	:Category:Data_mining_and_machine_learning_software	:Category:Kernel_methods_for_machine_learning
	:Category:Datasets_in_machine_learning	:Category:Applied_machine_learning
	:Category:Deep_learning	:Category:Support_vector_machines
	:Category:Ensemble_learning	:Category:Supervised_learning
	:Category:Evolutionary_algorithms	:Category:Structured_prediction
	:Category:Markov_models	:Category:Computational_learning_theory
	:Category:Log-linear_models	:Category:Cluster_analysis
	:Category:Loss_functions	:Category:Classification_algorithms
	:Category:Machine_learning_researchers	:Category:Reinforcement_learning
	:Category:Machine_learning_task	:Category:Unsupervised_learning
	:Category:Ontology_learning_(computer_science)	:Category:Inductive_logic_programming
		:Category:Decision_trees
		:Category:Clustering_criteria
		:Category:Cluster_analysis_algorithms
		:Category:Graphical_models
		:Category:AlphaGo
		:Category:Factor_analysis
		:Category:Structural_equation_models
		:Category:Genetic_algorithms
		:Category:Language_modeling
		:Category:Neural_network_software
		:Category:Social_network_analysis_software
		:Category:Datasets_in_computer_vision
		:Category:Deepfakes
		:Category:Nature-inspired_metaheuristics
		:Category:Gene_expression_programming
		:Category:Markov_networks
		:Category:Hidden_Markov_models
		:Category:Artificial_immune_systems
		:Category:Causal_inference

FIGURE 7.18 – The outputs of SPARQL query in DBpedia.

7.6 The example of Wikidata-Taxonomy

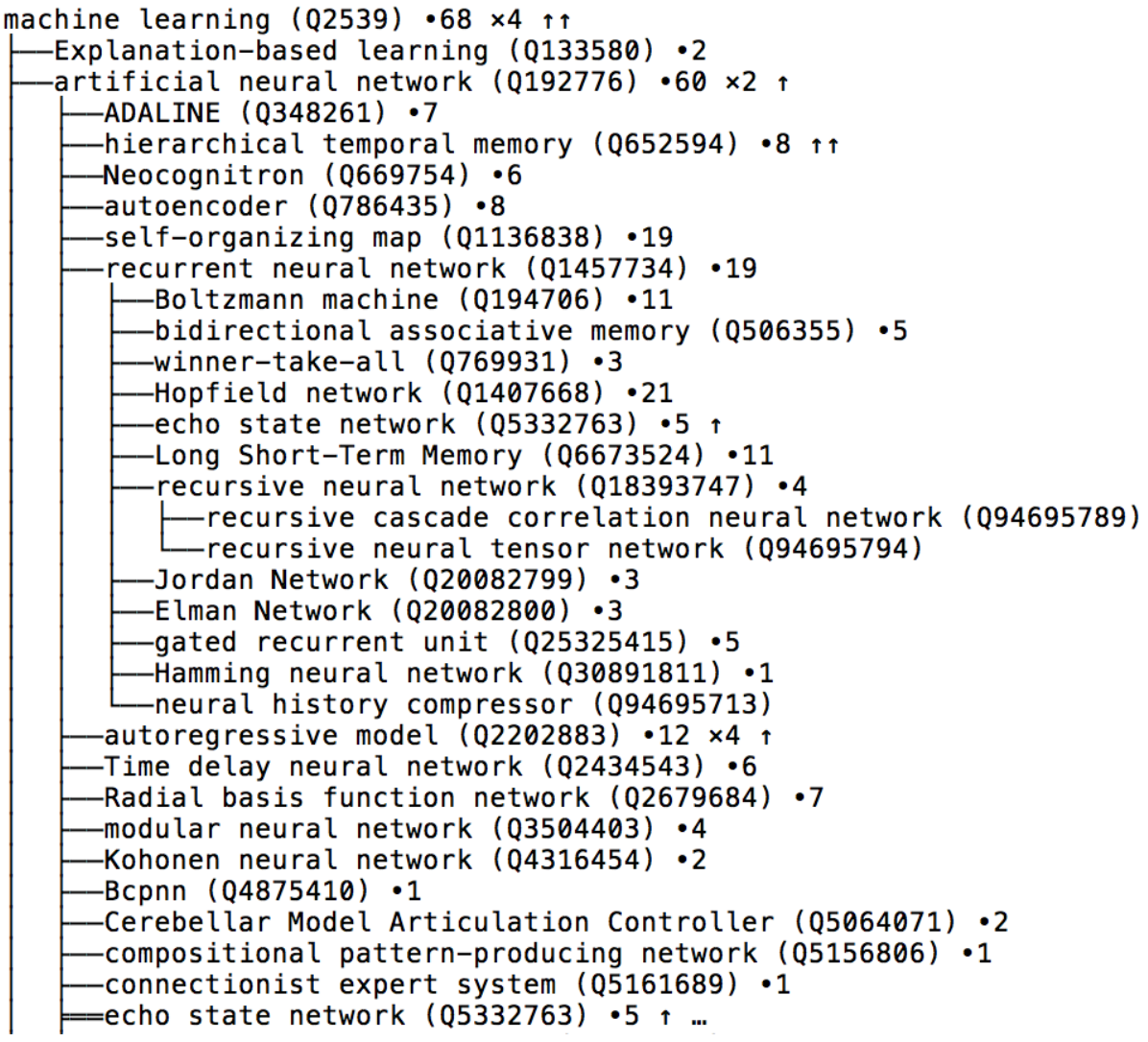


FIGURE 7.19 – The partial taxonomy of 'Machine Learning' by Wikidata-Taxonomy tool

7.7 The partial results of ’C6A4’ term clusters.

Cluster 1
computer graphic
3d depth image
behavioural image processing issue
bp mlp neural network
branch decomposition based algorithm design technique
common image analysis technique
comparative genomic analysis
continuous variable quantum key distribution protocol
current state art statistical image processing method
deep convolutional neural network model

Cluster 2
machine learning
agent based system
art feature selection method
automate computer vision based inspection system
base feature selection method
bi objective genetic algorithm
chaotic tent map
cla based classification algorithm
cloud based distributed system
complex statistical model

Cluster 3
cryptography
accurate fast image segmentation
aes-128 bit algorithm design consist
chebyshev chaotic map
coherent w state
computer graphics task
deep convolutional neural network
deep visual feature extraction
efficient elliptic curve cryptography
coherent ghz state

FIGURE 7.20 – The partial term clusters of ’C6A4’ case.

