Research on the Automatic Construction of the Resource Space Model for Scientific Literature

Lei He

Doctor of Philosophy

March 2019

Dedication

To my parents

Aston University

Research on the Automatic Construction of the Resource Space Model for Scientific Literature

Lei He

Doctor of Philosophy

March 2019

Abstract

The resource space model is a semantic data model to organize Web resources based on a classification of resources. The scientific resource space is an application of the resource space model on massive scientific literature resources. The construction of a scientific resource space needs to build a category (or concept) hierarchy and classify resources. Manual design suffers from heavy workload and low efficiency. In this thesis, we propose novel methods to solve the following two problems in the construction of a scientific resource space:

1. Automatic maintenance of a category hierarchy. A category hierarchy needs to evolve dynamically with new resources continually arriving so as to satisfy the dynamic requirements of the organization and management of resources. We propose an automatic maintenance approach to modifying the category hierarchy according to the hierarchical clustering of resources and show the effectiveness of this method by a series of comparison experiments on multiple datasets.

2. Automatic construction of a concept hierarchy. We propose a joint extraction model based on a deep neural network to extract entities and relations from scientific articles and build a concept hierarchy. Experimental results show the effectiveness of the joint model on the Semeval 2017 Task 10 dataset.

We also implement a prototype system of the scientific resource space. The prototype system enables the comparative summarization on scientific articles. A set of novel comparative summarization methods based on the differential topic models (dTM) are proposed in this thesis. The effectiveness of the dTM-based methods is shown by a series of experimental results.

Keywords: Resource Space Model, Scientific Literature, Category Hierarchy, Entity, Relation

Acknowledgements

First of all, I would like to express my great gratitude to my supervisor Prof. Hai Zhuge for giving me valuable guidance and strength to support me during my PhD study. His insightful ideas and all-enduring knowledge sharing always inspire me to keep improving my work. Nothing in this short space can express my appreciation and gratitude to him.

I am grateful to all members of the computer science group, in particular Dr. Xiaorui Jiang, Dr. Weiren Yu, Dr Yulan He who gave me inspirations to form the initial idea of this work. I am also thankful to the research and administrative staff at the School of Engineering and Applied Sciences for their advice and support during my PhD study. Many thanks to Sandra Mosley, Kanchan Patel and Helen Yard.

Moreover, I would like to gratefully thank all my friends and colleagues in and outside Aston University. Without their encouragement and support, my PhD research work could not have been accomplished in the present form. I regret that I cannot mention all their names here. I am very thankful to my friends Dr. Xi He, Wei Li, Mengyun Cao for their friendly company with endless support. Hearty thanks to Lin Gui, Fan Wang, Nick Powell, Vishwash Batra and Gabriele Pergola for the enjoyable time we had together.

Finally, I am grateful to my family: my parents for their endless love and support, grandparents and our extended family members for their enormous care and close attention in my life.

Thanks to all of you

Contents

List of Tables

List of Figures

List of Abbreviations

RSM	Resource Space Model
AI	Artificial Intelligence
NLP	Natural Language Processing
dTM	Differential Topic Model
XML	eXtensible Markup Language
DBLP	Digital Bibliography Library Project
ODP	Open Directory Project
DMOZ	directory.mozilla.org
AMHC	Automatic Maintenance of Hierarchical Category
LDA	Latent Dirichlet Allocation
UC_Score	Uncertain Score
HAC	Hierarchical Agglomerative Clustering
SL	Single-Link Function
CL	Complete-Link Function
AL	Average-Link Function
CE	Centroid Function
HPC	Hierarchical Partitional Clustering
hLDA	Hierarchical Latent Dirichlet Allocation
nCRP	Nested Chinese Restaurant Process
nHDP	Nested Hierarchical Dirichlet Processes
RST	Rhetorical Structure Theory
AZ	Argumentative Zoning

CoreSC	Core Scientific Concept
CRF	Conditional Random Field
SVM	Support Vector Machine
HMM	Hidden Markov Model
MEMM	Maximum Entropy Markov Model
POS	Part-Of-Speech Tagging
NER	Named Entity Recognition
SRL	Semantic Role Labelling
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long-short Term Memory Network
OOV	Out-Of-Vocabulary
MV-RNN	Matrix-Vector Recursive Neural Network
ILP	Integer Linear Programming
DIH	Distributional Inclusion Hypothesis
JER	Joint Entity and Relation Extraction
DAG	Directed Acyclic Graph
Hypernymy RE	Hypernymy Relation Extraction
MMR	Maximal Marginal Relevance
cc-LDA	Cross-Collection LDA
TAM	Topic-Aspect Model
SAGE	Sparse Additive Generative Model
Dir	Dirichlet Distribution
PMI	Pointwise Mutual Information
CDS	Complementary Dominating Set
MAP	Maximum A Posterior

Declaration

The thesis entitled *Research on the Automatic Construction of the Resource Space Model for Scientific Literature* submitted for the degree of PhD in Computer Science, has been composed by myself and has not been presented or accepted in any previous application for a degree. All sources of information have been acknowledged by means of references.

Portions of the work presented in this thesis have been published in the following journal and conferences.

1. Hai Zhuge, Lei He. Automatic maintenance of category hierarchy. *Future Generation Computer System.* 67: 1-12 (2017).

2. Lei He, Wei Li, Hai Zhuge. Exploring Differential Topic Models for Comparative Summarization of Scientific Papers. *COLING 2016*: 1028-1038.

3. Wei Li, Lei He, Hai Zhuge. Abstractive News Summarization based on Event Semantic Link Network. *COLING 2016*: 236-246.

Lei He

06 September 2019

1 Introduction

1.1 Background

With the rapid development of science and information technology, the number of scientific literature resources has been increasing exponentially. According to the STM (Scientific, Technical and Medical) 2015 report[1] (Ware & Mabe, 2015), there were about 28,100 scholarly peer-reviewed English journals and 6450 non-English journals in 2014 publishing around 2.5 million articles a year, which means a rate higher than one new article every 13 seconds. Figure 1.1 shows the number of scientific articles indexed by the world famous scientific databases in March 2015[2], among which Google Scholar indexed between 100 and 160 million documents including journal articles, books and grey literature, etc. Web of Science indexed 90 million articles and CrossRef database indexed 80 million digital objects including 58 million journal articles.

Figure 1.1: The number of indexed scientific literatures on scientific retrieval platforms.

In face of the explosive growth of scientific resources, the lack of effective resource organization and management methods greatly reduces the efficiency of scientific information acquisition. Currently, scientific resources are mainly organized in two ways: by metadata and by keywords. Most academic websites like DBLP organize scientific articles by metadata, such as year, author and publication, which provide no content information of articles. Other academic search engines like Google Scholar mainly organize scientific resources based on keywords, however, they can not provide complete and fine-grained semantic information such as task, process and mate-

[1] https://www.stm-assoc.org/2015_02_20_STM_Report_2015.pdf

[2] The data comes from the STM 2015 report.

rial[3]. These methods make it difficult for researchers to accurately accquire schientific resources of interest and force people to read more to extract useful information.

A UK study surveys the number of articles read by university faculty per year (Tenopir, Mays, & Wu, 2011). It reveals that the number increased steadily between 1977 and 1998 from 150 to 188 at an annual growth rate of 2%. After 2000, the number increased substantially from 188 in 1998 to 271 in 2006 and the annual growth rate increased to 8.5%. The STM report shows that the average time people spend on a single article remained at 45-50 minutes between 1977 and 1998, but dropped to just 30 minutes in 2006. The increase of reading has changed the way people read. It makes skipping reading and horizontal reading become a habitual way of information seeking and reading. In 2012, the CIBER (Consultants in Business Engineering Research) group found by analyzing publishers' log files that most users of scholarly websites only browse 1-3 pages in short session time (Nicholas & Clark, 2012). Researchers read quickly from one article to another rather than in-depth reading, so as to get more useful information from massive scientific resources. The change of the reading style has put forward new requirements on organization and management of scientific resources.

To increase the efficiency of the acquisition and utilization of scientific resources, it is necessary to organize them by both general metadata information and fine-grained semantic information in a united semeantic data model. Researchers can accurately and efficiently acquire targeted information with the model. For example, researcheres can directly locate an article with a certain task, process or material in a particular year. However, at present, there are no effective models in which both metadata and fine-grained semantic content are represented uniformly. In this thesis, we exploit the resource space model to organize scientific literature resources and simultaneously provide metadata descriptions and fine-grained semantic content descriptions.

1.2 The Resource Space Model

The Resource Space Model (*RSM*) is a semantic data model that coordinates multiple classification hierarchies to form a hierarchical classification space for specifying, storing, managing and retrieving various resources (Zhuge, 2004, 2007). Classification is a basic method for human to organize things and perceive the world. Resources can be classified in different perspectives. If we regard one classification perspective as a dimension, a multi-dimensional classification space can be formed where each dimension (axis) is defined by a set of coordinates, either flat or hierarchical, which are representing categories of resources.

Definition 1. Resource Space: *A resource space is a multi-dimensional space denoted as* $RS(X_0, X_1, ..., X_{n-1})$ *in which* X_i *is an axis consisting of a set of coordinates that can be flat or*

[3]TASK, PROCESS and MATERIAL are defined as three basic elements of scientific articles in SemEval 2017 Task 10 (Augenstein, Das, Riedel, Vikraman, & McCallum, 2017).

tree-structured. Every coordinate represents a category of resources. Each point in the resource space uniquely determines a relevant resource set (maybe an empty set). A resource space has a set of necessary attributes name, type, location, access privilege.

In an n-dimensional resource space, axes and coordinates together reflect the classification semantics of resources and constitute a resource space. The location of a resource depends on its category information. Resources classified into a same category are located in the same point sharing coordinates in a resource space. Users can query and locate resources by providing coordinates in a resource space.

Figure 1.2 shows an example of a 3-dimensional scientific resource space that organizing scientific literature resources in three perspectives, Year = {2000, $\cdots$, 2010}, Type = {Book, Thesis, In-collection, In-proceeding} and Author = {A, $\cdots$, Z}. Each point defines a class of scientific resources. For example, the point (2000, Book, A) represents all the books whose author name starts with A and published in the year of 2000. A category hierarchy can be defined with tree-structured coordinates on axes. For example, the coordinate "Thesis" on axis "Type" is classified into PhD. thesis and master thesis.

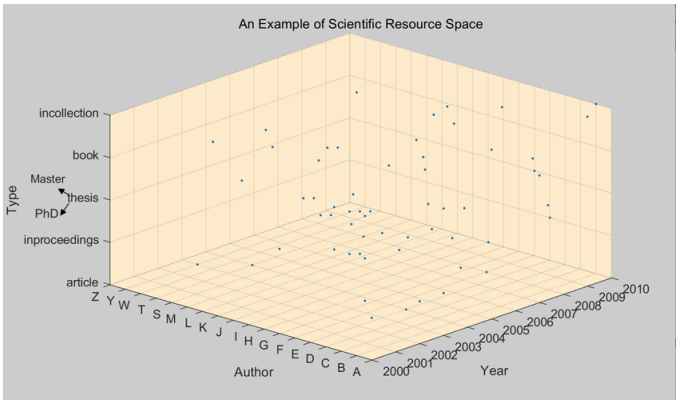


Figure 1.2: A 3-dimensional scientific resource space example.

The multi-dimensional hierarchical structure of a resource space supports multi-facet browsing, generalization and specialization on resources, which is designed to better satisfy the requirements of information acquisition.

Definition 2. Resource Space Schema: *A resource space schema is a five-tuple* $\{RS, A, C, S, dom\}$ *that defines the structure of a resource space:*

1. *RS is the name of a resource space;*
2. $A = \{X_i | 1 \leq i \leq n\}$ *is the set of axes;*
3. $C = \{C_{ij} | C_{ij} \in X_i, 1 \leq i \leq n\}$ *is the set of coordinates;*
4. *S is the power set of a resource set;*

5. *dom is a function mapping from the axis set A and the coordinate set C to S, defined as $A*C \rightarrow S$, that is, for any axis $X_i = \{C_{i1}, C_{i2}, ..., C_{ip}\}$ and coordinate C_{ij}, $dom(X_i, C_{ij}) = V_{ij}$, $V_{ij} \in S$, where $1 \leq i \leq n$, $1 \leq j \leq p$.*

As S and *dom* are fixed at the construction of a resource space, the resource space schema can be simplified as a three-tuple $\{RS, A, C\}$. The resource space schema is static but a resource space should evolve dynamically with new resources arriving so as to satisfy the dynamic requirements of the organization of resources.

To express the hierarchical relations between concepts in a resource space and achieve more efficient resource operations (insert, deletion and query, etc.), the RSM Schema Tree has been proposed to represent the resource space schema (Zhuge, 2007). An n-dimensional hierarchical resource space has the following resource space schema:

$$RS(X_0, X_1, ..., X_{n-1}) \tag{1.1}$$

$$X_0(C_{0,0}, C_{0,1}, ..., C_{0,i}) \tag{1.2}$$

$$... \tag{1.3}$$

$$X_{n-1}(C_{n-1,0}, C_{n-1,1}, ..., C_{n-1,j}). \tag{1.4}$$

The entire resource space can be represented as a tree and each dimension as a subtree. Thus an axis can be regarded as a 1-dimensional resource space.

Users operate resource space to manipulate resources. A set of normal forms are proposed to ensure the correctness of operations in a resource space (Zhuge, 2004, 2007). The first normal form (*1NF*) is to avoid redundancy. *1NF* guarantees that there are no duplicated axes and no duplicated coordinates on any axis in a resource space. The second normal form (*2NF*) resource space is a *1NF* resource space where coordinates on any axis are independent of each other, i.e., any two coordinates have no intersections with each other. The *2NF* avoids the semantic dependency between coordinates in a resource space and guarantees a fine classification system that enables a resource space to locate resources accurately. To clarify the third normal form (*3NF*), we first define two concepts *coordinate partition* and *axis partition*.

Definition 3. Coordinate Partition: *Given an axis $X = \{C_1, C_2, ..., C_n\}$ and C_i is a coordinate on another axis X', X forms a coordinate partition on C_i denoted as C_i/X, if and only if (1) $(R(C_j) \cap R(C_i)) \cap (R(C_k) \cap R(C_i)) = \phi, 1 \leq j < k \leq n$(2) $(R(C_1) \cap R(C_i)) \cup (R(C_2) \cap R(C_i)) \cup ... \cup (R(C_n) \cap R(C_i)) = R(C_i)$. $R(C)$ represents a class of resources defined by coordinate C.*

The coordinate partition classifies $R(C_i)$ into n classes: $R(C_i/X) = \{R(C_1) \cap R(C_i), R(C_2) \cap R(C_i), ..., R(C_n) \cap R(C_i)\}$.

Definition 4. Axis Partition: *Given two axes $X = \{C_1, C_2, ..., C_n\}$ and $X' = \{C'_1, C'_2, ..., C'_m\}$, X forms an axis partition on X' denoted as X'/X, if and only if X forms a coordinate partition on each coordinate on X'.*

Given two axes X and X', if X'/X and X/X', we say that X and X' are orthogonal to each other.

According to the above definitions, we can define a third normal form (*3NF*) resource space. The *3NF* resource space defines on the basis of a *2NF* resource space where any two axes are orthogonal to each other. The *3NF* ensures that any point can uniquely determine a class of resources. If two axes X and X' are orthogonal, they represent the same set of resources, i.e., $R(X) = R(X')$.

1.3 The Scientific Resource Space

1.3.1 Property Division of Scientific Literature Resources

The scientific resource space is an application of the resource space model. The multi-dimensional hierarchical structure of a resource space naturally supports multi-facet resource browsing and hierarchical query. The hierarchical coordinate system in a resource space enables different levels of abstraction on properties of scientific resources. According to the characteristics of the scientific resources, this thesis divides the resource properties into extrinsic properties and intrinsic properties.

The extrinsic properties provide the coarse-grained semantic descriptions for scientific articles based on the metadata, such as Year, Author, Publication and Category, which help to label and distinguish resources one from another. The extrinsic properties provide no descriptions related to the contents of scientific articles. Thus the extrinsic properties can be obtained by parsing metadata files without analysing contents of articles. Metadata are mostly saved in a well-structured XML format, such as the metadata files from ScienceDirect and DBLP.

The intrinsic properties provide the fine-grained semantic descriptions for scientific articles based on the contents, which can be obtained by extracting keyword entities using content analysis techniques such as lexical analysis, syntactic analysis and entity recognition. SemEval 2017 Task 10 proposed a new task of extracting keyphrases and relations from scientific papers and defined three basic types of entities: TASK, PROCESS and MATERIAL (Augenstein et al., 2017). TASK defines a class of entities that describe the research problem a paper trying to address. PROCESS defines entities that describe methods or equipment a paper studies or utilizes. MATERIAL defines entities that describe corpora or physical materials in a scientific paper. The three types of entities describe key content of a scientific paper.

This thesis regards the three types of entities as the intrinsic properties of scientific resources. Thus the intrinsic properties can be extracted from articles by recognizing different types of entities, which provides fine-grained descriptions of the body content and enables a larger variety of content retrieval for scientific literature resources. For example, it can help users to retrieve papers that study method X to solve task Y and use dataset Z in the experiments or retrieve

papers that utilize a variant of X to solve task Y.

The combination of the two types of properties can provide a comprehensive description for scientific literature resources. According to the property division, we classify the dimensions in a scientific resource space into macro dimensions and micro dimensions. Extrinsic properties constitute macro dimensions, while intrinsic properties comprise micro dimensions.

1.3.2 Dimension Division and Construction in the Scientific Resource Space

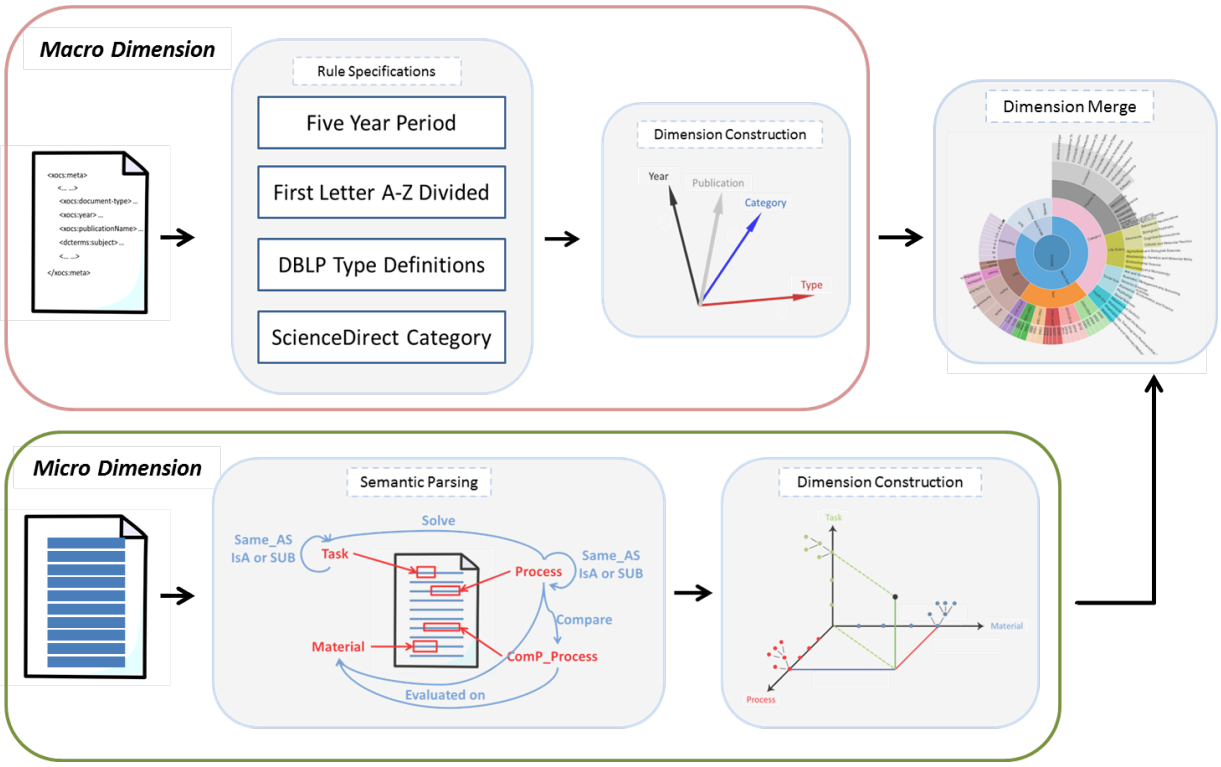


Figure 1.3: The overall framework of construction of scientific resource space.

The dimensions of the resource space model correspond to the properties of resources. According to the property division of scientific resources, the dimensions in a scientific resource space can be classified into macro dimensions and micro dimensions. Specifically, macro dimensions correspond to extrinsic properties, while micro dimensions correspond to intrinsic properties of scientific resources. This thesis constructs a scientific resource space with two types of dimensions to support browsing, retrieval and summarization services on scientific literature. The construction of macro dimensions requires parsing metadata files and the construction of micro dimensions needs content analysis on scientific articles. Figure 1.3 shows the overall framework of building a scientific resource space. The construction of macro dimensions and micro dimensions will be described in the following two subsections.

1.3.2.1 The Construction of Macro Dimensions

The construction of macro dimensions requires the parsing of metadata description files to extract extrinsic properties of scientific resources. ScienceDirect provides access to the world's

leading multidisciplinary online index for full-text scientific journal articles, including over 12 million articles from 3500 academic journals and 34,000 e-books. The articles are grouped into four main subject areas: Physical Sciences and Engineering, Life Sciences, Health Sciences, and Social Sciences and Humanities. The *D*igital *B*ibliography *L*ibrary *P*roject (*DBLP*) hosting more than 3.66 million scientific articles and other types of publications provides access to the online reference for open bibliographic information on major computer science journals and proceedings. Unlike other digital libraries which use relational databases, ScienceDirect and DBLP utilize XML files to store metadata of scientific literature resources. Both of them provide metadata description files in well-structured XML formats.

For the analysis of the metadata files in ScienceDirect and DBLP, we choose four extrinsic properties to build macro dimensions in a scientific resource space: Year, Publication, Type and Category.

Table 1.1 lists the descriptions and the construction rules for each of the macro dimensions. The Year dimension describes the publication time of scientific resources and divides 1990-2018 into six time periods of five years. The dimension of Publication groups scientific resources according to the names of academic journals. The Type dimension defines five types of scientific resources inspired by the types of publication records in DBLP, including (1) article: an article from a journal or magazine; (2) inproceedings: a paper from a conference or workshop; (3) book: an authored monograph or an edited collection of articles; (4) incollection: a chapter in a book; and (5) thesis: a PhD thesis or a Master thesis. The Category dimension defines the subject of scientific articles according to the ScienceDirect category hierarchy, which contains 4 top-level categories, 24 second-level categories and 238 third-level categories.

Table 1.1: Description of macro-dimensions and construction rules.

Macro Dimensions	Description	Construction Rules
Year	publication time of resources	1990-2018, five year in a time period
Publication	publication name	the first letter divided into A-Z
Type	type of resources	the DBLP type definition
Category	subject category of resources	the ScienceDirect subject taxonomy

When constructing macro dimensions in a scientific resource space, a major problem is that the existing category hierarchies can be too general to provide classification for some branch subjects, thus making it impossible to organize scientific resources in specific areas. For example, Artificial Intelligence (*AI*) is a leaf category in the 3-level ScienceDirect category hierarchy. However, AI covers a large variety of branch subjects in the field of computer science, including natural language processing, machine learning, knowledge representation, robotics, and so on. Directly applying ScienceDirect category hierarchy to organizing the scientific resources in the field of AI will result in the failure of producing these branch subject categories.

The category hierarchy in a resource space needs to evolve dynamically with resources con-

tinually residing in so as to adapt to the requirements of dynamic organization and management of resources. The category hierarchy may change because some new categories emerge or some existing categories need to be merged, which results in classfying resources into inappropriate categories during classification and generating less cohesive categories. A poor taxonomy will have a negative impact on the classification performance and result in the failure of locating resources accurately. Thus, how to adjust an existing category hierarchy to make it adapt to dynamically organizing resources in specific areas is one of the major research problems in this thesis.

The task of automatically constructing a resource space includes two aspects: (1) constructing dimensions of a resource space, and (2) inserting resources into a resource space.

After constructing the macro dimensions, resources need to be inserted into a scientific resource space by parsing metadata description files to extract extrinsic property values for each scientific article and make associations between property values and coordinates on macro dimensions.

1.3.2.2 The Construction of Micro Dimensions

The construction of micro dimensions relies on extracting the intrinsic properties of scientific resources. However, compared to the extrinsic properties that are explicitly expressed in metadata files, it is more difficult to extract the intrinsic properties which are usually hidden in unstructured scientific documents. This thesis refers to Semeval 2017 Task 10 which defines TASK, PROCESS and MATERIAL as three basic entity types in scientific articles (Augenstein et al., 2017). We regard these three fundamental types as the intrinsic properties of scientific resources. An entity represents one particular entity type, whereas an entity instance is a specific mention of an entity in scientific documents. In order to facilitate the narration, entity instances are called entities in the rest of this thesis.

According to the intrinsic properties of scientific resources, this thesis builds three micro dimensions in a scientific resource space: Task, Process and Material. Keyphrases describing research problems, such as Summarization and Sentiment Analysis, are used as coordinates on the Task dimension. Keyphrases describing methods or equipment, such as Topic Models and Integer Linear Programming, are used as coordinates on the Process dimension. Keyphrases describing corpora or physical materials, such as Twitter Data and online reviews, are used as coordinates on the Material dimension.

Constructing the hierarchical coordinate system on each micro dimension means building the concept hierarchy, which needs to recognize entities and extract Hyponym-of and Synonym-of relations through content analysis techniques, such as lexical analysis, syntactic analysis and entity recognition. Thus how to recognize the three types of entities and extract the relations (Hyponym-of and Synonym-of) between entities with the same entity types to construct micro

dimensions is a major research problem in this thesis.

Finally, macro dimensions and micro dimensions are merged to generate a complete scientific resource space. Figure 1.4 is a visualization of a high-dimensional scientific resource space. The hierarchical coordinate system is laid out radially, with the top of the hierarchy at the center and deeper levels farther away from the center. In the figure, *SciRSM* in the center represents the whole resource space which consists of "Macro Dimensions" and "Micro Dimensions". The "Macro Dimension" can be further unfolded into four dimensions: "Year", "Type", "Publication" and "Category", and the "Micro Dimension" can be further unfolded into three dimensions: "Task", "Process" and "Material". Each coordinate in the figure can be unfolded into sub-level coordinates recursively.

Figure 1.4: Dimensions of scientific resource space

1.4 Summary of Research Problems

This thesis mainly concerns itself with the automatic construction of a scientific resource space, which includes the construction of two types of dimensions. One type is the set of macro-

dimensions, which are based on the metadata of scientific articles, and the other type is the set of micro-dimensions, which are based on the contents of scientific articles. In the construction of the macro dimensions, this thesis studies the automatic maintenance of the category hierarchy and proposes an approach to modifying the category hierarchy so as to satisfy the requirements of dynamically organizing and managing scientific resources. In the construction of the micro dimensions, this thesis studies the automatic construction of the concept hierarchy, which consists of recognizing entities and extracting Hyponym-of and Synonym-of relations between entities from scientific articles.

A prototype system based on the scientific resource space is implemented to support browsing, retrieval and summarization services applied to the scientific literature. This thesis also proposes novel scientific summarization based on the concept of coordinate partition in the resource space model (referred to section 1.2), which is an application of the scientific resource space helping users retrieving and utilizing scientific resources.

1.4.1 Automatic Maintenance of the Category Hierarchy in Macro Dimensions

The category hierarchy plays an important role in a resource space and it should not only be consistent with the existing domain knowledge but also be appropriate to the existing resources available. The category hierarchies which have been manually created could better satisfy the domain knowledge than automatically generated category hierarchies. However, the following two problems will arise when organizing resources with existing category hierarchies:

1. The existing category hierarchy can be too general to provide detailed classification for some branch subjects, thus making it impossible to organize resources in specific areas.

2. The existing category hierarchy may change, for example, some new categories emerge. This will result in classifying resources into less relevant categories and destroying the cohesion of categories.

Thus the unpredictable diversity and the dynamicity of resources make it necessary to automatically adapt a category hierarchy to specific and dynamic resources.

In this thesis, the ScienceDirect taxonomy is used to build the category dimension in a scientific resource space. However, the category hierarchy is too general to provide detailed classification in specific subject areas. For example, directly using the ScienceDirect category hierarchy for the organization of scientific articles in the field of AI will result in missing branch subject categories, since AI is a leaf category in the ScienceDirect taxonomy.

These problems necessitate an automatic method for category hierarchy maintenance in order to construct a fine category dimension in a scientific resource space. Manual maintenance is rather tedious and difficult, because it is hard to discover changes within categories and

emerging topics. This motivates our studies on the challenging task of modifying a category hierarchy to make it more appropriate to specific resources and achieve better classification accuracy. This thesis proposes a general maintenance approach that is applicable to not only scientific articles but also other types of resources such as news and webpages.

1.4.2 Automatic Construction of the Concept Hierarchy in Micro Dimensions

A scientific resource space provides semantic descriptions on contents of scientific articles through the concept hierarchy in the micro dimensions. Specifically, a scientific resource space contains three micro dimensions: Task, Process and Material, which describe the contents related to research problem, methodology and data respectively.

Constructing the concept hierarchy in the micro dimensions means that we need to analyse the contents of scientific articles to recognize the three types of entities (Task, Process and Material) and extract two types of relations (Hyponym-of and Synonym-of) between entities. Each entity type corresponds to one dimension and the extracted entities of the same type are used to generate one concept hierarchy on each micro dimension. Thus how to recognize the three types of entities and extract the relations between entities with the same entity types is a key problem in constructing micro dimensions.

This thesis proposes a joint entity/relation extraction model based on deep neural network to automatically extract entities and relations from scientific articles. The entity recognition and relation extraction tasks are related to each other and thus can be modelled in a united deep neural network so as to prompt performance of each other.

1.4.3 Summarization Service in Scientific Resource Space

A scientific resource space supports a series of services to help users accurately and efficiently get useful information. This thesis proposes scientific comparative summarization based on the concept of coordinate partition in the resource space model. Scientific comparative summarization aims to summarizing the differences among a collection of scientific document groups.

Reviewing the definition of coordinate partition in section 1.2, for any coordinate C on an axis X' in a resource space, resources defined by coordinate C can be partitioned by an axis X other than X', and the coordinate partition on C produces n classes corresponding to the n coordinates $\{C_1, C_2, ..., C_n\}$ on axis X. The coordinate C is called the original coordinate and the axis X is called the partition axis.

In a scientific resource space, the summarization based on the concept of coordinate partition is performed by first choosing an original coordinate and a partition axis, and then conducting the coordinate partition to classify the resources under the original coordinate into categories on the partition axis, and generating a summary for each category. The summarization based on the coordinate partition in a scientific resource space is therefore a form of multi-document

summarization for comparing differences among categories. Figure 1.5 shows the coordinate partition and summary generation in a scientific resource space, where the original coordinate is chosen from the *Task* dimension and the partition axis is the *Process* dimension. The partition produces three categories associated with the three coordinates on the *Process* dimension.

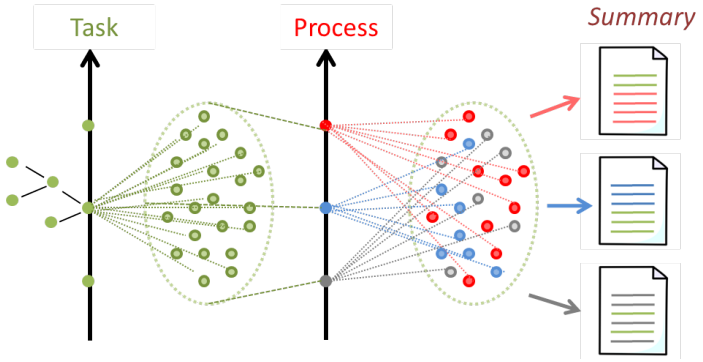


Figure 1.5: Coordinate Partition in scientific resource space and comparative summary generation.

The coordinate partition can classify resources defined by any coordinate using any other partition axis, and then generate a summary for each category. In a scientific resource space, the summary based on the coordinate partition could help researchers to solve some practical problems in scientific information retrieval. For example, it can facilitate the comparison on different methods or on different research problems. Summaries produced by partitioning resources under an original coordinate on the *Task* dimension and using the *Process* dimension as the partition axis could help to compare different methods applied to a same problem. Also, summaries produced by partitioning resources under an original coordinate on the *Process* dimension and using the *Task* dimension as the partition axis could help to compare different research problems solved by one same method.

The summarization based on the coordinate partition in a scientific resource space possesses the following two characteristics:

1. The scientific articles in different categories produced by the coordinate partition are belonging to the same original coordinate, thus the categories contain a large number of similar content. The content represented by the original coordinate is the common theme shared by the different categories.

2. The categories also contain some specific content unique to each category, which correspond to coordinates on the partition axis. The category-specific content represented by each coordinate on the partition axis is the category-specific theme.