Notes : this result is trained based on Computer Science corpus with 50 topics. We only extracted 10 terms for each cluster. The underlined terms of each cluster belong to the same classes for evaluation.

7.8 The methods of selecting seed words for topics.

FIGURE 7.21 – An example of selecting the seed words regarding different number of topics.

Notes : This example lists all of the seed words for Reuter corpus. Each column includes the seed words of one core concepts (e.g. in bold). Each topic will be assigned with one seed word in horizontal direction from left to right (e.g. the blue arrows), until the required number of topics is achieved.

BIBLIOGRAPHIE

[1] P. CIMIANO, *Ontologies*. Springer, 2006.

[2] N. GUARINO, D. OBERLE et S. STAAB, « What is an ontology ? » In *Handbook on ontologies*, Springer, 2009, p. 1-17.

[3] P. BUITELAAR, P. CIMIANO et B. MAGNINI, « Ontology learning from text : An overview, » *Ontology learning from text : Methods, evaluation and applications*, t. 123, p. 3-12, 2005.

[4] Z. S. HARRIS, « Distributional Structure, » *WORD*, t. 10, *2-3*, p. 146-162, 1954. DOI : 10.1080/00437956.1954.11659520. eprint : https://doi.org/10.1080/00437956.1954.11659520. adresse : https://doi.org/10.1080/00437956.1954.11659520.

[5] W. WONG, W. LIU et M. BENNAMOUN, « Tree-traversing ant algorithm for term clustering based on featureless similarities, » *Data Mining and Knowledge Discovery*, t. 15, *3*, p. 349-381, 2007.

[6] D. M. BLEI, « Probabilistic topic models, » *Commun. ACM*, t. 55, *4*, p. 77-84, 2012. DOI : 10.1145/2133806.2133826. adresse : http://doi.acm.org/10.1145/2133806.2133826.

[7] K. BARKER et S. SZPAKOWICZ, « Semi-automatic recognition of noun modifier relationships, » in *COLING 1998 Volume 1 : The 17th International Conference on Computational Linguistics*, 1998.

[8] R. YANGARBER, R. GRISHMAN, P. TAPANAINEN et S. HUTTUNEN, « Automatic acquisition of domain knowledge for information extraction, » in *COLING 2000 Volume 2 : The 18th International Conference on Computational Linguistics*, 2000.

[9] D. M. BLEI, A. Y. NG et M. I. JORDAN, « Latent dirichlet allocation, » *Journal of machine Learning research*, t. 3, *Jan*, p. 993-1022, 2003.

[10] D. ANDRZEJEWSKI et X. ZHU, « Latent dirichlet allocation with topic-in-set knowledge, » in *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, 2009, p. 43-48.

[11] J. JAGARLAMUDI, H. DAUMÉ III et R. UDUPA, « Incorporating lexical priors into topic models, » in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, p. 204-213.

[12] T. R. GRUBER, « Toward principles for the design of ontologies used for knowledge sharing ? » *International journal of human-computer studies*, t. 43, *5-6*, p. 907-928, 1995.

[13] T. R. GRUBER et al., « A translation approach to portable ontology specifications, » *Knowledge acquisition*, t. 5, *2*, p. 199-221, 1993.

[14] W. N. BORST, « Construction of engineering ontologies for knowledge sharing and reuse., » 1999.

[15] R. STUDER, V. R. BENJAMINS et D. FENSEL, « Knowledge engineering : principles and methods, » *Data & knowledge engineering*, t. 25, *1-2*, p. 161-197, 1998.

[16] L. BURITA, P. GARDAVSKY et T. VEJLUPEK, « K-GATE Ontology Driven Knowledge Based System for Decision Support, » *Journal of Systems Integration*, t. 3, *1*, p. 19-31, 2012.

[17] D. OBERLE, S. LAMPARTER, S. GRIMM, D. VRANDEČIĆ, S. STAAB et A. GANGEMI, « Towards Ontologies for Formalizing Modularization and Communication in Large Software Systems, » *Appl. Ontol.*, t. 1, *2*, p. 163-202, avr. 2006, ISSN : 1570-5838.

[18] A. GANGEMI, C. CATENACCI et M. BATTAGLIA, « Inflammation ontology design pattern : an exercise in building a core biomedical ontology with descriptions and situations., » *Studies in health technology and informatics*, t. 102, p. 64-80, 2004.

[19] A. SCHERPA, C. SAATHOFFA, T. FRANZA et S. STAABA, « Designing Core Ontologies, » *Applied Ontology*, t. 3, p. 1-3, 2009.

[20] J. HOIS, M. BHATT et O. KUTZ, « Modular Ontologies for Architectural Design., » in *FOMI*, 2009, p. 66-77.

[21] B. C. GRAU, I. HORROCKS, Y. KAZAKOV et U. SATTLER, « A logical framework for modularity of ontologies., » in *IJCAI*, t. 2007, 2007, p. 298-303.

[22] M. EL GHOSH, H. NAJA, H. ABDULRAB et M. KHALIL, « Application of Ontology Modularization for Building a Criminal Domain Ontology, » in *AI Approaches to the Complexity of Legal Systems*, Springer, 2015, p. 394-409.

[23] M. KEET, *An introduction to ontology engineering*. Maria Keet, 2018, t. 1.

[24] J. ROGIER, *W3C Workshop on Semantic Web in Oil amp ; Gas Industry*, 2008. adresse : https://www.w3.org/2008/12/ogws-report.html.

[25] M. DÖRR, « The cidoc crm-an ontological approach to semantic interoperability of metadata, 2001, » *AI Magazine, Special Issue on Ontologies, Nov*, 2002.

[26] E. PRESTES, J. L. CARBONERA, S. R. FIORINI, V. A. JORGE, M. ABEL, R. MADHAVAN, A. LOCORO, P. GONCALVES, M. E. BARRETO, M. HABIB et al., « Towards a core ontology for robotics and automation, » *Robotics and Autonomous Systems*, t. 61, *11*, p. 1193-1204, 2013.