The above two characteristics mean that we cannot simply use generic summarization methods to produce summaries based on the coordinate partition. Generic summarization methods always summarize the important information that is delivered in most of the documents. When summarizing with generic summarization methods, sentences talking about the common theme

are likely to be selected, which leads to the occurrence of common information in each category summary. The summarization based on the coordinate partition in a scientific resource space aims to provide comparative summaries by comparing different categories and capturing the distinctiveness of each category. Thus it necessitates a scientific comparative summarization method that captures more of the unique content concerning category-specific themes and reduces the content concerning the common theme. This thesis proposes a comparative summarization method based on the differential topic model, which is able to generate comparative summaries for scientific articles.

1.5 Significance of Automatic Construction Methods

The scientific resource space is an instance of the resource space model, which is aimed at organizing, storing, managing and retrieving massive scientific literature resources. The following steps have been proposed in order to manually design a general resource space in a bottom-up manner (Zhuge, 2004; Zhuge & Xing, 2012):

1. *Resource analysis* is an investigation of the application scope to learn resources and build a resource dictionary for all resources that need to be organized in a resource space. The resource dictionary is usually stored in a structured XML file containing resource properties as elements such as name, author, version, location and privilege.

2. *Top-down resource partition* is performed to form a consensus on top-level resource partition, since designers may have different views on resource classification. Categories can be subdivided top-down to provide fine-grained classification semantics.

3. *Low-dimensional resource spaces* (usually 2-dimensional spaces) can be first constructed and then joined to form a complete resource space. This step includes determining the number of spaces, building axes in each space, generating coordinate hierarchies for each axis and checking normal form constrains in the space. Building low-dimensional resource spaces is much easier than directly building a high-dimensional space.

4. *Joining low-dimensional spaces* is implemented by merging a set of low-dimensional resource spaces in order to generate a complete resource space that offers a universal resource view.

The above construction process provides a general guidance for building a resource space for any type of resources. However, different types of resources have different properties and resource properties essentially determine specific construction methods for different resource spaces. Thus it is difficult to provide a general framework to build a resource space for all types of resources.

In terms of building a scientific resource space, the construction process requires a subject category or a concept hierarchy to classify resources. Manual construction involves a heavy workload and low efficiency. The final resulting resource space can also be influenced by many individual factors, such as personal knowledge and design skills. To ease the process of manual design, this thesis studies the problem of automatic construction of a resource space for scientific literature resources, which is of great significance to the application of the resource space model.

1.6 Contributions

This thesis uses the resource space model to organize scientific literature resources and proposes automatic methods to construct a scientific resource space. It is a specific application of the resource space model for improving the efficiency of storing, retrieving and utilizing scientific resources. A scientific resource space contains two types of dimensions: macro-dimensions describe the metadata of scientific articles and support metadata retrieval, while micro-dimensions describe the content of scientific articles and support content retrieval.

This thesis mainly focuses on the automatic construction of a scientific resource space, including an automatic maintenance approach to modifying the category hierarchy in the macro dimension and an automatic construction approach to creating concept hierarchies in micro dimensions. The scientific resource space can support summarization service based on the concept of coordinate partition in the resource space model so as to facilitate the comparison of different methods or different research problems. A series of comparative summarization methods based on the differential topic model are proposed. The main contributions of this thesis are as follows:

1. The category hierarchy in the macro-dimension needs to evolve in order to satisfy the dynamic requirements of organization and management of resources. This thesis proposes an automatic maintenance approach, which modifies the original category hierarchy according to the hierarchical clustering of resources. A series of comparison experiments on Reuters-21578, 20Newsgroups, DMOZ and scientific articles from ScienceDirect provide evidence that the method is effective.

2. This thesis proposes a joint entity/relation extraction model based on a deep neural network to automatically extract three types of entities (Task, Process and Material) and two types of relations (Hyponym-of and Synonym-of) from scientific articles in order to build concept hierarchies on micro-dimensions in a scientific resource space. Experimental results on Semeval 2017 Task 10 dataset for the tasks of entity recognition and relation extraction show the effectiveness of the joint model.

3. This thesis proposes the new task of scientific comparative summarization based on the concept of coordinate partition. Novel comparative summarization methods based on

the differential topic model are designed to provide summarization service in the scientific resource space. This thesis creates a new dataset for the scientific comparative summarization task and shows the effectiveness of the proposed comparative summarization methods on this dataset.

1.7 Thesis Outline

The remainder of this thesis consists of five chapters, which can be summarized as follows:

Chapter 2: This chapter focuses on the automatic maintenance of the category hierarchy. We first analyse the reasons why category hierarchies change, a situation which necessitates our research on category hierarchy maintenance. We review the related work on category hierarchy generation and maintenance, and analyse four typical structures of category hierarchies that need modification. A two-phase maintenance approach is proposed to modify the category hierarchy, which relies on a hierarchical clustering of the resources. Finally, a series of experiments on various datasets are conducted to compare the classification performances for three types of hierarchies. The modified hierarchy is evaluated against two baselines, namely the orinal hierarchy and the automatically generated hierarchy. The experimental results show that the modified hierarchy outperforms the other two types of hierarchies, which proves the effectiveness of our hierarchy maintenance approach.

Chapter 3: In this chapter, we address the problem of the automatic generation of the concept hierarchies. We first analyse the problem and divide the task into two subtasks of entity recognition and relation extraction. Three areas of related work on scientific discourse analysis, entity recognition and relation extraction are introduced. Then we propose a joint entity/relation extraction model based on deep neural network to automatically extract entities (Task, Process and Material) and relations (Hyponym-of and Synonym-of) from scientific articles in order to build concept hierarchies on micro-dimensions in a scientific resource space. Finally we conduct experiments on the Semeval 2017 Task 10 scientific dataset for the tasks of entity recognition and relation extraction. Experimental results show the effectiveness of the joint model on both tasks.

Chapter 4: We describe the implementation of a prototype system of the scientific resource space which supports browsing, retrieval and summarization services on scientific articles. This chapter demonstrates the function design and the user interface design of the prototype system. The development of an advanced system based on the prototype is still ongoing.

Chapter 5: This chapter focuses on the summarization service in a scientific resource space. We first introduce the new task of scientific comparative summarization based on the concept of coordinate partition. We then review the related work on various types of summarization: generic multi-document summarization, comparative summarization and update summarization. Next, comparative summarization methods based on the differential topic model are proposed,

including two *dTM* models (*dTM*-Dirichlet and *dTM*-SAGE) and two sentence scoring strategies. Finally, we collect and annotate 129 scientific papers for the comparative summarization task and conduct a number of experiments on this dataset to show the effectiveness of the *dTM*-based methods in terms of the summarization performance.

Chapter 6: This chapter summarizes the outcomes of each previous chapter and discusses future research directions with possible solutions.

2 Macro-dimension Construction: Automatic Maintenance of Category Hierarchy

2.1 Overview of the Problem

Classification is a daily activity of grouping and distinguishing objects according to their commonalities and characters to help human understand and perceive the world. Hierarchical classification is a basic method to organize large-scale resources in different categories at different abstraction levels. Categories at higher levels are more general than those at lower levels. Hierarchical categories have been widely used to organize Web resources such as Open Directory Project (*ODP*), Wikipedia, and Yahoo! Directory.

Compared with flat classification systems, hierarchical classification systems enable easier browse and retrieval on resources. Users prefer to search along defined categories, especially when they have no acquaintance with the domain knowledge. It has been shown that hierarchical classification systems outperform their flat counterparts in training efficiency, classification efficiency, and classification accuracy (Tang, Zhang, & Liu, 2006). However, the impact of a poor category hierarchy will directly lead to the failure of resource classification and information retrieval. Whether a category hierarchy could have positive impacts on classification and retrieval depends on the following two aspects:

1. whether the category hierarchy could express fine classification semantics and fit for resources;

2. whether the category hierarchy could guarantee the classification accuracy.

Category hierarchy may well fit resources at the time of construction, but categories may change after continually adding new diverse resources. The unpredicted diversity and dynamicity of resources make it necessary to adapt category hierarchies to new coming resources. In March 2014, Open Directory Project (*ODP*) created a new category relating to Malaysia Airlines flight 370 under category *Accidents* and earlier in May 2013 it added *Wearable Electronics* under *Hardware*. The ACM classification system has also modified its classification hierarchy 3 times (in 1991, 1998 and 2012) during the last twenty years. Some commercial websites like Amazon and eBay adjusted their category hierarchy more frequently, since there often emerge new types of items. Given the fast growth of Web resources, continuously accommodating large amount of new diverse resources into a hierarchy is necessary.

Apart from the necessity of adjusting Web category hierarchy to adapt Web resources, our everyday-increasing personal resources also need to be organized in an appropriate category hierarchy so that they can be searched and managed in an efficient way. For example, researchers download large number of scientific papers to keep up with the updating knowledge and save them in the hierarchical file system. However, with limited time and energy they dont have enough time to read every paper, not to mention that they can hierarchically classified each paper into the most relevant category and keep modifying the category hierarchy by creating a new category or merging old categories to best fit the resources. When a researcher changes his research interest, the hierarchy will need global modifications as well as local adjustments.

RSM is a way to manage big volume of resources by multiple classifications. Since the initial design of a space needs to be adapted to manage new resources, the maintenance of category hierarchy is a key component of the RSM. To build the "*Category*" dimension in a scientific resource space, we utilize the ScienceDirect taxonomy as an initial category hierarchy, however, this category hierarchy is too general to fit specific resources. For example, the ScienceDirect category hierarchy fails to organize resources in Artificial Intelligence (AI) because AI is a leaf category in the hierarchy.

Therefore, it necessitates our research on category hierarchy maintenance. Manual maintenance is rather tedious and difficult, because it is hard to discover changes on categories and emerging topics in large number of resources. This motivates our study on the challenging task of automatically modifying a category hierarchy to make it more appropriate to specific resources and achieve better classification accuracy. We focus on the general text resources and leave the problem of extending our method to other types of resources (like image resources and video resources) to the following study.

In this chapter, we propose a method called Automatic Maintenance of Hierarchical Category ($AMHC$) for modifying the category hierarchy through two-phase adjustments, namely the global phase and the local phase. It can be used to make the category hierarchy (such as Wikipedia, ODP and Yahoo! Directory) more suitable for organizing the specific resources that existing category hierarchies are too general to organize.

The global phase is performed to adjust the category hierarchy according to a cluster tree, since hierarchical clustering can provide a data-driven method for automatic discovery of similarity relations between categories. It can help detect inappropriately located categories and directly adjust them to appropriate position from a global point of view. The global phase makes the pre-defined categories satisfy the pattern of resources by combining the category hierarchy and the cluster tree.

The local phase is performed to detect topical changes in some categories by Latent Dirichlet Allocation (LDA) topic model (Blei, Ng, & Jordan, 2003). The statistical topic models can discover a broad range of hidden themes but lack of interpretability. However, human-defined

categories tend to be acceptable but they tend not to cover the themes exhaustively. The local phase combines category hierarchy and topic model, making the pre-defined categories better reflect the topics of the resources.

The global phase makes cross-branch adjustments which cover a wide range of a hierarchy. The local phase uses three elementary operations (namely Merge, Pull-Up and Split) to modify a category that is only related to its parent or sibling category in a local range.

To evaluate the quality of a hierarchy, we propose a new evaluation measure that considers not only the balance of the hierarchical structure but also the ability of expressing classification. The measure uses the Entropy to measure the uncertainty of classification, balance of structure and resource distribution.

We conduct experiments on the datasets of various scales. The AMHC-modified hierarchy is evaluated against the original hierarchy and the automatically generated hierarchy. Our experimental results show that classifiers trained on the modified hierarchy can get better classification performance than that on the original hierarchy and automatically generated hierarchy, which verifies that the modified hierarchy has more topically cohesive categories than the other two hierarchies. Besides, the comparison of the evaluation measures also shows that the proposed measure is more suitable for evaluating the quality of a hierarchy than the traditional measures.

2.2 Related Work

2.2.1 Category Hierarchy Generation

Category hierarchy generation is to construct a tree-structured hierarchy from a set of documents reflecting different levels abstraction. One line of research explored traditional hierarchical clustering, either agglomerative or divisive, generating a tree-structured hierarchy by grouping documents according to a similarity measure. A parallel line explored the hierarchical probabilistic topic models, with the goal of learning a latent topic hierarchy from a corpus of documents.

2.2.1.1 Hierarchy Generation based on Hierarchical Clustering

One technique route of category hierarchy generation relies on traditional hierarchical clustering to generate a tree structure (called dendogram) to represent a sequence of partitions with one most inclusive cluster at the top and single-point clusters at the bottom. Each intermediated cluster is produced by merging two similar clusters from the lower level or splitting a cluster from the higher level. According to the generation process of intermediated clusters, hierarchical solutions can be divided into two categories: agglomerative algorithms and partitional algorithms.

Hierarchical agglomerative clustering (HAC) algorithms build a hierarchy in a bottom-up

manner by initially assigning each document to one cluster and merges the most similar pair of clusters at each step until there is only one left (Guha, Rastogi, & Shim, 1998; Karypis, Han, & Kumar, 1999). Typical hierarchical agglomerative clustering procedure is described in (Steinbach, Karypis, & Kumar, 2000). The core of a HAC algorithm is the function used to measure the similarity between each pair of clusters C_i and C_j. Four typical inter-cluster similarity functions are introduced below:

1. Single-Link (SL) function (Sneath & Sokal, 1973). The single-link function measuring the similarity of two clusters is defined as the maximum similarity between two documents from each cluster. By the single-link function, the similarity of two clusters C_i and C_j is given by $Sim_{SL}(C_i, C_j) = Max_{d_a \in C_i, d_b \in C_j} cos(d_a, d_b)$;

2. Complete-Link (CL) function (King, 1967). The complete-link function of two clusters C_i and C_j is defined as the smallest similarity between two documents from each cluster. That is: $Sim_{CL}(C_i, C_j) = Min_{d_a \in C_i, d_b \in C_j} cos(d_a, d_b)$;

3. Average-Link (AL) function (Jain & Dubes, 1988). The average-link function of two clusters C_i and C_j is defined as the average of all pairwise similarities between the documents in both clusters: $Sim_{AL}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{d_a \in C_i, d_b \in C_j} cos(d_a, d_b)$;

4. Centroid (CE) function (Aggarwal, Gates, & Yu, 1999). The centroid function of two clusters C_i and C_j is defined as the similarity between the centroids of the two clusters: $Sim_{CE}(C_i, C_j) = cos(c_i, c_j)$.

Discussions on the Similarity Functions. Different similarity functions have different impact on the dendrogram structures. Clusters produced by the single-link function are usually isolated but not cohesive. It tends to produce long chains consisting of loose clusters. At the other extreme, the complete-link function produces tight and cohesive clusters that may not be isolated. The average-link function represents a compromise between the two extremes and it can perform simple, efficient and stable hierarchical clustering. The centroid method is another commonly used similarity measurement function that can perform as well as the average-link function.

Time Complexity Analysis. There are two time-consuming steps in HAC algorithms. One is to compute pairwise similarities between all the documents, which will cost $O(n^2)$ time complexity. The other step is to recursively select the most similar pair of clusters to merge. A simple method is to re-compute the gains achieved by merging each pair of clusters after each level of the agglomeration and select the most promising pair. At the i^{th} agglomeration step, this costs $O((n-i)^2)$ time, leading to an overall time complexity of $O(n^3)$.

Hierarchical partitional clustering (HPC) algorithms carries out top down with one most inclusive cluster, and then split a least cohesive cluster at each step until it reaches the expected

number of clusters (Dhillon & Modha, 2001). In most cases, partitional approaches are inferior to the agglomerative approaches in terms of clustering quality such as F-measure and Entropy measure (Puzicha, Hofmann, & Buhmann, 2000). A key point of HPC approaches is to decide which cluster to split. Five typical cluster criterion functions are shown in Table 2.1 whose optimization drives the entire clustering process (Steinbach et al., 2000). The clustering problem can be stated as at each step selecting a cluster to split such that the value of a particular criterion function is optimized.

Table 2.1: Clustering criterion functions.

Name	Criterion Functions
I1	Maximize $\sum_{r=1}^{k} n_r(\frac{1}{n_r^2}\sum_{d_i,d_j\in C_r} cos(d_i,d_j))$
I2	Maximize $\sum_{r=1}^{k}\sum_{d_i\in C_r} cos(d_i,c_r)$
E1	Minimize $\sum_{r=1}^{k} n_r cos(c_r,c)$
H1	Maximize $\frac{I1}{E1}$
H2	Maximize $\frac{I2}{E1}$

Discussions on the Criterion Functions. The first criterion function $I1$ (Puzicha et al., 2000) is to maximize intra-cluster similarity that is the sum of average pairwise similarities between documents of the same cluster weighted according to the size of each cluster. The second criterion function $I2$ (Steinbach et al., 2000) is also trying to maximize the intra-cluster similarity, however, it uses a different representing form in which each cluster is represented by its own centroid vector and thus $I2$ is to maximize the similarity between each document and the centroid of the cluster that the document is assigned to. It has been shown both by theoretical proof and by experimental results in (Zhao, Karypis, & Fayyad, 2005) that $I2$ is more biased to choose clusters with smaller intra-cluster similarity compared to their higher intra-cluster similarity counterparts (Steinbach et al., 2000).