[27] M. BLÁZQUEZ, M. FERNÁNDEZ, J. M. GARCIA-PINAR et A. GÓMEZ-PÉREZ, « Building ontologies at the knowledge level using the ontology design environment. »

[28] M. FERNÁNDEZ-LÓPEZ, A. GÓMEZ-PÉREZ et N. JURISTO, « Methontology : from ontological art towards ontological engineering, » 1997.

[29] O. MEDELYAN, I. H. WITTEN, A. DIVOLI et J. BROEKSTRA, « Automatic construction of lexicons, taxonomies, ontologies, and other knowledge structures, » *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, t. 3, *4*, p. 257-279, 2013.

[30] T. R. GRUBER, « A translation approach to portable ontology specifications, » *Knowledge acquisition*, t. 5, *2*, p. 199-220, 1993.

[31] S. DESPRESS et S. SZULMAN, « Merging of Legal Micro-ontologies from European Directives, » *Artif. Intell. Law*, t. 15, *2*, p. 187-200, juin 2007, ISSN : 0924-8463.

[32] E. RILOFF et J. SHEPHERD, « A corpus-based approach for building semantic lexicons, » *arXiv preprint cmp-lg/9706013*, 1997.

[33] D. DAVIDOV et A. RAPPOPORT, « Classification of semantic relationships between nominals using pattern clusters, » in *Proceedings of ACL-08 : HLT*, 2008, p. 227-235.

[34] A. GANGEMI, C. CATENACCI, M. CIARAMITA et J. LEHMANN, « Modelling ontology evaluation and validation, » in *European Semantic Web Conference*, Springer, 2006, p. 140-154.

[35] O. KUTZ et J. HOIS, « Modularity in ontologies, » *Applied Ontology*, t. 7, p. 109-112, avr. 2012. DOI : 10.3233/AO-2012-0109.

[36] G. BESBES et H. BAAZAOUI-ZGHAL, « Modular ontologies and CBR-based hybrid system for web information retrieval, » *Multimedia Tools and Applications*, t. 74, *18*, p. 8053-8077, 2015.

[37] N. B. MUSTAPHA, M.-A. AUFAURE, H. B. ZGHAL et H. B. GHEZALA, « Modular ontological warehouse for adaptative information search, » Springer, 2012, p. 79-90.

[38] C.-S. LEE, Y.-F. KAO, Y.-H. KUO et M.-H. WANG, « Automated ontology construction for unstructured text documents, » *Data & Knowledge Engineering*, t. 60, *3*, p. 547-566, 2007.

[39] H. POON et P. DOMINGOS, « Unsupervised ontology induction from text, » in *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*, 2010, p. 296-305.

[40] A. CARLSON, J. BETTERIDGE, B. KISIEL, B. SETTLES, E. R. HRUSCHKA JR et T. M. MITCHELL, « Toward an architecture for never-ending language learning., » in *Aaai*, Atlanta, t. 5, 2010.

[41] W. WONG, W. LIU et M. BENNAMOUN, « Ontology learning from text : A look back and into the future, » *ACM Computing Surveys (CSUR)*, t. 44, *4*, p. 1-36, 2012.

[42] R. L. CIMIANO P. de Mantaras et L. SAITIA, « Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text, » in *16th european conference on artificial intelligence conference proceedings*, t. 110, 2004, p. 435.

[43] X. JIANG et A.-H. TAN, « Mining ontological knowledge from domain-specific text documents, » in *Fifth IEEE International Conference on Data Mining*, IEEE, 2005, p. 665-668.

[44] A. B. RIOS-ALVARADO, I. LOPEZ-AREVALO et V. J. SOSA-SOSA, « Learning concept hierarchies from textual resources for ontologies construction, » *Expert Systems with Applications*, t. 40, *15*, p. 5907-5915, 2013.

[45] D. FAURE, C. NÉDELLEC et C. ROUVEIROL, « Acquisition of Semantic Knowledge using Machine learning methods : The System "ASIUM", » Universite Paris Sud, rapp. tech., 1998.

[46] W. WANG, P. M. BARNAGHI et A. BARGIELA, « Learning SKOS relations for terminological ontologies from text, » IGI Global, 2011, p. 129-152.

[47] M. RANI, A. K. DHAR et O. VYAS, « Semi-automatic terminology ontology learning based on topic modeling, » *Engineering Applications of Artificial Intelligence*, t. 63, p. 108-125, 2017.

[48] G. A. MILLER, R. BECKWITH, C. FELLBAUM, D. GROSS et K. J. MILLER, « Introduction to WordNet : An on-line lexical database, » *International journal of lexicography*, t. 3, *4*, p. 235-244, 1990.

[49] M. A. HEARST, « Automated discovery of WordNet relations, » *WordNet : an electronic lexical database*, t. 2, 1998.

[50] G. SALTON et C. BUCKLEY, « Term-weighting approaches in automatic text retrieval, » *Information processing & management*, t. 24, *5*, p. 513-523, 1988.

[51] S. T. DUMAIS, G. W. FURNAS, T. K. LANDAUER, S. DEERWESTER et R. HARSHMAN, « Using Latent Semantic Analysis To Improve Access To Textual Information, » in *SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS*, ACM, 1988, p. 281-285.

[52] S. C. DEERWESTER, S. T. DUMAIS, T. K. LANDAUER, G. W. FURNAS et R. A. HARSHMAN, « Indexing by Latent Semantic Analysis, » *JASIS*, t. 41, *6*, p. 391-407, 1990. DOI : 10.1002/(SICI)1097-4571(199009)41:6\<391::AID-ASI1\>3.0.CO;2-9. adresse : https://doi.org/10.1002/(SICI)1097-4571(199009)41:6%5C%3C391::AID-ASI1%5C%3E3.0.CO;2-9.

[53] A. NIEKLER et C. KAHMANN, « Extracting process graphs from medical text data, » in *International Semantic Web Conference*, Springer, 2016, p. 76-88.

[54] K. W. CHURCH et P. HANKS, « Word association norms, mutual information, and lexicography, » *Computational linguistics*, t. 16, *1*, p. 22-29, 1990.

[55] K. LINDÉN et J. PIITULAINEN, « Discovering synonyms and other related words, » in *Proceedings of CompuTerm 2004 : 3rd International Workshop on Computational Terminology*, 2004, p. 63-70.