Instead of optimizing the intra-cluster similarity of $I1$ and $I2$, the third criterion function $E1$ is trying to find a solution that minimizes the inter-cluster similarities between clusters to make them distinguishable from each other as much as possible. The idea behind this solution is to separate the documents of each cluster from the entire collection and thus it minimizes the similarity between the centroid vector of each cluster and the centroid vector of the entire collection.

The $H1$ and $H2$ criterion functions are respectively obtained by combining $I1$ with $E1$ and $I2$ with $E1$ trying to get better clustering quality in terms of maximum intra-cluster similarity and minimum inter-cluster similarity.

Time Complexity Analysis. One of the advantages of the HPC algorithms is that it has relatively low time complexity. It has been shown in (Zhao et al., 2005) that a two-way HPC algorithm can achieve in a linear time complexity of the number of documents, since in most cases the number of iterations consumed by selecting a cluster to split is rather small (less than

20) and is independent of the number of documents. Therefore, the overall time complexity in building a dendrogram of HPC containing $n-1$ bisections is $O(nlogn)$.

Hierarchical clustering helps generate a tree-structured dendrogram to reflect the similarity relationship of categories, but unfortunately it can only generate a binary tree with too many small clusters, since hierarchical clustering could only merge a pair of clusters in HAC or bisect a cluster into two sub-clusters in HPC at each step. Therefore, how to generate a multi-way hierarchical taxonomy is a vital problem faced by the hierarchical clustering approaches.

Many studies focus on this problem and propose different solutions to transform a binary tree into a multi-way tree as a category hierarchy. One representative study is to employ HAC with the single-link similarity function to build a hierarchy (Aggarwal et al., 1999), where centroids of each category are used as initial seeds. To change the binary tree into a multi-way category hierarchy, all the clusters whose similarity is higher than a threshold value are merged. It has been claimed that the generated hierarchy is at least not worse than the pre-defined one from the experimental results in terms of the cluster quality and human interpretability. A major problem of this method is that it is hard to find an ideal global threshold that determines the merging process for all categories.

A method HAC+P is proposed to overcome the problem by adding a post-processing min-max partition to change the binary tree into a multi-branch tree (Chuang & Chien, 2004). In min-max partition process, the hierarchy is recursively decomposed into sub-hierarchies by selecting the best level to minimize a criteria function that considers the cluster set quality and the cluster number preference. It is a simple approach for category hierarchy generation and has been widely used as a baseline, however, in most cases the criteria function is prone to the upper cut levels. The difficulties of setting too many parameters make this method perplexed.

A linear discriminant projection is proposed to transform all data into a lower dimensional space and HAC with the centroid function is employed to generate a binary tree (T. Li, Zhu, & Ogihara, 2007). Then the binary tree is changed into a two-level multi-way category hierarchy by clipping the binary tree at the point where the cluster merging distances increase sharply. It has been shown that the generated category hierarchy could guarantee the maximum inter-class separation between clusters and group the most similar categories at the top level, but the reasonability of the two-level hierarchy is unclear.

2.2.1.2 Hierarchy Generation based on Hierarchical Probabilistic Topic Models

A parallel line of study explored the hierarchical probabilistic topic models, such as hLDA and nHDP, so as to learn a latent topic hierarchy from a corpus of documents. In such hierarchies, each internal node or topic reflects the shared terminology or vocabulary of the documents.

The hierarchical latent dirichlet allocation (*hLDA*) model (Griffiths, Jordan, Tenenbaum, & Blei, 2004) is to learn a tree-structured topic hierarchy from a corpus of documents by placing a

structure prior on possible hierarchies. In hLDA, the nested Chinese restaurant process (*cCRP*) is used as the nonparametric Bayesian prior. It is limited in that each document is generated from the topics on a single path of the tree. According to nCRP, hLDA first chooses a path for each document and then samples L-dimensional topic mixture proportions along the path from a Dirichlet distribution. Finally, it draws each word in the document from the L topics on the path from the root to a leaf. This single-path limitation has practical drawbacks in modelling cross-field documents with parallel topics, because hLDA restricts any two topics of a document must have a relationship that one topic is a subtopic of the other.

To overcome the limitations in hLDA, the most recent model nested hierarchical Dirichlet processes (*nHDP*) is proposed by (Paisley, Wang, Blei, & Jordan, 2015), which develops a new Bayesian nonparametric prior nHDP to replace nCRP providing uncertainty on possible tree structures. This new prior enables each word in a document to have access to the entire tree rather than a single path, through associating each document a document-specific distribution on the paths within the tree.

However, the limitation of hierarchical topic models for the task of hierarchy generation is that each internal node is a distribution of words, thus lacks of interpretability. The word-distribution-represented topics are far from what we expect as a category. In hierarchical topic models, the internal nodes just reflect the co-occurrences of words rather than the summarization of children nodes.

In short, hierarchical topic models are not suitable for directly constructing a reliable and satisfactory category hierarchy to organize and classify resources. It needs much more post-processing operations on the tree to transform it into a subject-based category hierarchy to become semantically meaningful.

2.2.2 Category Hierarchy Maintenance

Different from the hierarchy generation, hierarchy maintenance focuses on the modification of an existing hierarchy to make it better reflect the topics of its resources and achieve higher classification accuracy.

A method of modifying a hierarchy using three operations (Promote, Merge and Demote) is proposed (Tang et al., 2006). For each category, promote operation is tested, followed by merge and demote operations, in a top-down manner. The operation comes into effect if it can improve the classification accuracy. The approach iterates the process until no improvement can be observed. In experiments, this method outperforms clustering-based hierarchy generation method in terms of classification accuracy. However, there are two major problems. One is that this method has a high time-complexity since it tests three operations on all nodes in the hierarchy. The other is that the modification cant change leaf categories and retains less cohesive leaf categories in a hierarchy, which occurs in most cases of real life applications.

A data-driven approach for hierarchy maintenance defines three operations (Sprout, Merge and Assign) with reference to an auxiliary hierarchy that covers a similar set of topics (Yuan, Cong, Sun, Lin, & Thalmann, 2012). This method can discover finer categories by projecting the documents in the given hierarchy to an auxiliary hierarchy. However, the discovery of some new topics depends on the auxiliary hierarchy which is not always easy to get, so in some cases it will become a limitation of this method.

As for hierarchy evaluation, it is non-trivial for computers to simulate human evaluation method, judging whether the hierarchy taxonomy can reflect accurate classification semantics and keep balance among all branches and whether the resources are evenly distributed. Most of the current studies rely on F-measure, Precision and Recall to evaluate the hierarchical classification methods (Y. Yang & Liu, 1999; Sun & Lim, 2001). However, these measures are not adequate to evaluate the quality of a hierarchy since they have completely ignored the impact of the structural balance and the resource distribution. For the hierarchy maintenance task, an evaluation measure that considers different aspects of a hierarchy is required.

To conclude, Table 2.2 shows the comparisons of the five multi-way category hierarchy generation and maintenance methods from the following six aspects: (1) whether the method needs an initial hierarchy to guide the generation or maintenance process; (2) whether the method uses an auxiliary hierarchy to help find new topics; (3) the final category hierarchy is a binary tree or a multi-way tree; (4) the final category hierarchy is a two-level hierarchy or a multi-level hierarchy; (5) whether the method can change inappropriate leaf categories or not; (6) whether the method uses a new hierarchy measure to evaluate the quality of a category hierarchy.

Table 2.2: Comparisons of typical category hierarchy construction and maintenance methods.

Aspects	Aggarwal,1999	Li,2007	Chuang,2004	Tang,2006	Yuan,2012	Our AMHC
Use initial hierarchy	NO	NO	NO	YES	YES	YES
Use auxiliary hierarchy	NO	NO	NO	NO	YES	NO
Multi-way or Binary Tree	Binary Tree	Multi-Way	Multi-Way	Multi-Way	Multi-Way	Multi-Way
Multi-level or Two-level	Multi-level	Two-level	Multi-level	Multi-level	Multi-level	Multi-level
Change leaf categories	YES	NO	NO	NO	YES	YES
Use new hierarchy measure	NO	YES	YES	NO	NO	YES

From Table 2.2, it can be found that category hierarchy generated by hierarchical clustering algorithms requires too many user inputs (Chuang & Chien, 2004), otherwise it can only generate a binary hierarchy (Aggarwal et al., 1999) or a two-level hierarchy (T. Li et al., 2007). As for category hierarchy maintenance, most researches try to modify a hierarchy by brutally testing operations on all categories which results in high time-complexity. Although some pruning strategies are proposed, it sacrifices the global optimal solution and fails to adjust cross-branch inappropriate categories (Tang et al., 2006). In addition, some approaches rely on an auxiliary hierarchy to discover new emerging topics, which limits the application scope and reduces the feasibility of the method (Yuan et al., 2012).

Different methods are also compared with our AMHC approach in Table 2.2. Although our AMHC approach relies on HAC to generate a binary cluster tree to judge the similarity between categories, the modified hierarchy is a multi-way tree that keeps similar levels of abstraction to the original hierarchy satisfying human understanding of taxonomy, which bypasses the problem of only generating a two-level hierarchy in (T. Li et al., 2007). It can also solve the problem of failing to make cross-branch adjustments in (Tang et al., 2006) by adding a global modification phase that significantly speeds up the cross-branch movements of inappropriately located categories thus reducing the time-complexity. It split less cohesive leaf categories to overcome unchanged leaf categories. Compared to (Yuan et al., 2012), an auxiliary hierarchy is not necessary to discover new topics, since we use LDA topic model in the local phase to detect the topics and guide the Merge, Pull-Up and Split operations.

2.3 Typical Structures Analysis

When using existing hierarchical categories, such as ODP, Wikipedia and Yahoo! Directory, to organize resources, inconsistences often exist between hierarchical categories and resources, which leads to inefficient management of the resources. There are four typical cases of a category hierarchy that need adjustments:

- Case 1: Parent category can no longer represent its child category.
- Case 2: Two categories under the same parent share too many common features to distinguish them clearly.
- Case 3: A category belongs to more than one parent category.
- Case 4: Leaf categories become less cohesive with new coming resources.

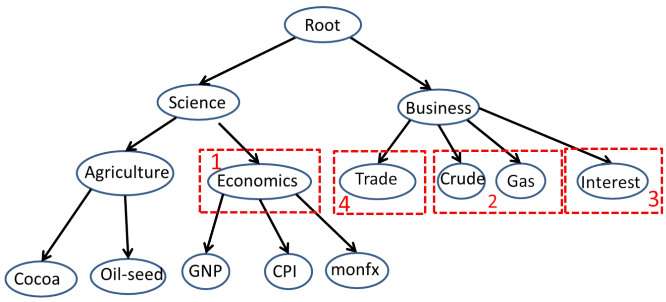


Figure 2.1: The original category hierarchy.

The above cases are illustrated by Figure 2.1, a category hierarchy that is generated for Reuters-21578 dataset according to ODP directory. The categories marked by dashed rectangles in red colour correspond to the above cases. Clustering the categories in the hierarchy produced a cluster tree shown in Figure 2.2, where each node number represents the merge order in the

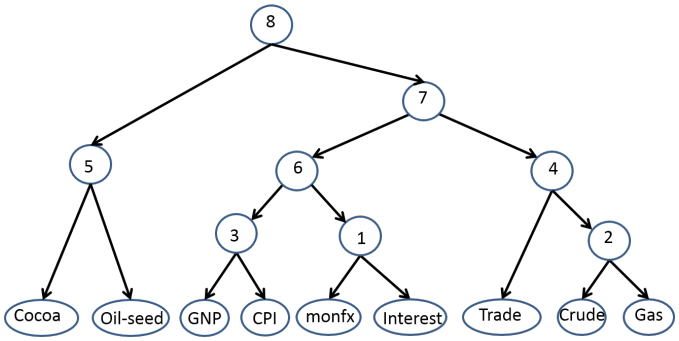


Figure 2.2: The cluster tree of the categories in the original hierarchy.

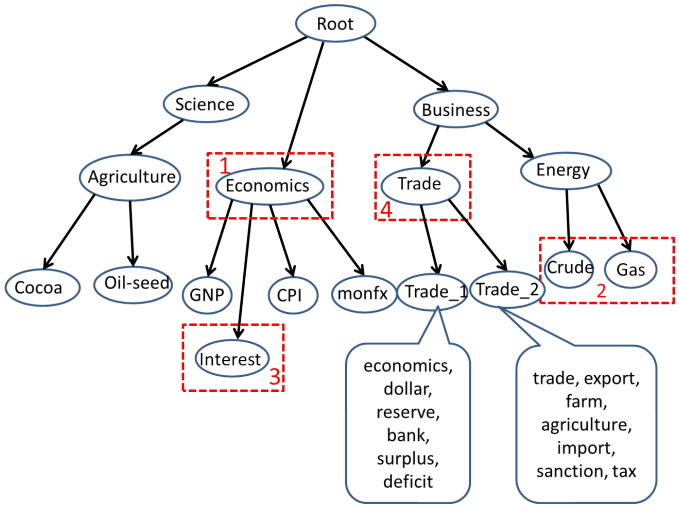


Figure 2.3: The modified category hierarchy.

hierarchical clustering process. The smaller the node number is, the earlier the node generates and the more similar the two categories are.

Four modification strategies are proposed to modify the typical inappropriate categories in the original category hierarchy according to the cluster tree. Figure 2.3 shows a modified category hierarchy evolved from the original hierarchy in Figure 2.1.

Modification Strategy for Case 1. Pull the child category up to its parent level to avoid the inappropriate influence from the parent.

In Figure 2.1, the categories Agriculture and Economics are both under category Science, but the cluster tree in Figure 2.2 shows that the node 6 representing Economics and the node 4 representing Business have a larger similarity, thus resources of Retuers-21578 in Economics are more related to the category Business than to the category Agriculture. A better solution is to pull Economics up to the upper level as shown in Figure 2.3. This operation leads to a better classification performance according to the Macro-F1 that raised from 0.84 to 0.93.

Modification Strategy for Case 2. Merge similar categories under the same parent to form a super node.

By selecting the common features, we can firstly distinguish the similar categories from others and then focus on more specific features to separate the similar categories at the lower level.

As shown in Figure 2.1, the category Business contains two similar subcategories Crude and Gas. They can be merged into a super category Energy shown in Figure 2.3. As the result of the operation, we get a better classification performance indicated by Macro-F1, increasing from 0.65 to 0.79.

Modification Strategy for Case 3. If a category belongs to more than one parent, the category should be put under the most related parent to achieve better classification accuracy.

As shown in Figure 2.1, the category Interest is originally under the category Business, however, in the cluster tree Figure 2.2 the most similar category is monfx which under the category Economics, thus we move the category Interest to put it under a new parent Economics in Figure 2.3. After this operation, the Macro-F1 increases from 0.83 to 0.90.

When new resources are continually added to the category hierarchy, leaf categories are more likely to emerge new topics, and thus become less cohesive. The following strategy is necessary.

Modification Strategy for Case 4. Split the less cohesive leaf category into finer subcategories.

Applying this strategy to split the less cohesive category Trade in Figure 2.1 into Trade-1 (a category related to the relationship between trade and economics) and Trade-2 (a category related to the import and export trade policy among countries) as shown in Figure 2.3. After this operation, leaf categories become more cohesive and the category intra-similarity increased from 0.683 to 0.734.

Based on the above four cases and modification strategies, we develop a two-phase category hierarchy maintenance method. The global phase solves the issue of case 3 by directly moving inappropriate child categories to their better parents within a global scope. The local phase addresses the other three cases through detecting topical changes in some categories and using three elementary operations (Merge, Pull-Up and Split) to modify a category that is only related to its parent or sibling category within a local range. The two-phase approach can make a hierarchical category more suitable to organize the resources that cannot be represented by existing categories.

2.4 Automatic Maintenance

Making abstraction among categories and measuring the similarity between categories are two basic behaviors to generate a category hierarchy. Humans are good at making abstraction but limited in ability to calculate the similarities between large-scale resources. Computing models are good at calculating the similarities between large-scale resources but limited in ability to make abstraction. To make both advantages of humans and computers, our Automatic Maintenance of Hierarchical Category ($AMHC$) approach use a global phase and a local phase to maintain the category hierarchy within two different scales.

The global phase gets initial human-defined hierarchy and then makes use of hierarchical

clustering to get similarity between categories to detect inappropriately located categories. The local phase detects topical changes by LDA topic model (Blei et al., 2003) and then adjusts with three local operations: Merge, Pull-Up and Split.