[56] H. N. FOTZO et P. GALLINARI, « Learning" Generalization/Specialization" Relations between Concepts-Application for Automatically Building Thematic Document Hierarchies., » in *RIAO*, Citeseer, 2004, p. 143-155.

[57] E. ZAVITSANOS, G. PALIOURAS, G. A. VOUROS et S. PETRIDIS, « Discovering subsumption hierarchies of ontology concepts from text corpora, » in *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, IEEE, 2007, p. 402-408.

[58] S. BORGO, « Towards Ontology Composition from Cognitive Libraries., » in *JOWO@ IJCAI*, 2015.

[59] Z. S. HARRIS, « Distributional structure, » *Word*, t. 10, *2-3*, p. 146-162, 1954.

[60] V. SHWARTZ, E. SANTUS et D. SCHLECHTWEG, « Hypernyms under siege : Linguistically-motivated artillery for hypernymy detection, » *arXiv preprint arXiv :1612.04460*, 2016.

[61] WIKIPEDIA, *Cluster analysis — Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/w/index.php?title=Cluster%20analysis&oldid=1002271612, [Online ; accessed 07-February-2021], 2021.

[62] R. FELDMAN et J. SANGER, *The Text Mining Handbook, Advanced Approaches in Analyzing Unstructured Data*, 2006. DOI : 10.1017/cbo9780511546914.

[63] C. AGGARWAL et C. ZHAI, « Mining Text Data, » 2012.

[64] C. C. AGGARWAL et C. ZHAI, « A survey of text clustering algorithms, » in *Mining text data*, Springer, 2012, p. 77-128.

[65] D. C. C. AGGARWAL, « Machine Learning for Text, » in *Springer International Publishing*, 2018.

[66] T. MIKOLOV, W.-t. YIH et G. ZWEIG, « Linguistic regularities in continuous space word representations, » in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics : Human language technologies*, 2013, p. 746-751.

[67] T. TOMOKIYO et M. HURST, « A language model approach to keyphrase extraction, » in *Proceedings of the ACL 2003 workshop on Multiword expressions : analysis, acquisition and treatment*, 2003, p. 33-40.

[68] Z. LIU, P. LI, Y. ZHENG et M. SUN, « Clustering to find exemplar terms for keyphrase extraction, » in *Proceedings of the 2009 conference on empirical methods in natural language processing*, 2009, p. 257-266.

[69] Z. DING, Q. ZHANG et X.-J. HUANG, « Keyphrase extraction from online news using binary integer programming, » in *Proceedings of 5th International Joint Conference on Natural Language Processing*, 2011, p. 165-173.

[70] W. X. ZHAO, J. JIANG, J. HE, Y. SONG, P. ACHANAUPARP, E.-P. LIM et X. LI, « Topical keyphrase extraction from twitter, » in *Proceedings of the 49th annual meeting of the association for computational linguistics : Human language technologies*, 2011, p. 379-388.

[71] D. BOURIGAULT, « Surface grammatical analysis for the extraction of terminological noun phrases, » in *COLING 1992 Volume 3 : The 15th International Conference on Computational Linguistics*, 1992.

[72] S. AUBIN et T. HAMON, « Improving term extraction with terminological resources, » in *International Conference on Natural Language Processing (in Finland)*, Springer, 2006, p. 380-387.

[73] M. GRINEVA, M. GRINEV et D. LIZORKIN, « Extracting key terms from noisy and multitheme documents, » in *Proceedings of the 18th international conference on World wide web*, 2009, p. 661-670.

[74] A. V. D. LEAKE, « Jump-starting concept map construction with knowledge extracted from documents, » in *In Proceedings of the Second International Conference on Concept Mapping (CMC*, Citeseer, 2006.

[75] C. Q. NGUYEN et T. T. PHAN, « An ontology-based approach for key phrase extraction, » in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 2009, p. 181-184.

[76] J. TURIAN, L. RATINOV et Y. BENGIO, « Word representations : a simple and general method for semi-supervised learning, » in *Proceedings of the 48th annual meeting of the association for computational linguistics*, 2010, p. 384-394.

[77] M. BARONI, G. DINU et G. KRUSZEWSKI, « Don't count, predict ! a systematic comparison of context-counting vs. context-predicting semantic vectors, » in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, 2014, p. 238-247.

[78] D. D. LEWIS, « An evaluation of phrasal and clustered representations on a text categorization task, » in *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, 1992, p. 37-50.

[79] A. MOSCHITTI et R. BASILI, « Complex linguistic features for text classification : A comprehensive study, » in *European Conference on Information Retrieval*, Springer, 2004, p. 181-196.

[80] Y. YANG, « An evaluation of statistical approaches to text categorization, » *Information retrieval*, t. 1, *1*, p. 69-90, 1999.

[81] R. B. CLARIANA et R. KOUL, « A computer-based approach for translating text into concept map-like representations, » in *Proceedings of the first international conference on concept mapping*, 2004, p. 14-17.

[82] J. PUNURU et J. CHEN, « Learning non-taxonomical semantic relations from domain texts, » *Journal of Intelligent Information Systems*, t. 38, *1*, p. 191-207, 2012.

[83] Y. MATSUO et M. ISHIZUKA, « Keyword extraction from a single document using word co-occurrence statistical information, » *International Journal on Artificial Intelligence Tools*, t. 13, *01*, p. 157-169, 2004.

[84] K. BARKER et N. CORNACCHIA, « Using noun phrase heads to extract document keyphrases, » in *conference of the canadian society for computational studies of intelligence*, Springer, 2000, p. 40-52.

[85] G. SALTON et M. MCGILL, *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1984, ISBN : 0-07-054484-0.

[86] D. D. LEE et H. S. SEUNG, « Learning the parts of objects by non-negative matrix factorization, » *Nature*, t. 401, *6755*, p. 788, 1999.

[87] D. M. BLEI, A. Y. NG et M. I. JORDAN, « Latent Dirichlet Allocation, » *J. Mach. Learn. Res.*, t. 3, p. 993-1022, 2003. adresse : http://jmlr.org/papers/v3/blei03a.html.

[88] Y. BENGIO, R. DUCHARME, P. VINCENT et al., « A Neural Probabilistic Language Model, » in *Journal of Machine Learning Research*, Citeseer, 2000.

[89] T. MIKOLOV, K. CHEN, G. CORRADO et J. DEAN, « Efficient estimation of word representations in vector space, » *arXiv preprint arXiv :1301.3781*, 2013.

[90] T. MIKOLOV, I. SUTSKEVER, K. CHEN, G. CORRADO et J. DEAN, « Distributed representations of words and phrases and their compositionality, » *arXiv preprint arXiv :1310.4546*, 2013.

[91] D. BOLLEGALA, T. MAEHARA et K.-i. KAWARABAYASHI, « Unsupervised cross-domain word representation learning, » *arXiv preprint arXiv :1505.07184*, 2015.

[92] B. MCCANN, J. BRADBURY, C. XIONG et R. SOCHER, « Learned in translation : Contextualized word vectors, » *arXiv preprint arXiv :1708.00107*, 2017.

[93] M. E. PETERS, M. NEUMANN, M. IYYER, M. GARDNER, C. CLARK, K. LEE et L. ZETTLEMOYER, « Deep contextualized word representations, » *arXiv preprint arXiv :1802.05365*, 2018.

[94] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, L. KAISER et I. POLOSUKHIN, « Attention is all you need, » *arXiv preprint arXiv :1706.03762*, 2017.

[95] J. DEVLIN, M.-W. CHANG, K. LEE et K. TOUTANOVA, « Bert : Pre-training of deep bidirectional transformers for language understanding, » *arXiv preprint arXiv :1810.04805*, 2018.

[96] J. A. HARTIGAN et M. A. WONG, « Algorithm AS 136 : A k-means clustering algorithm, » *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, t. 28, *1*, p. 100-108, 1979.

[97] L. KAUFMAN et P. J. ROUSSEEUW, *Finding groups in data : an introduction to cluster analysis*. John Wiley & Sons, 2009, t. 344.

[98] B. J. FREY et D. DUECK, « Clustering by passing messages between data points, » *science*, t. 315, *5814*, p. 972-976, 2007.

[99] M. ESTER, H.-P. KRIEGEL, J. SANDER, X. XU et al., « A density-based algorithm for discovering clusters in large spatial databases with noise., » in *KDD*, t. 96, 1996, p. 226-231.

[100] G. GOVAERT et M. NADIF, « Latent Block Model for Contingency Table, » *Communications in Statistics - Theory and Methods*, t. 39, *3*, p. 416-425, 2010.

[101] S. BASU, A. BANERJEE et R. MOONEY, « Semi-supervised clustering by seeding, » in *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*, Citeseer, 2002.

[102] U. BUATOOM, W. KONGPRAWECHNON et T. THEERAMUNKONG, « Document Clustering Using K-Means with Term Weighting as Similarity-Based Constraints, » *Symmetry*, t. 12, *6*, p. 967, 2020.

[103] ——, « Improving Seeded k-Means Clustering with Deviation-and Entropy-Based Term Weightings, » *IEICE TRANSACTIONS on Information and Systems*, t. 103, *4*, p. 748-758, 2020.

[104] E. SCHUBERT et P. J. ROUSSEEUW, « Faster k-medoids clustering : improving the PAM, CLARA, and CLARANS algorithms, » in *International conference on similarity search and applications*, Springer, 2019, p. 171-187.

[105] M. JHA, « Document clustering using k-medoids, » *arXiv preprint arXiv :1504.01183*, 2015.

[106] P. T. NGUYEN, K. ECKERT, A. RAGONE et T. DI NOIA, « Modification to K-Medoids and CLARA for effective document clustering, » in *International Symposium on Methodologies for Intelligent Systems*, Springer, 2017, p. 481-491.

[107] E. KERSTENS, « Non-Exhaustive, Overlapping k-medoids for Document Clustering, » in *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020.

[108] R. KURADA et D. K. K. PAVAN, « Novel text categorization by amalgamation of augmented k-nearest neighborhood classification and k-medoids clustering, » *arXiv preprint arXiv :1312.2375*, 2013.

[109] Y. LI, S. LI, W. XU et J. GUO, « Analyzing semantic orientation of terms using Affinity Propagation, » in *2012 8th International Symposium on Chinese Spoken Language Processing*, IEEE, 2012, p. 30-34.

[110] I. QASIM, J.-W. JEONG, J.-U. HEU et D.-H. LEE, « Concept map construction from text documents using affinity propagation, » *Journal of Information Science*, t. 39, *6*, p. 719-736, 2013.

[111] Y. CUI, D. LIU, Q. LI, Z. QIU et X. YANG, « DTR : A Novel Topic Generate Algorithm Based on Dbscan and TextRank, » in *The International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, Springer, 2019, p. 425-433.

[112] L. O'CONNOR et S. FEIZI, « Biclustering Using Message Passing, » in *Advances in Neural Information Processing Systems 27*, Z. GHAHRAMANI, M. WELLING, C. CORTES, N. D. LAWRENCE et K. Q. WEINBERGER, éd., Curran Associates, Inc., 2014, p. 3617-3625. adresse : http://papers.nips.cc/paper/5603-biclustering-using-message-passing.pdf.

[113] R. ŁANCUCKI, P. FOSZNER et A. POLANSKI, « Searching through scientific pdf files supported by bi-clustering of key terms matrices, » in *International Conference on Man–Machine Interactions*, Springer, 2017, p. 144-153.

[114] Y. LIU, Z. LI, H. XIONG, X. GAO et J. WU, « Understanding of internal clustering validation measures, » in *2010 IEEE international conference on data mining*, IEEE, 2010, p. 911-916.

[115] P. J. ROUSSEEUW, « Silhouettes : a graphical aid to the interpretation and validation of cluster analysis, » *Journal of computational and applied mathematics*, t. 20, p. 53-65, 1987.

[116] J. C. DUNN, « Well-separated clusters and optimal fuzzy partitions, » *Journal of cybernetics*, t. 4, *1*, p. 95-104, 1974.

[117] E. AMIGÓ, J. GONZALO, J. ARTILES et F. VERDEJO, « A comparison of extrinsic clustering evaluation metrics based on formal constraints, » *Information retrieval*, t. 12, *4*, p. 461-486, 2009.