2.4.1 Phase 1: Global Modification

A hierarchy evolves when the number of new resources reaches a certain degree. To adjust the category hierarchy, we need to detect the pattern change of similarity between categories to guide the category hierarchy evolvement. Hierarchical clustering can generate a cluster tree that reflects the similarity of categories, but it can only generate a binary tree with specific clusters.

Algorithm 1: AMHC Global Modification

```
   Input: Cla_HT, Clu_HT
   Output: HT
 1 Eva_Score = evaluateHT(Cla_HT);
 2 HT ← Cla_HT;
 3 AdjustNodeList=null;
 4 AdjustNodeList ← Mapping(Cla_HT, Clu_HT);
 5 while AdjustNodeList ≠ null do
 6 |   Node ← getNode(AdjustNodeList);
 7 |   H_List ← generateCandidates(Node,HT,Clu_HT);
 8 |   [H_temp,score] ← getBest(H_list);
 9 |   if score < Eva_score then
10 |   |   Eva_score=score;
11 |   |   HT=HT_temp;
12 |   end
13 end
14 HT ← PostProcess(HT);
15 return HT;
```

How to adjust a category hierarchy according to the hierarchical cluster tree of resources and keep the levels of abstraction similar to the original one is the main problem of global modification. To address the problem, we firstly build one-to-one mappings between categories in category hierarchy and cluster tree, and then adjust the category hierarchy. Algorithm 1 illustrates the global modification process.

Two trees are used in the global modification algorithm. One is the classification tree, which is a pre-existing category hierarchy. Each node in the tree represents a category corresponding to a set of resources. This classification tree may contain inappropriately located categories and our global modification algorithm improves the classification tree by adjusting these categories into appropriate positions. The output of the algorithm is also a classification tree that is evolved from this initial classification tree.

The other is the cluster tree, which is a binary tree generated by hierarchical agglomerative clustering algorithm. Instead of building the cluster tree from the resources set, the hierarchical agglomerative clustering algorithm uses information from the pre-existing category hierarchy and builds the solution from a category set, that is, the set of leaf nodes in the classification

tree. The cluster tree is built by firstly assigning each leaf category to its own cluster and then repeatedly merging pairs of clusters to obtain a single all-inclusive cluster. The average-link function (Jain & Dubes, 1988) is used to determine the most similar pair of clusters to be merged at each step. The cluster tree truly reflects the similarity relationship of categories, but it is hard to regard a binary tree as a good category hierarchy.

The general process of global modification consists of two major procedures mapping procedure (line 4) and candidates generating procedure (line 7). The algorithm takes a classification tree Cla_HT and a cluster tree Clu_HT as the input and then outputs the final modified category hierarchy HT. It firstly evaluates Cla_HT (line 1) by the proposed evaluation measure. The smaller the value, the better quality a hierarchy has. Then, it proceeds with the mapping procedure (line 4) between Cla_HT and Clu_HT. After that, we will get a list of categories to be adjusted (AdjustNodesList). For each node in the list (line 5-11), it generates the candidates (line 7) and gets the best one H_temp (line 8). It tests the evaluation score of H_temp and decides whether to accept it or not (line 9-10). At last, it carries out a post-process (line 12) on the final hierarchy to avoid unary branching situations that commonly occur in candidates.

Figure 2.4 shows an example of the global modification on the original category hierarchy built in section 2.3 and only the category Interest needs modification (the square leaf node) in the global phase. It includes the mapping procedure and candidates generating procedure.

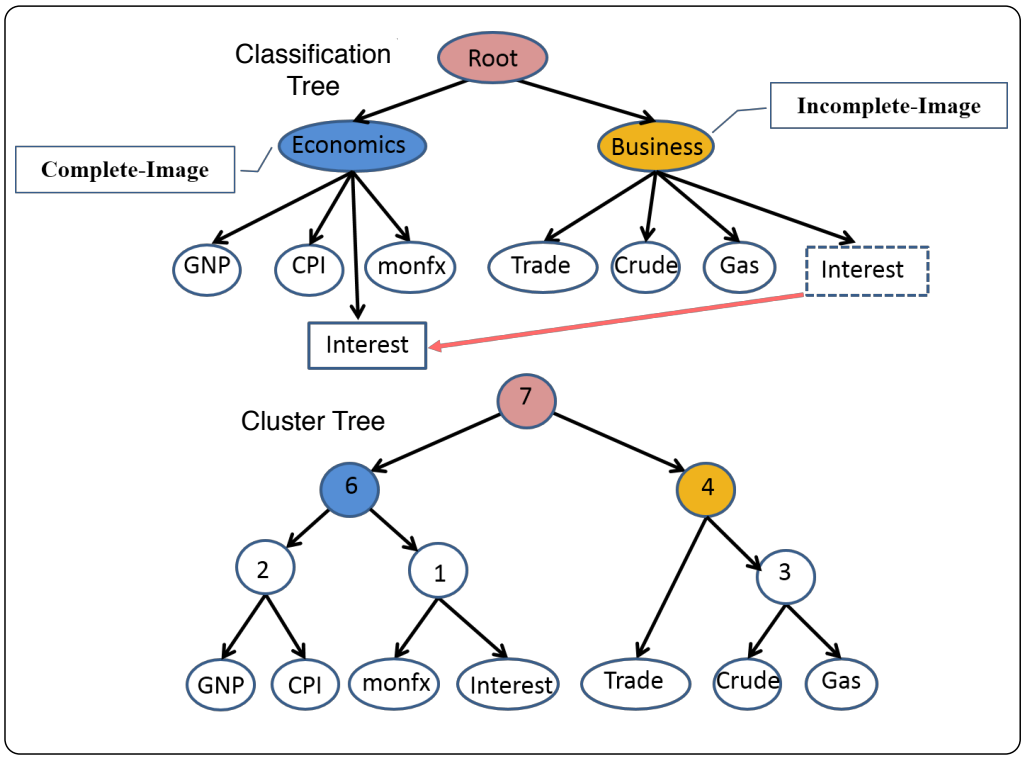


Figure 2.4: The global modification example.

2.4.1.1 Mapping Procedure

To build link between the category hierarchy and the cluster tree, we define two types of mapping Complete-Image and Incomplete-Image and give a new concept of Pattern Consistence based on Complete-Image to reflect whether the category hierarchy has the consistent topical clusters within that cluster tree.

Definition 5. Node with Labels *Given a Category Hierarchy Tree H_c or a Cluster Tree C_T, the Label Set of a Node n in H_c or C_T is defined as follows:*

$$\forall n \in Leaf(H_c) \text{ or } Leaf(C_T), Labels(n) = Cat\_ID \tag{2.1}$$

$$\forall n \in Internal(H_c) \text{ or } Internal(C_T), Labels(n) = \cup_{n^* \in Child(n)} Labels(n^*) \tag{2.2}$$

where $Leaf(X)$ is the set of all leaf nodes in a tree rooted by Node X and $Internal(X)$ is the set of all non-leaf nodes in X, $Labels(n)$ is the label set of Node n.

Using the label set, we define the following concepts.

Definition 6. Complete-Image *Given an internal node n in H_c, if there exists a node n^* in C_T, such that:*

$$Labels(n^*) \supseteq Labels(n) \tag{2.3}$$

$$\nexists n' \in Sub\_node(n^*), Labels(n') \supseteq Labels(n) \tag{2.4}$$

Then there is a Complete-Image mapping between n and n^*.

Definition 7. Incomplete-Image *Given an internal node n of H_c, if there exists a node n^* in C_T and $L(n^*) = Lables(n^*) \cap Labels(n)$, such that:*

$$\forall n' \in C_T, L(n') = Lables(n') \cap Labels(n), |L(n^*)| \geq |L(n')| \tag{2.5}$$

$$\nexists m \in Sub\_node(n^*), Labels(m) \supseteq L(n^*) \tag{2.6}$$

Then there is an Incomplete-Image mapping between n and n^*.

Definition 8. Pattern Consistence *Given a Hierarchy Tree H_c and a Cluster Tree C_T, there is a one-to-one mapping between leaf categories of H_c and C_T. The classification pattern of H_c and the clustering pattern of C_T are satisfied with* Pattern Consistence *under the following two conditions:*

$$\forall n \in Internal(H_c), \exists n^* \in C_T, f(n) = n^* \tag{2.7}$$

$$\forall n_1, n_2 \in Internal(H_c), if\ \ f(n_1) = n_1^*, f(n_2) = n_2^*, then\ n_1^* \neq n_2^* \tag{2.8}$$

where f is the function of Complete-Image.

It ensures that if H_c and C_T are satisfied with *Pattern Consistence*, then for each node in H_c there is a different mapping node in C_T and the mapping type is the Complete-Image.

Mapping is to find a Complete-Image for each node in category hierarchy and if all nodes can be mapped by Complete-Image into the cluster tree, we dont need to modify the hierarchy. If some nodes are mapped by Incomplete-Image, then there are some categories to be adjusted so as to achieve *Pattern Consistence*.

In Figure 2.4, the mapping procedure is a top-down manner to build one-to-one mappings from nodes in a classification tree to nodes in a cluster tree. Pairs of nodes in the same colour between the classification tree and the cluster tree form one-to-one mappings. In this example, the category Economics and node 6 (in blue) form a mapping of Complete-Image, while the category Business and node 4 (in yellow) form a mapping of Incomplete-Image. The category Interest causes the Incomplete-Image between the nodes, thus we add it into the AdjustNodesList and adjust it through candidate generation procedure to satisfies *Pattern Consistence* with the cluster tree.

2.4.1.2 Candidates Generating Procedure

The mapping procedure returns a list of categories that destroy the pattern consistence and should be relocated at more appropriate positions in the category hierarchy. We generate the candidates by testing two modification strategies and accept the best one which improves the classification performance most to update the category hierarchy. The two strategies are defined as follows:

1. Get the nearest neighbour category in the cluster tree and insert the node as a sibling of that neighbour in the classification tree.

2. Get the nearest ancestor that has been mapped in the cluster tree and insert the node as a child of that ancestor in the classification tree.

In the global modification example shown in Figure 2.4, we adjust the category Interest by first finding its nearest neighbour category monfx and then insert Interest as a sibling of monfx according to the second strategy.

2.4.2 Phase 2: Local Adjustments

Global modification is to break up some obviously inappropriate parent-child relations to make the original hierarchy satisfy with the Pattern Consistence of the clustering results. However, the satisfaction of Pattern Consistence cannot guarantee the best expression of classification. For example, a global modification can solve the problems of case 1 and case 3 which are described in section 2.3, but it cannot handle case 2 and case 4. Since different category

hierarchy can all satisfy Pattern Consistence with the same cluster tree, it is necessary to do some localized adjustments on the category hierarchy.

We define three elementary operations to conduct local adjustments:

- Pull-Up: pull up one node to its parents level to be a sibling of its parent.
- Merge: merge two nodes under the same parent into one.
- Split: split a leaf node into finer nodes and add these new nodes as the children of the leaf node.

Local adjustment is achieved by testing the three elementary operations on some specific nodes. With the feedback of the classification results, we can pick up nodes satisfying the following premises: $P \ll \bar{P}$ and $P \gg R$, where P and R represent the classification precision and recall of each category and $\bar{P}$ is the average precision of categories at the same level.

We set trigger conditions for each operation. If a category satisfies the trigger conditions, we test the corresponding operation and compare the new evaluation score with the original one to make a decision whether to accept the operation or not. A new evaluation measure is proposed in section 2.4.3 to judge whether the quality of a hierarchy is improved.

We conduct LDA topic model to make the category associated with a topic distribution that gives a coarse description of the category. LDA is a probabilistic generative model (Blei et al., 2003), where documents are represented as random mixtures over latent topics and a topic is a distribution over words. For each category, we compute the average topic mixtures over documents to get the category-topic distribution (the mean document-topic distribution over documents in the category). We can use the category-topic distribution to represent the inner pattern of categories.

Gibbs sampler (Minka, 2000) is applied to infer the topic distribution and the word distribution. In our experiment, we empirically set the number of topics $K = 100$ and hyper-parameters $\alpha = 50/K$ and $\beta = 0.1$. After obtaining the topic distribution, we can use it to define trigger conditions for Merge, Pull-Up and Split operations.

2.4.2.1 Merge Operation

When performing the Merge operation, we need to detect whether there is another category that is similar to a certain degree with the current one under the same parent. Merge operation is triggered if the category similarity exceeds a threshold value. The key challenges to set the trigger condition for Merge operation include the following two questions:

1. How to measure the similarity between categories?
2. How to set the threshold value?

We define the similarity of two categories using category-topic distribution.

Definition 9. Category Similarity *Given two categories A and B with their topic distributions* $\vec{\theta_A}$ *and* $\vec{\theta_B}$*, the similarity is defined as:*

$$Sim(A, B) = \sum_{k \in K} I_{\theta_{Ak} \neq 0}(\theta_{Ak}) I_{\theta_{Bk} \neq 0}(\theta_{Bk}) \theta_{Ak} log \frac{\theta_{Ak}}{\theta_{Bk}} \tag{2.9}$$

where $I_A(x)$ *is the indicator function, if* $x \in A$, $I_A(x)$ *is equal to 1 else 0.* θ_{Ak} *and* θ_{Bk} *represent the* k^{th} *topic proportion of category A and B. The smaller the value, the more similar the category is. This metric says that two categories similar to each other share a similar combination of topics.*

We show the general Merge procedure in Algorithm 2. Suppose that we pick up category A to check. Then we compute the most similar category to A under the same parent, denoted as category B (line 5-7). The threshold value can be set to the minimum category similarity between B and any other categories under the same parent except A (line 8-9). If the Merge operation can improve the hierarchy, then we accept it (line 10-14).

Algorithm 2: Merge Operation Procedure.

```
   Input: A, Cla_HT
   Output: HT
 1 HT ← Cla_HT ;
 2 O_flag = True ;
 3 Pa ← getParent(A) ;
 4 Eva_Score = evaluateHT(Cla_HT) ;
 5 while O_flag do
 6 |   Clist ← getChildren(Pa) ;
 7 |   foreach node n in Clist do
 8 |   |   B ← arg min Sim(A, n);
 9 |   |   s = Sim(A, B) ;
10 |   end
11 |   foreach node n in Clist do
12 |   |   Threshold = min Sim(n, B) ;
13 |   end
14 |   if s < Threshold then
15 |   |   [A, H_temp] ← Merge(A,B,HT) ;
16 |   |   score = evaluateHT(Pa, H_temp);
17 |   |   if score < Eva_Score then
18 |   |   |   HT ← H_temp;
19 |   |   |   Eva_score = score;
20 |   |   end
21 |   end
22 |   else
23 |   |   O_floag = False ;
24 |   end
25 end
26 return HT ;
```

2.4.2.2 Pull-Up Operation

If a parent node cannot cover the topics of its child category, it should be pulled-up to the upper level in order to avoid the influence from the inappropriate parent node. For each category, we define the $Cover\_Ratio$ for a given parent category A and its child B as the trigger condition.

Definition 10. Cover Ratio *Given a parent category A and its child B with their topic distributions $\vec{\theta_A}$ and $\vec{\theta_B}$, the Cover Ratio is defined as:*

$$Cover\_Ratio(A, B) = \sum_{Topic_k \in Keyset(B)} (log\theta_{Ak} + log\theta_{Bk}) \tag{2.10}$$

where $Keyset(B)$ is the significant topic set consisting of the top-k major topics in category B. θ_{Ak} and θ_{Bk} represent the k^{th} topic proportion of category A and B respectively. If $Cover\_Ratio(A, B)$ exceeds a threshold value, then we say that category A can cover its child category B, otherwise A cant cover B. Pull-Up operation is triggered if category A cant cover its child category B.

Suppose that we pick up category B to check. Category A is B's parent. For Pull-Up operation, the threshold value can be set to the average $Cover\_Ratio$ of all the children under category A with a multiplier $\delta \in [0, 1]$ to control the degree of coverage. Too small δ will overload CPU to test improper Pull-Up operations, while too big δ may lead to missing some necessary Pull-Up modifications on inappropriately located categories. Thus δ is empirically set to 0.7 in our study. There is another way to set δ according to the resource distribution on child category B. In this way, δ is set to the percentage of the number of resources in category B to the number of resources in the parent category A.

As the general procedure of Pull-Up operation is just similar to the Merge operation, we dont give the full algorithm for it.

2.4.2.3 Split Operation

As new resources are increasingly added into the category hierarchy, some of them cant find proper categories and we may put them under less relevant categories. This behaviour will lead to less cohesive categories, especially for leaf categories. When less relevant resources in a leaf category accumulate to a certain degree, we need to split the category into finer sub-categories.

Split is operated when the category cohesion is smaller than a threshold value and the percentage of the number of resources in the category to the number of resources in its parent category is larger than a threshold value (empirically set to 50% in our experiment). For each category, we define the concept of category cohesion.

Definition 11. Category Cohesion *Given a category A with its topic distribution $\vec{\theta_A}$, the Cohesion is defined as:*

$$Coh(A) = \sum_{Topic_i, Topic_j \in Keyset(A)} (log\theta_{Ai} + log\theta_{Aj}) * Dist(i,j) \tag{2.11}$$

where $Keyset(A)$ is the significant topic set of category A. θ_{Ai} and θ_{Aj} represent the i^{th} and j^{th} topic proportion of category A respectively. Since topic is represented by a distribution over words, $Dist(i,j)$ computes the cosine similarity of word distribution between $Topic_i$ and $Topic_j$. The smaller the value of $Coh(A)$, the less cohesive the category A is.

Suppose that we pick up category A to check for Split operation. The threshold value can be set to the average category cohesion of all the categories under the same parent with category A also with a multiplier $\xi \in [0,1]$. ξ is set to the ratio of the number of resources in category A to the number of resources in its parent category in our experiment.

Unlike the other two operations, how to perform the Split operation is a major problem. Clustering algorithms can help partition topics in the significant topic set, but it is still difficult to anticipate a proper number of clusters. A split with neither too few nor too many subcategories is preferable to humans. To solve this problem, we firstly use hierarchical clustering algorithm to generate a binary tree of topics. The average-linkage function defined as the average of all similarities among the topics in both clusters is used to measure the similarity between any pair of clusters. Then we apply Min-Max Partitioning proposed in (Chuang & Chien, 2004) to select the best cutting level that minimizes the criteria function combining the *cluster set quality* and the *cluster number preference*.

Let C be a set of clusters. The cluster set quality $Q(C)$ is calculated as:

$$Q(C) = \frac{1}{|C|} \sum_{C_i \in C} \frac{Sim(C_i, \bar{C_i})}{Sim(C_i, C_i)} \tag{2.12}$$

where $Sim(C_i, \bar{C_i})$ is the inter-similarity between cluster C_i and C_j (j≠i). Let $Sim(C_i, C_j)$ be the average of all pairwise similarities among the topics in C_i and C_j. $Sim(C_i, C_i)$ is the intra-similarity within cluster C_i. Let $Sim(C_i, C_i)$ be the average of all pairwise similarities among topics within C_i. The smaller the value $Q(C)$, the better the quality of the cluster set C is.

The cluster number preference uses a gamma distribution function to measure the degree of preference on the number of clusters at each layer. We change $\alpha!$ into $(\alpha-1)!$ to make this formula reflect the preference cluster number. Let C be a set of clusters. The cluster number preference $N(C)$ is calculated as:

$$\begin{aligned} f(x) &= \frac{1}{(\alpha-1)!\beta^{\alpha}} x^{\alpha-1} e^{-x/\beta} \\ N(C) &= f(|C|) \end{aligned} \tag{2.13}$$

where α and β are two parameters to tune the smoothness of the preference function and they are empirically set as $\alpha = 3$ and $\beta = N_{clus}/2$. N_{clus} is the expected number of clusters and in our experiment it is empirically set to the square root of the number of topics in the significant topic set.

The best cutting level l should minimize the criteria function of $(Q(C(l)))/(N(C(l)))$, where $C(l)$ is the set of clusters produced by cutting level l on the hierarchical clustering binary tree.

The Split operation uses generated clusters on the best cutting level as new finer subcategories and uses the top-k ranked keywords of the topic nearest to the centroid of the cluster to re-label the new category.

2.4.3 Evaluation Measure

To evaluate the quality of a hierarchy, we propose Uncertainty Score that combines structural aspect and classification aspect to judge whether a hierarchy is comprehensive to use. Previous studies on hierarchy generation and hierarchy maintenance mainly use F-Measure (Yuan et al., 2012), macro-averaged recall (Tang et al., 2006) or classification accuracy (Y. Yang & Liu, 1999) to guide the hierarchy evolvement. However, all these traditional measures only aim to judge the performance of classification algorithms instead of the hierarchy itself.

An evaluation approach to judging the quality of a hierarchy proposed in (Chuang & Chien, 2004) lists several qualitative measures including:

1. *Cohesiveness,* which is for judging whether the instances in each category are semantically similar.

2. *Isolation,* which is for judging whether categories under the same parent are discriminative from each other.

3. *Hierarchy,* which is for judging whether hierarchical categories go more and more specific from top to bottom with different comprehensive abstraction levels.

4. *Navigation Balance,* which is for judging whether the number of child categories for each internal category is appropriate.

5. *Readability,* which is for judging whether the concepts represented by each category are easy to understand.

Each measure can be assigned numerical scores by humans to reflect the satisfactory degree. However, there is no united calculation form of these measures, thus they can only be judged in an isolated way. For the hierarchy maintenance task, we need an evaluation measure for hierarchies that can be automatically computed in a clear united form. That is why it is necessary to propose Uncertainty Score in this thesis for automatic hierarchy maintenance.

A good hierarchy is expected to classify resources into each category not only with high classification accuracy but also with a relatively high certainty at each level. The larger the certainty is, the less ambiguity of classification semantics the hierarchy has. Besides the classification aspect, an appropriate hierarchy should try to keep navigation balance among all branches and to avoid heavily leaning on one side. Furthermore, we should also consider whether resources are evenly distributed to the categories of the same level, which is beneficial to user retrieval.

Uncertainty Score (UC_Score) uses the Entropy to measure the classification uncertainty, the balance of the hierarchical structure and the uniformity of resources distribution. Entropy is an effective and widely-adopted measure of the uncertainty for a random variable in the field of information theory (Shannon, 1948). The three aspects of a hierarchy in fact measure the uncertainty for classification, structure and distribution and that is why we name the evaluation measure UC_Score (UC is short for uncertainty).

Therefore, we define UC_Score to evaluate the quality of a hierarchy by considering three aspects of a hierarchy: the classification uncertainty represented by H_c, the structural balance represented by H_s, and the resource distribution represented by H_r.

The UC_Score of a tree-structured hierarchy rooted by node n can be recursively calculated level by level in a top-down manner. As shown in Figure 2.5, each node in the hierarchy is associated with three values represented by UC_Score, CH_UC and Eva. The UC_Score value is a final evaluation value of the hierarchy rooted by the node. The CH_UC value is an average of UC_Score over all the children nodes. The Eva value is an evaluation value only related to the current node instead of the hierarchy. In Figure 2.5, the UC_Score value of a node includes two parts. One is its own Eva value and the other is the CH_UC value.

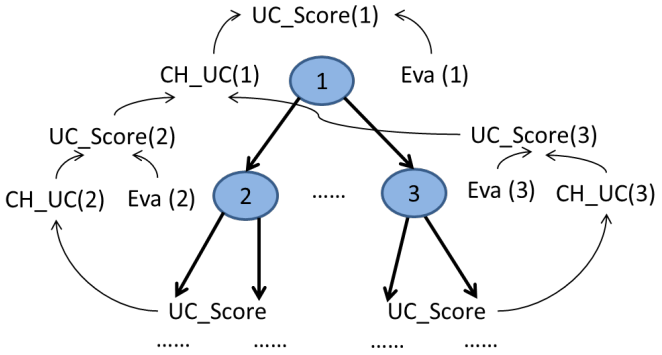


Figure 2.5: UC_Score calculation example.

The calculation of UC_Score of node n is defined as:

$$UC\_Score(n) = \begin{cases} \frac{1}{L} \times \{Eva(n) + \gamma \times CH\_UC(n)\}, & non-leaf node; \\ 0, & leaf node. \end{cases} \tag{2.14}$$

The UC_Score of non-leaf node n includes its own Eva value and the CH_UC value (average UC_Score over its children nodes) with a discount factor γ. L is the number of levels. The discount factor γ is to control the degree of effect on the final UC_Score from categories

on different levels. It is empirically set to 0.8. The discount factor for each level will have an accumulated effect when going down the hierarchy. The lower the level, the less effect it will have on the final UC_Score of a hierarchy.

The calculation of CH_UC of node n is defined as:

$$CH\_UC(n) = \frac{1}{m} \sum_{n^* \in Child(n)} UC\_Score(n^*) \tag{2.15}$$

The CH_UC value of node n is an average UC_Score over all children nodes. In the formula, n^* is the child node of n. $Child(n)$ is a set of children nodes of n. m is the size of $Child(n)$.

The calculation of Eva value of node n is defined as:

$$Eva(n) = \frac{H_c}{\alpha H_s + (1-\alpha) H_r} \tag{2.16}$$

The Eva value of node n is computed by combining three variables of the current node n: the classification uncertainty H_c, the structural balance H_s and the resource distribution H_r. The three aspects will be detailed respectively. α is a balance factor between H_s and H_r, and is set to 0.5 empirically in our study.

2.4.3.1 Classification Uncertainty

The classification uncertainty of a hierarchy reflects the ability to express classification semantics. A preferable category hierarchy is expected to contain categories with maximum intra-category similarity and inter-category discrimination. In other words, resources within a category should be semantically similar and resources from different categories should be discriminative from each other. Category hierarchies satisfying these two characteristics can express clear classification semantics.

When performing classification, a preferable hierarchy is expected to classify resources into each category not only with high classification accuracy but also with a relatively high certainty at each level. The larger the certainty is, the less ambiguity of classification semantics the category hierarchy has.

The resources classification uncertainty is represented by H_c. For each resource r, we get a probability distribution $p_{r_1}, p_{r_2}, \cdots, p_{r_m}$ with which it is classified into m child categories of node n. We compute the entropy of this probability distribution divided by the max entropy to make the value fall into the interval $[0, 1]$. The max entropy is calculated by classifying the resources into m categories with the same probability of $1/m$.

Thus, the calculation of H_c is defined as:

$$H_c = \frac{1}{R} \times \sum_{r=1}^{R} \frac{H(p_{r_1}, p_{r_2}, \cdots, p_{r_m})}{H\left(\frac{1}{m}, \cdots, \frac{1}{m}\right)} \tag{2.17}$$

R is the number of resources in the category of node n. m is the number of child nodes of n. $H(\cdot)$ represents the entropy of the parameters and the parameters must be a probability

distribution. p_{r_i} is the probability of the r^{th} resource classified to the i^{th} child category of node n.

2.4.3.2 Structural Balance

Structural balance is important for user navigation to category hierarchy. A well-structured hierarchy should keep appropriate number of child categories for each internal category.

The balance of the hierarchical structure is represented by H_s. We compute the entropy of a probability distribution with which the number of leaf categories assigned to each child node. To make the value fall into the interval $[0, 1]$, it should be divided by the max entropy that is calculated by offering each child node with equal number of leaf categories.

The calculation of H_s is defined as:

$$H_s = \frac{H\left(\frac{C_1}{C}, \frac{C_2}{C}, \cdots, \frac{C_m}{C}\right)}{H\left(\frac{1}{m}, \frac{1}{m}, \cdots, \frac{1}{m}\right)} \tag{2.18}$$

C is the number of leaf categories assigned to node n. m is the number of child nodes of n. C_i is the number of leaf categories assigned to the i^{th} child node of n. $H(\cdot)$ represents the entropy of the parameters and the parameters must be a probability distribution.

2.4.3.3 Resource Distribution

It is beneficial to user retrieval if resources are evenly distributed to categories in a hierarchy. So we consider it as an aspect of the evaluation measure of a hierarchy.

Whether resources are evenly distributed or not is represented by H_r. We calculate the entropy of a probability distribution with which the number of resources assigned to each child node. It should also be divided by the max entropy to make the value fall into the interval $[0, 1]$.

The calculation of H_r is defined as:

$$H_r = \frac{H\left(\frac{R_1}{R}, \frac{R_2}{R}, \cdots, \frac{R_m}{R}\right)}{H\left(\frac{1}{m}, \frac{1}{m}, \cdots, \frac{1}{m}\right)} \tag{2.19}$$

R is the number of resources assigned to node n. m is the number of child nodes of n. R_i is the number of resources assigned to the i^{th} child node of n. $H(\cdot)$ represents the entropy of the parameters and the parameters must satisfy the constraints of being a probability distribution.

2.5 Experiment and Results

2.5.1 Datasets

We use Reuters-21578, 20Newsgroups and DMOZ (Open Directory Project) datasets in our experiments, which are standard datasets for data classification.

Reuters-21578 data set contains documents collected from 135 categories mainly related to economy. We construct a subset from the original dataset. Reuters-25 includes 25 categories

among the 135 topics after removing categories that has less than 10 documents in the training set and test set. For each category, we just retain documents with a single label.

20Newsgroup has about 20,000 articles evenly divided into among 20 categories. We use the "Bydate" version for a standard train/test split.

DMOZ dataset is the largest human-edited directory on the Web with over 5,169,995 sites listed in over 1,017,500 categories. We just extract a meaningful 3-level hierarchy from the original one, including 8 top categories from the total 16 ones in DMOZ taxonomy, including Arts, Business, Computers, Health, Games, Recreation, Science and Sports. Under these categories, we choose 188 categories within the three levels as our hierarchy. After data collecting and cleaning, we remain 46,636 documents.

Table 2.3: Information of datasets

DataSet	Number of Categories			Number of Documents	
	Lev 1	Lev 2	Lev 3	Train	Test
Reuters-25	7	25	N/A	2760	994
20 Newsgroup	7	20	N/A	11293	7061
DMOZ	8	59	121	32654	13982

The general characteristics of our experiment datasets are summarized in Table 2.3, from which we can find that the smallest data set Reuters-25 just contains 3,754 documents and the largest data set DMOZ contains 46,636 documents. The total number of leaf categories varieties from 25 to 121.

All the datasets are attached with an original coarse hierarchy dividing the topics into several groups of similar classification semantics. They are used as the initial hierarchy by our *AMHC* approach.

To pre-process the datasets, we remove the stop words with stop word list and prune words occurring less than 5 times and less than 3 documents across the corpus and perform the stemming operations with Porter Stemmer. For feature extraction (Salton & Buckley, 1988), we select the top 1000 words by the information gain, which is frequently used as a basic feature-goodness criterion in the field of data mining. It measures the number of bits of information obtained for category prediction by knowing the presence or the absence of a term (feature) in a document.

2.5.2 Hierarchies

There are four types of hierarchies in our experiments, listed as follows:

- Baseline Hierarchy 1: Original Hierarchy (OH). This topic hierarchy is attached to each dataset dividing the topics into several groups of similar classification semantics. However, it has many inconsistencies with resources.

- Baseline Hierarchy 2: Automatically Generated Hierarchy (AH). This hierarchy is generated by the approach HAC+P proposed by (Chuang & Chien, 2004).

- Modified Hierarchy 1: Modified Original Hierarchy (M_OH). This hierarchy is modified from the original hierarchy by our AMHC approach.

- Modified Hierarchy 2: Modified Automatically Generated Hierarchy (M_AH). This hierarchy is modified from the automatically generated hierarchy by our AMHC approach.

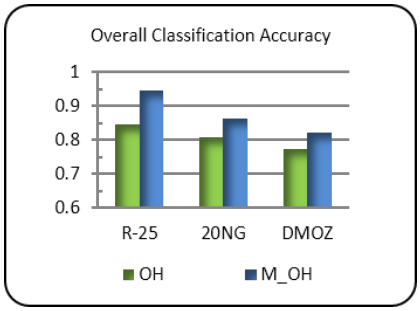
2.5.3 Evaluation Results

To investigate the effectiveness of our AMHC approach, we conduct two groups of comparison experiments. One group is on the original hierarchy (OH) and its modified hierarchy (M_Oh). The other group is between the automatic generated hierarchy (AH) and the AMHC modified hierarchy (M_AH). In our experiments, LibSVM (Chang & Lin, 2011) is used as the base classifier to implement the standard hierarchical SVM (T.-Y. Liu et al., 2005). We used all the default settings, including the radial basis function kernel. We get a validation set by splitting the training set into two small subsets (70% for training and 30% for validation). JGibbLDA (Phan & Nguyen, 2007) is applied for LDA topic modelling.

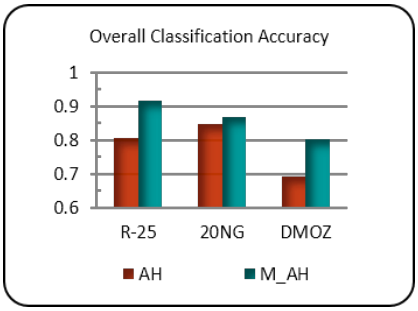
We use classification accuracy as the overall evaluation measure, which equals the proportion of correctly classified instances. It is more suitable to evaluate multi-class classification tasks than F1-Measure, Precision and Recall (Sun & Lim, 2001), since those measures are defined for a specific category. However, the classification accuracy cant reflect the classification performance on each category, so we also list Macro-F1 and show some categories F1-Measure to explain the overall improvements brought by hierarchy evolvements. We also calculate UC_Score with $\alpha = 0.5$ and $\gamma = 0.8$.