[118] L. GALÁRRAGA, G. HEITZ, K. MURPHY et F. SUCHANEK, « Canonicalizing Open Knowledge Bases, » *CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management*, p. 1679-1688, nov. 2014. DOI : 10.1145/2661829.2662073.

[119] P. BALDI, S. BRUNAK, Y. CHAUVIN, C. A. ANDERSEN et H. NIELSEN, « Assessing the accuracy of prediction algorithms for classification : an overview, » *Bioinformatics*, t. 16, *5*, p. 412-424, 2000.

[120] J. GORODKIN, « Comparing two K-category assignments by a K-category correlation coefficient, » *Computational biology and chemistry*, t. 28, *5-6*, p. 367-374, 2004.

[121] L. HUBERT et P. ARABIE, « Comparing partitions, » *Journal of classification*, t. 2, *1*, p. 193-218, 1985.

[122] N. X. VINH, J. EPPS et J. BAILEY, « Information theoretic measures for clusterings comparison : Variants, properties, normalization and correction for chance, » *The Journal of Machine Learning Research*, t. 11, p. 2837-2854, 2010.

[123] M. MISURACA, M. SPANO et S. BALBI, « BMS : An improved Dunn index for Document Clustering validation, » *Communications in Statistics-Theory and Methods*, t. 48, *20*, p. 5036-5049, 2019.

[124] W. M. RAND, « Objective criteria for the evaluation of clustering methods, » *Journal of the American Statistical association*, t. 66, *336*, p. 846-850, 1971.

[125] S. WAGNER et D. WAGNER, *Comparing clusterings : an overview*. Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.

[126] ——, *Comparing clusterings : an overview*. Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.

[127] *3.3. Metrics and scoring : quantifying the quality of predictions¶*. adresse : https://scikit-learn.org/stable/modules/model_evaluation.html#matthews-corrcoef.

[128] T. M. COVER et J. A. THOMAS, « Elements of Information Theory John Wiley & Sons, » *New York*, t. 68, p. 69-73, 1991.

[129] F. GANZ, P. BARNAGHI et F. CARREZ, « Automated semantic knowledge acquisition from sensor data, » *IEEE Systems Journal*, t. 10, *3*, p. 1214-1225, 2014.

[130] B. FORTUNA, D. MLADENIČ et M. GROBELNIK, « Semi-automatic construction of topic ontologies, » in *Semantics, Web and Mining*, Springer, 2005, p. 121-131.

[131] T. LOUGE, M. H. KARRAY et B. ARCHIMÈDE, « Investigating a Method for Automatic Construction and Population of Ontologies for Services : Performances and Limitations, » in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, IEEE, 2018, p. 1-6.

[132] B. J. FREY et D. DUECK, « Clustering by passing messages between data points, » *science*, t. 315, *5814*, p. 972-976, 2007.

[133] P. R. TOGATOROP, R. SIAGIAN, Y. NAINGGOLAN et K. SIMANUNGKALIT, « Implementation of ontology-based on Word2Vec and DBSCAN for part-of-speech, » in *Proceedings of the 5th International Conference on Sustainable Information Engineering and Technology*, 2020, p. 51-56.

[134] E. GIANNAKIDOU, V. KOUTSONIKOLA, A. VAKALI et Y. KOMPATSIARIS, « Co-clustering tags and social data sources, » in *2008 The Ninth International Conference on Web-Age Information Management*, IEEE, 2008, p. 317-324.

[135] J. DE KNIJFF, F. FRASINCAR et F. HOGENBOOM, « Domain taxonomy learning from text : The subsumption method versus hierarchical clustering, » *Data & Knowledge Engineering*, t. 83, p. 54-69, 2013.

[136] C. HUANGFU, Y. ZENG et Y. WANG, « Creating Neuroscientific Knowledge Organization System Based on Word Representation and Agglomerative Clustering Algorithm, » *Frontiers in neuroinformatics*, t. 14, p. 38, 2020.

[137] O. OZDIKIS, P. SENKUL et H. OGUZTUZUN, « Semantic expansion of tweet contents for enhanced event detection in twitter, » in *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, IEEE, 2012, p. 20-24.

[138] M. KOSKELA, A. F. SMEATON et J. LAAKSONEN, « Measuring concept similarities in multimedia ontologies : Analysis and evaluations, » *IEEE Transactions on Multimedia*, t. 9, *5*, p. 912-922, 2007.

[139] M. A. SARWAR, M. AHMED, A. HABIB, M. KHALID, M. A. ALI, M. RAZA, S. HUSSAIN et G. AHMED, « Exploiting Ontology Recommendation Using Text Categorization Approach, » *IEEE Access*, 2020.

[140] Z. XU, M. HARZALLAH et F. GUILLET, « Comparing of Term Clustering Frameworks for Modular Ontology Learning, » sér. Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 2 : KEOD, Seville, Spain : SCITEPRESS - Science and Technology Publications, sept. 2018, p. 128-135.

[141] SPACY, *SpaCy :Industrial-strength Natural Language Processing (NLP) with Python and Cython, Explosion AI*, https://github.com/explosion/spaCy, Accessed : 2019-5-10, 2019.

[142] T. ARNOLD, « A Tidy Data Model for Natural Language Processing using cleanNLP, » *The R Journal*, t. 9, *2*, p. 1-20, 2017. adresse : https://journal.r-project.org/archive/2017/RJ-2017-035/index.html.

[143] C. D. MANNING, M. SURDEANU, J. BAUER, J. FINKEL, S. J. BETHARD et D. MCCLOSKY, « The Stanford CoreNLP Natural Language Processing Toolkit, » in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, p. 55-60.

[144] P. GAMALLO et S. BORDAG, « Is singular value decomposition useful for word similarity extraction ? » *Language resources and evaluation*, t. 45, *2*, p. 95-119, 2011.

[145] S. ORAMAS, L. E. ANKE, M. SORDO, H. SAGGION et X. SERRA, « ELMD : An automatically generated entity linking gold standard dataset in the music domain, » in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, p. 3312-3317.

[146] K. KAMRAN, B. DONALD, H. MOJTABA, J. M. KIANA, G. MATTHEW et B. LAURA, *Web of Science Dataset*, Mendeley Data, v6, 2018. adresse : http://dx.doi.org/10.17632/9rw3vkcfy4.6.