Figure 2.6 consists of 6 figures comparing the classification performance and the hierarchy quality on different hierarchies in terms of the three measures: classification accuracy, Macro-F1 and UC_Score.

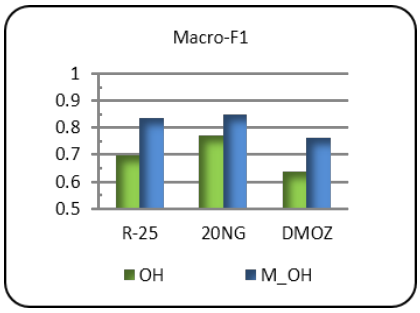
Figure 2.7 shows the 5 categories' F1-Measure that improved most by M_OH in Reuters-25 dataset. Category Money-sy increases mostly by 12.7%. In OH, almost all documents in Money-sy are misclassified into Money-fx. Money-fx and Interest are less distinguishable, however, in M_OH we group Money-fx and Interest to enhance their common features and it can also enable easier discrimination of Money-sy. At a lower level we use more specific features to separate Money-fx and Interest, increasing 4.8% and 6.8% respectively. For Livestock (9.4%) and Jobs (7.2%) we adjust them in the first phase by the cross-branch movements to place them under more suitable parents that can better reflect their classification features. This is why we can get the overall improvement of 12.1% on classification accuracy (Figure 2.6-a) and 19.8% on Macro-F1 (Figure 2.6-c) with M_OH.



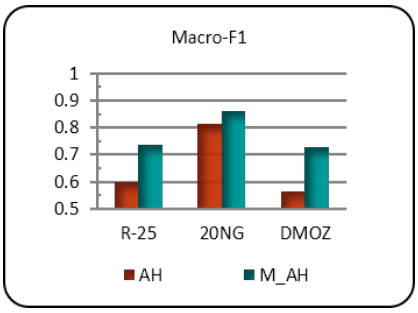
(a) Comparison of classification accuracy between OH and M_OH



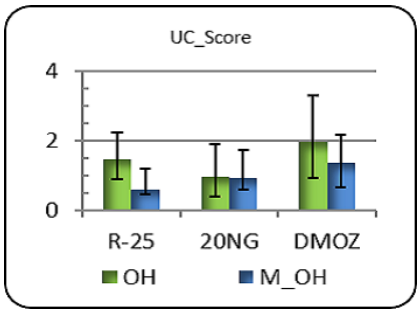
(b) Comparison of classification accuracy between AH and M_AH



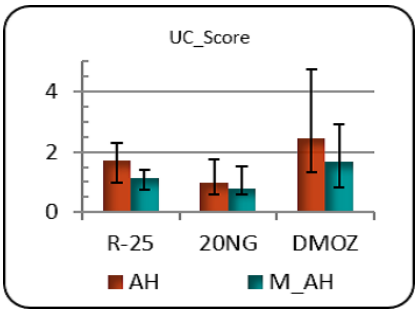
(c) Comparison of Macro-F1 between OH and M_OH



(d) Comparison of Macro-F1 between AH and M_AH



(e) Comparison of UC_Score between OH and M_OH



(f) Comparison of UC_Score between AH and M_AH

Figure 2.6: Comparisons on classification performance between category hierarchies.

20Newsgroup achieves almost the same results on AH and M_AH around 85% of the classification accuracy (Figure 2.6-b), which outperform their counterparts (OH and M_OH), since auto-generated hierarchy clusters alt.atheism and talk.region.misc whose resources are more similar. In M_OH, we can still observe improvements of 6.5% on classification accuracy (Figure 2.6-a) and 10.5% on Macro-F1 (Figure 2.6-c). Because sci.crypt and soc.religion.christian are rearranged into their more related parent and cluster, this change directly contributes to the improvement.

The ODP category hierarchy is human-edited and its original hierarchy is already a good one to express clear classification, reaching 77.5% of classification accuracy (Figure 2.6-a) compared with 69.4% on AH (Figure 2.6-b). This also shows the inadequate power of HAC+P in generating large taxonomies with wider range of topics. However, an improvement of accuracy (15.7%) is achieved on M_AH, reaching 80.3%, which is almost the same as that on M_OH (82.1%).

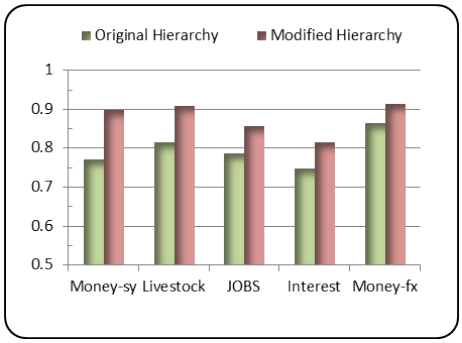


Figure 2.7: The most improved 5 categories' F1-Measures on Reuters-25 dataset.

This indicates that AMHC approach can reach a satisfactory hierarchy no matter how terrible conditions the initial hierarchy has.

Compared with classification accuracy and Macro-F1, *UC_Score* has opposite tendency that the smaller the value, the better the hierarchy is, but it reflects consistent results with the other two evaluation measures. In Figure 2.6-e and Figure 2.6-f, we show the min/max *UC_Scores* (error bars) of different levels for each hierarchy and the final *UC_Score* of the whole hierarchy. The shorter the bar is, the more consistent quality evaluations of different levels of a hierarchy has. In terms of *UC_Score*, the hierarchy M_OH on Reuters-25 (Figure 2.6-e) and the hierarchy M_AH on DMOZ (Figure 2.6-f) have the largest improvement with 60.3% and 40.0% respectively. In addition, *UC_Score* is more sensitive when detecting bad evolvement of a hierarchy. For example, it can abort the Merge operation of Business and Economics in Reuters-25, Recreation and Sports in DMOZ on the first level of the hierarchy, which will result in a heavily skewed tree structure in spite of an increasing in F1-Measure. *UC_Score* falls into a larger value range $[0, +\infty)$ and considers more aspects of a hierarchy. That is why it can show more reliable and effective results.

2.6 Case Study

This section conducts a case study which applies our AMHC approach to modifying ScienceDirect category hierarchy so as to investigate the effectiveness of AMHC on scientific literature resources.

In this case study, we collected scientific articles under the category Physical Sciences and Engineering from ScienceDirect databases, covering almost 13 branch subjects in the field of computer science, physical science and material science. The corpus contains 3780 articles in computer science, 1267 physical papers and 967 papers in material science. Figure 2.8 shows the original ScienceDirect category hierarchy on the 13 categories, where the number represents the distribution of scientific articles on each category. Through parsing the XML file, we extract title, abstract and full-text for each article and generate a 180MB TXT file as the final corpus. The final corpus consists of 6014 scientific articles, containing 26,783,935 words in total and the vocabulary size is 263,192. The corpus is randomly split into two parts, 70% as a training

set with 4217 papers and 30% as a test set with 1797 papers. LibSVM (Chang & Lin, 2011) is used with the default settings as the base classifier to implement the standard hierarchical SVM (T.-Y. Liu et al., 2005).

ScienceDirect category hierarchy is too general to provide fine classification for some specific subjects, thus we apply our AMHC approach to modifying the original category hierarchy and the modified hierarchy is shown in Figure 2.8. Specifically, the modifications include: (1) Pull-Up operations: Pull up category Mathematics and category Statistical and Nonlinear Physics to the upper level; (2) Merge operations: Merge category Artificial Intelligence and category Computer Vision and Pattern Recognition, category Materials Chemistry and category Nanotechnology, and generate two "Temporary Node" named by automatic mention suggestion; (3) Split operations: Split the category Artificial Intelligence into two subcategories, one is Natural Language Processing represented by a keyword set {sentence, semantic, syntactic, $\cdots$} and the other is Knowledge Representation and Reasoning represented by a keyword set {logics, bayesian, reasoning, $\cdots$}; Split the category Computer Vision into two subcategories, one is Image Processing represented by a keyword set {image, object, convolutional, $\cdots$} and the other is Learning Algorithm represented by a keyword set {classification, learning, supervised, $\cdots$}. The modified categories are consistence with the 2012 ACM computing classification system.

We compare the classification performance between the original category hierarchy and the modified category hierarchy. The classification accuracy and F1-Measure are used to evaluate the overall classification performance and the individual category performance. The overall classification accuracy reaches 70.8% on the modified category hierarchy, which achieves an improvement of 13.5%, compared with 62.4% on the original ScienceDirect hierarchy. The most improved categories are Mathematics and Statistical and Nonlinear Physics, whose F1-Measure increases by 7.5% and 6.2% respectively. In addition, merging Artificial Intelligence and Computer Vision and Pattern Recognition enforces the expression of their common features, thus contributing to an increase of F1-Measures by 4.9% and 3.6%.

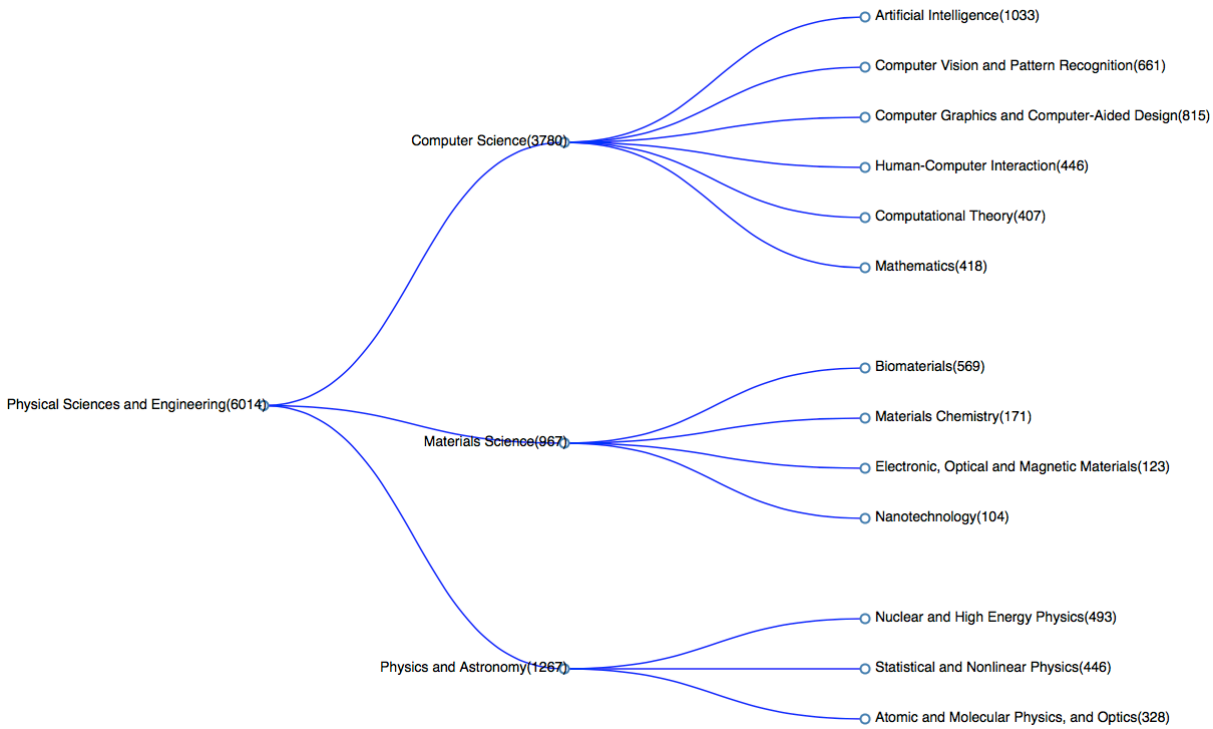


Figure 2.8: The original ScienceDirect category hierarchy.

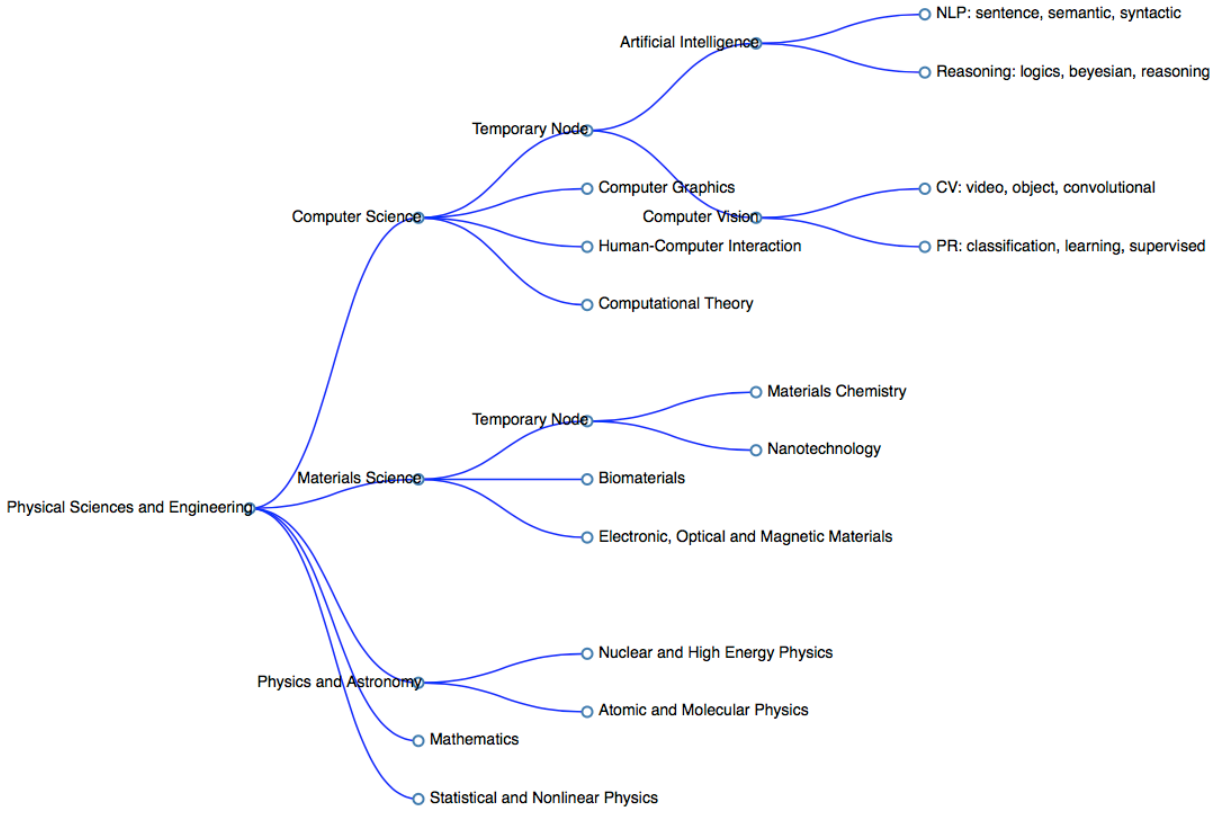


Figure 2.9: The modified ScienceDirect category hierarchy.

3 Micro-dimension Construction: Automatic Generation of Scientific Concept Hierarchy

3.1 Overview of the Problem

The scientific resource space consists of three micro dimensions: Task, Process and Material, corresponding to the three intrinsic properties of scientific literature resources as shown in Figure 3.1. The task dimension describes research problems a paper trying to address. The process dimension describes methodologies or devices that a paper studies or utilizes. The material dimension describes corpora or materials in a scientific paper. The three micro dimensions in a scientific resource space cover most of typical questions that researchers care most, for example, which paper address which task, use which method and test on which dataset.

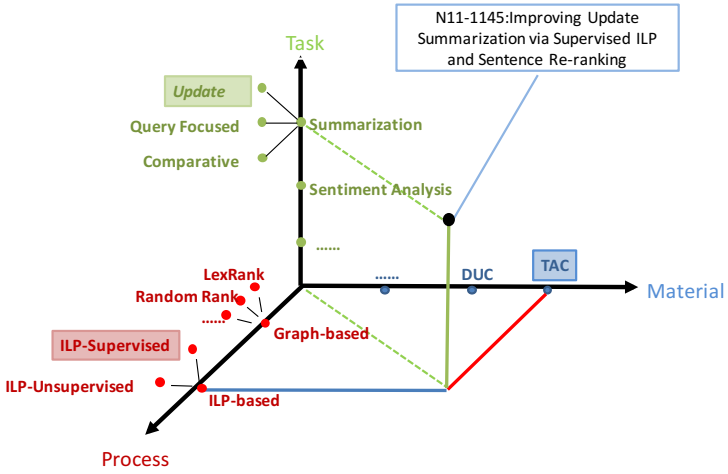


Figure 3.1: A micro-dimension space example.