[147] M. L. MCHUGH, « Interrater reliability : the kappa statistic, » *Biochemia medica : Biochemia medica*, t. 22, *3*, p. 276-282, 2012.

[148] Z. XU, M. HARZALLAH, F. GUILLET et R. ICHISE, « Towards a Term Clustering Framework for Modular Ontology Learning, » in *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*, Springer, 2018, p. 178-201.

[149] R. ALGHAMDI et K. ALFALQI, « A Survey of Topic Modeling in Text Mining, » *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, t. 6, *1*, 2015, ISSN : 2156-5570. DOI : 10.14569/ijacsa.2015.060121.

[150] T. HOFMANN, « Probabilistic Latent Semantic Indexing, » in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, sér. SIGIR '99, Berkeley, California, USA : Association for Computing Machinery, 1999, p. 50-57, ISBN : 1581130961. DOI : 10.1145/312624.312649. adresse : https://doi.org/10.1145/312624.312649.

[151] X. WANG et A. MCCALLUM, « Topics over time : a non-Markov continuous-time model of topical trends, » in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, p. 424-433.

[152] D. M. BLEI et J. D. LAFFERTY, « Dynamic topic models, » in *Proceedings of the 23rd international conference on Machine learning*, 2006, p. 113-120.

[153] R. M. NALLAPATI, S. DITMORE, J. D. LAFFERTY et K. UNG, « Multiscale topic tomography, » in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, p. 520-529.

[154] T.-H. CHEN, S. W. THOMAS et A. E. HASSAN, « A survey on the use of topic models when mining software repositories, » *Empirical Software Engineering*, t. 21, *5*, p. 1843-1919, 2016.

[155] I. T. JOLLIFFE, « Principal components in regression analysis, » in *Principal component analysis*, Springer, 1986, p. 129-155.

[156] WIKIPEDIA, *Tf–idf — Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/w/index.php?title=Tf%E2%80%93idf&oldid=1008400463, [Online ; accessed 05-March-2021], 2021.

[157] K. NIGAM, A. K. MCCALLUM, S. THRUN et T. MITCHELL, « Text Classification from Labeled and Unlabeled Documents using EM, » in *MACHINE LEARNING*, 1999, p. 103-134.

[158] D. M. ANDRZEJEWSKI, M. CRAVEN et X. ZHU, « Incorporating domain knowledge in latent topic models, » thèse de doct., University of Wisconsin–Madison, 2010.

[159] T. L. GRIFFITHS et M. STEYVERS, « Finding scientific topics, » *Proceedings of the National academy of Sciences*, t. 101, *suppl 1*, p. 5228-5235, 2004.

[160] A. ASUNCION, M. WELLING, P. SMYTH et Y. W. TEH, « On Smoothing and Inference for Topic Models, » in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, sér. UAI '09, Montreal, Quebec, Canada : AUAI Press, 2009, p. 27-34, ISBN : 9780974903958.

[161] Y. HU, J. BOYD-GRABER, B. SATINOFF et A. SMITH, « Interactive topic modeling, » *Machine learning*, t. 95, *3*, p. 423-469, 2014.

[162] D. ANDRZEJEWSKI, X. ZHU et M. CRAVEN, « Incorporating domain knowledge into topic modeling via Dirichlet forest priors, » in *Proceedings of the 26th annual international conference on machine learning*, 2009, p. 25-32.

[163] S. BASU, I. DAVIDSON et K. WAGSTAFF, *Constrained clustering : Advances in algorithms, theory, and applications*. CRC Press, 2008.

[164] J. D. MCAULIFFE et D. M. BLEI, « Supervised topic models, » in *Advances in neural information processing systems*, 2008, p. 121-128.

[165] S. LACOSTE-JULIEN, F. SHA et M. I. JORDAN, « DiscLDA : Discriminative learning for dimensionality reduction and classification, » in *Advances in neural information processing systems*, 2009, p. 897-904.

[166] M. THELEN et E. RILOFF, « A bootstrapping method for learning semantic lexicons using extraction pattern contexts, » in *Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002)*, 2002, p. 214-221.

[167] L. LI, B. ROTH et C. SPORLEDER, « Topic models for word sense disambiguation and token-based idiom detection, » in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, p. 1138-1147.

[168] D. RAMAGE, D. HALL, R. NALLAPATI et C. D. MANNING, « Labeled LDA : A supervised topic model for credit attribution in multi-labeled corpora, » in *Proceedings of the 2009 conference on empirical methods in natural language processing*, 2009, p. 248-256.

[169] B. V. BARDE et A. M. BAINWAD, « An overview of topic modeling methods and tools, » in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2017, p. 745-750.

[170] J. BOYD-GRABER, D. BLEI et X. ZHU, « A topic model for word sense disambiguation, » in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007, p. 1024-1033.

[171] D. KIM, H. WANG et A. OH, « Context-dependent Conceptualization, » sér. IJCAI '13, Beijing, China : AAAI Press, 2013, p. 2654-2661.

[172] J. CHENG, Z. WANG, J.-R. WEN, J. YAN et Z. CHEN, « Contextual text understanding in distributional semantic space, » in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, p. 133-142.

[173] C. CHEMUDUGUNTA, A. HOLLOWAY, P. SMYTH et M. STEYVERS, « Modeling documents by combining semantic concepts with unsupervised statistical learning, » in *International Semantic Web Conference*, Springer, 2008, p. 229-244.

[174] J. ZHOU, Y. FAN et J. ZHANG, « Generating Knowledge Maps for Songs and Users in Music Market with Probabilistic Topic Model, » in *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, IEEE, 2019, p. 83-92.

[175] Y. LIU, Z. LIU, T.-S. CHUA et M. SUN, « Topical word embeddings, » in *Proceedings of the AAAI Conference on Artificial Intelligence*, t. 29, 2015.

[176] W. WU, H. LI, H. WANG et K. Q. ZHU, « Probase : A probabilistic taxonomy for text understanding, » in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 2012, p. 481-492.

[177] L. NIU, X. DAI, J. ZHANG et J. CHEN, « Topic2Vec : learning distributed representations of topics, » in *2015 International conference on asian language processing (IALP)*, IEEE, 2015, p. 193-196.

[178] B. SHI, W. LAM, S. JAMEEL, S. SCHOCKAERT et K. P. LAI, « Jointly learning word embeddings and latent topics, » in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, p. 375-384.

[179] R. S. RANDHAWA, P. JAIN et G. MADAN, « Topic modeling using distributed word embeddings, » *arXiv preprint arXiv :1603.04747*, 2016.

[180] D. NEWMAN, J. H. LAU, K. GRIESER et T. BALDWIN, « Automatic evaluation of topic coherence, » in *Human language technologies : The 2010 annual conference of the North American chapter of the association for computational linguistics*, 2010, p. 100-108.

[181] M. RÖDER, A. BOTH et A. HINNEBURG, « Exploring the space of topic coherence measures, » in *Proceedings of the eighth ACM international conference on Web search and data mining*, 2015, p. 399-408.

[182] J. CHANG, S. GERRISH, C. WANG, J. BOYD-GRABER et D. BLEI, « Reading tea leaves : How humans interpret topic models, » *Advances in neural information processing systems*, t. 22, p. 288-296, 2009.

[183] D. MIMNO, H. WALLACH, E. TALLEY, M. LEENDERS et A. MCCALLUM, « Optimizing semantic coherence in topic models, » in *Proceedings of the 2011 conference on empirical methods in natural language processing*, 2011, p. 262-272.

[184] P. BOJANOWSKI, E. GRAVE, A. JOULIN et T. MIKOLOV, « Enriching word vectors with subword information, » *Transactions of the Association for Computational Linguistics*, t. 5, p. 135-146, 2017.

[185] M. BELFORD et D. GREENE, « Comparison of Embedding Techniques for Topic Modeling Coherence Measures., » in *LDK (Posters)*, 2019, p. 1-5.

[186] C. SIEVERT et K. SHIRLEY, « LDAvis A method for visualizing and interpreting topics, » in *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, 2014, p. 63-70.

[187] J. CHUANG, C. D. MANNING et J. HEER, « Termite Visualization techniques for assessing textual topic models, » in *In Proceedings of the International Working Conference on Advanced Visual Interfaces*, ACM, 2012, p. 74-77.

[188] Z. XU, M. HARZALLAH, F. GUILLET et R. ICHISE, « Modular Ontology Learning with Topic Modelling over Core Ontology, » *Procedia Computer Science*, t. 159, p. 562-571, jan. 2019. DOI : 10.1016/j.procs.2019.09.211.

[189] M. D'AQUIN, A. SCHLICHT, H. STUCKENSCHMIDT et M. SABOU, « Criteria and Evaluation for Ontology Modularization Techniques, » *Modular ontologies*, t. 5445, p. 67-89, jan. 2009.

[190] J. N. LEVI, *The syntax and semantics of complex nominals*. Academic Press New York, 1978.

[191] M. LIBERMAN et R. SPROAT, « The stress and structure of modified noun phrases in English, » *Lexical matters*, p. 131-181, 1992.

[192] MEDIAWIKI, *API :Opensearch — MediaWiki, The Free Wiki Engine*, [Online ; accessed 23-May-2020], 2020. adresse : https://www.mediawiki.org/w/index.php?title=API:Opensearch&oldid=3586276.

[193] S. AUER, C. BIZER, G. KOBILAROV, J. LEHMANN, Z. IVES et et AL., « DBpedia : A Nucleus for a Web of Open Data, » in *PROC. 6TH INT'L SEMANTIC WEB CONF*, Springer, 2007.

[194] D. VRANDEČIĆ et M. KRÖTZSCH, « Wikidata : A Free Collaborative Knowledgebase, » *Commun. ACM*, t. 57, *10*, p. 78-85, sept. 2014, ISSN : 0001-0782.

[195] H. REICHENBACH, *Elements of Symbolic Logic*. London : Dover Publications, 1947.

[196] M. HOFFMAN, F. R. BACH et D. M. BLEI, « Online learning for latent dirichlet allocation, » in *advances in neural information processing systems*, 2010, p. 856-864.

UNIVERSITE BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE NANTES

Titre : Améliorer LDA pour L'apprentissage d'ontologie

Mot clés : ontology learning, LDA, term clustering, prior knowledge embedding

Resumé : Cette thèse vise à tirer profit du modèle sémantique LDA pour améliorer la conceptualisation des termes en vue de l'apprentissage d'ontologie à partir de textes, où des termes similaires sont regroupés en fonction de concepts de base prédéfinis. Nous avons exploré le cadre classique du regroupement de termes et étudié l'impact des techniques de représentation des termes. Nous avons proposé des stratégies de regroupement de termes (term clustering) basées sur LDA, où des connaissances préalables sont utilisées pour semi-superviser LDA. De plus, nous avons construit la structure taxonomique de l'ontologie, en appliquant en interne les cadres de sous-catégorisation sur les phrases nominatives et en bénéficiant en externe des bases de connaissances. Notre stratégie de regroupement basée sur LDA a été plus performante que la majorité des travaux de regroupement dans le cadre classique. Notre approche optimale d'intégration des connaissances préalables a dépassé les performances de LDA de base et de seeded LDA. Le regroupement basé sur LDA pourrait contribuer à améliorer la formation des concepts à partir de termes pour l'apprentissage d'ontologie.

Title : Enhancing LDA for Ontology Learning

Keywords : ontology learning, LDA, term clustering, prior knowledge embedding

Abstract : This dissertation aims to enhance LDA's utilities of conceptualizing terms towards ontology learning, where similar terms are clustered to the predefined core concepts. We explored the classic workflow of term clustering and studied the clustering impacts of the terms representation techniques. Comparatively, we proposed the LDA based clustering strategy, where the prior knowledge embedding techniques are applied to semi-supervise the LDA for the more satisfying clusters. In addition, we built up the taxonomic structure of the ontology, by internally applying the subcategorization frames over noun phrases and externally benefitting from the knowledge bases. The experiment results showed that our proposed LDA based clustering strategy outperformed the majority of the clustering works in the classic workflow. Our optimal prior knowledge embedding approach exceeded the performance of basic LDA and Seeded LDA but dropped behind the Z-label LDA. This dissertation suggests that the LDA based clustering strategy could contribute to the anticipating term conceptualizations for ontology learning.