The construction of micro dimensions in a scientific resource space is in fact to extract the intrinsic properties from unstructured scientific texts and build concept hierarchies in each micro dimension respectively. The extraction of scientific intrinsic properties needs semantic parsing on scientific articles, including recognizing the three basic types of entities and extract Hyponym-of and Synonym-of relations between entities, in order to generate the hierarchical coordinate system in each micro dimension. The Hyponym-of relation and Synonym-of relation are fundamentally used in the construction of ontology, knowledge base and knowledge graph.

In a scientific resource space, macro dimensions help users retrieve scientific articles according to the document-level category information through text classification, while micro dimensions provide users more sophisticated retrieval service through mining smaller text units (sentences

or phrases) to get fine-grained entity information. Thus phrase-level entity recognition is a major research problem addressed in this section.

The micro dimensions based on the content analysis of scientific documents have potentials to improve scientific information retrieval in the following aspects:

(1) Enrich the query diversity: Scientific resource space enables diverse query descriptions by making use of micro dimensions and concept hierarchies. For example, combining Task, Process and Material dimensions could generate query statements like "Apply method X to address problem Y and test on dataset Z" to retrieve scientific articles that utilize method X to solve task Y and use dataset Z in experiments; or combining Task and Process dimensions and using the concept hierarchy on Process dimension could flexibly generate query statements like "Apply variants of method X to address problem Y" to retrieve scientific papers that utilize a variant of X to solve problem Y; or combining Task and Process dimensions but using the concept hierarchy on Task dimension could generate query statements like "Apply method X to address sub-problem of Y" to retrieve papers that utilize method X to solve related sub-tasks of Y. The micro dimensions in a scientific resource space enrich the diversity of query statements and thus enhance the information retrieval service in a scientific resource space.

(2) Provide accurate query description: In a scientific resource space, the micro dimensions provide fine-grained semantic descriptions for the contents of scientific articles, thus it could enable more accurate query descriptions. For example, imagine a new PhD. student who wants to study a particular task of summarization which aims to detect and summarize novel information in a document set under the assumption that users have already learnt another related document set. This particular task is a subtask of text summarization called update summarization, but unfortunately the immature student is ignorant of this terminology. In such situation, it is difficult for the user to describe an accurate query statement to retrieve scientific papers on this task. In a scientific resource space, the macro category dimension could only direct the user to search in the category of text summarization, while the micro task dimension will explicitly guide the user to the update summarization along the concept hierarchy. Figure 3.1 shows an example micro-dimensional space for text summarization, where users could directly retrieve scientific papers that apply ILP-supervised method to address update summarization task. Micro dimensions provide fine-grained query descriptions that help return accurate retrieval results and thus improve the information retrieval service in a scientific resource space.

3.2 Related Work

This section will discuss the related work from four aspects: scientific discourse analysis, entity recognition, relation extraction and concept hierarchy generation.

3.2.1 Scientific Discourse Analysis

Scientific discourse analysis is based on the conventions in scientific writing. Some exist at the word-level as standard scientific expressions, such as a preference for deverbal nominalisations and the passive voice. Some are sentence-level conventions that use lexical or phrasal features to express different argumentative functions of fixed rhetorical expectations and organize several sentences sharing a same rhetorical function into a text block (or a zone). Others exist at the section-level as traditions in paper organisation, for example, a regular paper in a computing linguistics conference usually consists of introduction, related work, methodology, experiments and conclusion. Current studies on scientific discourse analysis mainly focus on sentence-level conventions and develop Rhetorical Structure Theory (RST) and Zone Analysis to analyse the structure and content in scientific documents.

Rhetorical structure theory (Mann & Thompson, 1988; Marcu, 2000) captures local rhetorical relations (Contrast, Antithesis, Concession, etc.) between segments of coherent texts and constructs a hierarchical discourse tree with the rhetorical relations to reveal the text organization. Zone analysis is a theory about the categorization of sentences according to global rhetorical functions in scientific articles, which is widely used in the structure analysis of scientific articles. The scientific discourse analysis based on the theory of zone analysis that is to annotate sentences with different rhetorical functions has a close relationship with the scientific semantic parsing in this chapter, so this section will discuss zone analysis in details. Previous researches on zone analysis design different rhetorical function annotation schemas according to research objectives and focus aspects.

Simone Teufel first proposed argumentative zoning (AZ) (Teufel, Carletta, & Moens, 1999; Teufel & Moens, 2002), an annotation schema that creates categories based on the ownership of knowledge claims (KC) and classify sentences into seven categories according to rhetorical status: Aim, Background, Basis, Contrast, Other, Own and Textual, where Aim states the research goal; Background introduces general background knowledge; Basis describes existing KC that provides basis for new KC; Contrast is an existing KC that is contrasted, compared, or presented as weak; Other is a description of other existing KC; Own describes any other aspect of new KC and Textual indicates papers textual structure. Bayesian classifier is applied to classify sentences in annotated corpus of computer linguistics papers.

Later Teufel et al. modified AZ model and created AZ-II with 15 finer grained categories in a two-level hierarchy, and tested it on chemistry articles (Teufel, Siddharthan, & Batchelor, 2009). Other work refined AZ model and exploited it to analyse the structure information in constrained scope of scientific articles, that usually in abstract section (J. Lin, Karakos, Demner-Fushman, & Khudanpur, 2006; Ruch et al., 2007). At the meantime, a separate line of work investigated the performance of various classification models on scientific sentence categorization based on AZ, such as Conditional Random Field (CRF) (Hirohata, Okazaki, Ananiadou,

& Ishizuka, 2008) and Maximum Entropy Markov Model (MEMM) (Teufel & Kan, 2011). In addition, AZ has also been tested to annotate smaller text units, for example, De Waard et al. developed an annotation at the clause level (de Waard, Buitelaar, & Eigner, 2009); Nawaz et al. (Nawaz, Thompson, McNaught, & Ananiadou, 2010) and Thompson et al. (Thompson, Nawaz, McNaught, & Ananiadou, 2011) proposed a multi-dimensional schema to annotate biological events in scientific papers. However, there is no consensus on the optimal text unit in studies of scientific discourse analysis.

Apart from AZ model, there is another important zone analysis model called core scientific concept (CoreSC) (Liakata, Teufel, Siddharthan, & Batchelor, 2010; Liakata, Saha, Dobnik, Batchelor, & Rebholz-Schuhmann, 2012), an annotation schema to classify sentences based on hierarchically-organized scientific concepts. The first level of the hierarchy consists of 11 categories corresponding to the main structure of scientific papers, including Hypothesis, Motivation, Background, Goal, Object, Method, Experiment, Model, Observation, Result and Conclusion. The categories on the second level describe the properties of the concepts, for example, the novelty (e.g. New or Old) of a Method. In the paper, conditional random field (*CRF*) and support vector machine (*SVM*) are used as the classifiers to classify sentences in annotated papers. (Ravenscroft, Oellrich, Saha, & Liakata, 2016) proposed a multi-label annotation task based on CoreSC and public a text corpus in the domain of cancer risk assessment (*CRA*) called Multi-CoreSC CRA corpus. They showed classification improvements in the recognition of CoreSC on this new corpus.

Argumentative zoning (*AZ*) and core scientific concept (*CoreSC*) are theories on the categorization of sentences regarding to scientific discourse analysis, however, they are different in the goals of the annotation schemas. Argumentative zoning focuses on the ownership of knowledge claims thus the categories clearly distinguish the new KC proposed by authors and existing KC proposed by others. CoreSC theory emphasizes on the recognition of core concepts in scientific papers and the categories are suitable to capture structural information in scientific papers, that is Problem - Methedology - Experiment - Result. Both argumentative zoning and CoreSC are theories on the categorization of sentences, thus fail to analyse on smaller text units (e.g. phrases) to extract scientific entities to build micro dimensions in scientific resource space.

3.2.2 Entity Recognition and Relation Extraction

Entity recognition and relation extraction are two fundamental tasks in natural language processing. This section will review some typical and important work in the two tasks.

3.2.2.1 Entity Recognition

The scientific entity recognition can be regarded as a special type of entity recognition task on scientific papers. Related work includes the investigation on the general entity recognition

task in NLP.

Traditional entity recognition approaches mainly rely on the statistical machine learning, which uses annotated corpus to train models and then predict entities on new unknown documents by the model. Entity recognition is typically formulated as a sequence labelling problem using *BIO* schema or *BILOU* schema, where B- prefix represents the beginning of an entity, I- indicates the current token is inside of an entity and O indicates that the current token belongs to none of entities. In *BILOU* schema, B-, I-, L- indicates the beginning, inside, last token of a multi-token entity while U represents a unit-length entity that is differentiated from multi-token entities. Most existing sequence labelling models are based on statistical machine learning, which include Hidden Markov Model (*HMM*), Maximum Entropy Markov Model (*MEMM*) and Conditional Random Fields (*CRF*).

Hidden Markov Models are generative probabilistic models, which compute a joint probability over paired observation and label sequences. It is not practical to enumerate all possible observation sequences for most tasks to calculate a joint probability. Moreover, HMMs require each atomic element in observation sequences to be independent of each other, however, real observation sequences usually contain multiple interacting features or long-range dependencies between elements in the observation sequences. These difficulties motivate the development of alternative conditional models, such as MEMMs and discriminative Markov Models, which compute the conditional probabilities of possible label sequences given an observation sequence. Therefore, conditional models could save modelling efforts on observations and the calculation of conditional probabilities of label sequences could depend on non-independent features of the observation sequence. MEMMs are discriminative probabilistic models, where each state has an exponential model that takes the observation features as input and outputs a probability distribution over next states. Due to the local computation of states, MEMMs generally share a weakness of label bias problem.

CRF (Lafferty, McCallum, & Pereira, 2001) is a discriminative model which provides a sequence labelling framework for entity recognition. CRF attains both advantages from generative models and discriminative models. Specifically, it avoids enumerating all possible observation sequences to compute joint possibility in generative models and relax the very strict independent assumptions on observations to achieve tractability. Meanwhile, it solves the typical label bias problem that suffered by most discriminative models. It has been shown that CRFs perform better than HMMs and MEMMs on multiple sequence labelling tasks, such as named entity recognition, part-of-speech tagging in NLP.

Traditional approaches based on statistical machine learning heavily rely on complicated feature engineering and domain-specific knowledge to design effective features to train a supervised model on small valuable annotated corpus. Recently, deep learning approaches have become popular due to its end-to-end learning power to enable automatic feature learning pro-

cess, which have been employed to produce promising results on large variety of NLP tasks, such as named entity recognition, part-of-speech tagging, language model and speech recognition.

Collobert et al. proposed a unified neural network architecture (Collobert et al., 2011) and applied it to multiple basic natural language processing tasks including part-of-speech tagging (*POS*), named entity recognition (*NER*), chunking and semantic role labelling (*SRL*). This architecture avoids task-specific feature engineering and learns representations from large amounts of unlabelled data. In the paper, convolutional neural network (*CNN*) is employed to solve the common variable-length sequences problem for sequence labelling tasks, which consists of a general convolutional layer and a max pooling layer to extract sentence-level features. For sequence labelling tasks like NER or SRL, there always exist dependencies between tags in a sequence, for example there is no possibility for I-PER following B-LOC in NER task. To tackle this problem, this paper proposed a sentence level log-likelihood scoring in training, which takes account not only the tag probability for all words in a sentence but also the transition probability from one tag to another tag. Good performance and minimal computation requirements are achieved by the system under this architecture on all four tasks.

There are studies exploiting multiple variations of recurrent neural networks (*RNNs*) on sequence labelling tasks, including Long-short term memory network (*LSTM*), bidirectional-LSTM, LSTM+CRF, bidirectional LSTM+CRF (Huang, Xu, & Yu, 2015). Experimental results show that the architecture of bidirectional LSTMs outperforms other neural networks in terms of expressing the global features of sentences and achieves higher accuracy in entity recognition tasks. However, RNNs and CNNs both fail to capture the compositionality of natural language, thus recursive neural network is proposed to learn compositional vector representations for phrases and sentences by syntactic parsing (Socher, Huval, Manning, & Ng, 2012). Recursive neural network is applied to entity recognition and relation extraction by learning compositional vector representation for each node in a constituency tree and then predicting based on these representations (Khashabi, 2013). The tree-structured neural networks have the capacity to make full advantage of the compositional information of natural language and have been proved effective on most of NLP tasks. In addition to the construction of standard neural networks based on word-level embeddings, other works explored character-level vector representations to solve out-of-vocabulary (OOV) problems and achieved state-of-the-art results on named entity recognition (Chiu & Nichols, 2015) and multilingual language processing (Gillick, Brunk, Vinyals, & Subramanya, 2015).

3.2.2.2 Relation Extraction

Relation extraction is another fundamental task in the field of natural language processing, which plays an important role in various tasks, such as information extraction, question answering, machine translation and ontology construction. The goal of relation extraction is to

identify the semantic relation between pairs of annotated entities in given documents, that is, relation extraction is built on the basis of entity recognition. According to whether the extraction needs labelled documents for training process, relation extraction approaches can be classified as supervised relation classification (Kambhatla, 2004; GuoDong, Jian, Jie, & Min, 2005; Tratz & Hovy, 2010) and unsupervised clustering methods (Hasegawa, Sekine, & Grishman, 2004; Chen, Ji, Tan, & Niu, 2005). Currently, supervised relation classification between entities has been fully studied and achieved better results, therefore in this chapter we focus on supervised relation extraction methods and introduce some important work.

Traditional supervised relation classification approaches can be classified as feature-based relation classification and kernel-based relation classification. Feature-based relation classification approaches heavily rely on different sets of features extracted from sentences to train a classifier (e.g. logistic regression) to predicate the relationship between pairs of entities. Generally, three types of features are proved effective in relation classification. Lexical features concentrate on the given entities, including word, lemma and part-of-speech of the entity token and its surrounding tokens. Syntactic features are based on the syntactic parsing of the sentence, including the set of dependency relations on the shortest dependency path between the two given entities. Semantic features include entity class, entity mention and entity hypernyms in a concept hierarchy like WordNet.

A large number of studies focus on extracting more effective features to improve relation classification performance. Kambhatla combined the three types of features and trained a maximum entropy classifier to classify relations (Kambhatla, 2004). Tratz and Hovy extended Kambhatlas work by adding contextual features of entities in sentences and achieved better classification results (Tratz & Hovy, 2010). However, feature engineering is complicated and different sets of human-designed features are usually duplicate with each other, thus it is difficult to improve the relation classification performance if the features are chosen less effective (GuoDong et al., 2005).

Kernel-based methods provide an alternative way to use rich representations (e.g. syntactic parsing trees) of the input data samples without explicit feature extraction. Kernel-based approaches rely on elaborately designed kernels to learn the similarity between two data samples. Numerous researches try to improve kernel methods by exploring different similarity measures. Zelenko et al. first defined the kernel between two shallow parse trees to compute their similarity by extracting the least common subtree connecting the two entity nodes (Zelenko, Aone, & Richardella, 2003). Buneseu and Mooney designed a dependency tree kernel based on the path between two entities of interest in a dependency tree (Bunescu & Mooney, 2005). They proposed an important idea that the relation is strongly indicated by the shortest path between the entities in a dependency tree. But the kernel simply counts the number of common word classes at each node on the shortest path, leading the method suffering from low recall. Qian et al. proposed a

composite kernel for relation classification, which combines a tree kernel and a linear kernel to fully capture the syntactic structural information and entity semantic information (Qian, Zhou, Kong, Zhu, & Qian, 2008). Several kernels are compared and analysed in (M. Wang, 2008), among which convolution tree kernel with syntactic features has been proved effective with regarding to relation extraction. Kernel-based methods need a large amount of labelled data for training, however, labelled data is valuable and insufficient in most real applications.

Traditional feature-based methods and kernel-based methods depend either on complicated feature engineering or on carefully designed kernels, which require other NLP tools (e.g. dependency parsing) for pre-processing and thus leads to the problem of error-propagation. Recently, due to the powerful capacity of automatic learning features, deep neural networks have been widely used in NLP tasks and have shown promising results in relation extraction.

Socher et al. proposed a novel Recursive Matrix-Vector Model (MV-RNN) for relation classification (Socher et al., 2012) which learns a matrix-vector representation for each node in a syntactic parsing tree, where the vector captures the semantic information of a constituent and the matrix captures how it changes the meaning of neighbouring words. Each parent nodes vector representation is recursively computed by combining the children nodes representations and finally the compositional vector representation for the whole sentence is computed. Relation classification based on MV-RNN first computes the vector representation for the nearest common ancestor node of the two given entity nodes and then use the vector as features to train the classification model.

Zeng et al. explored convolutional neural network for this task and proposed a deep CNN model to combine lexical-level entity-related features and sentence-level features to train a softmax classifier for relation predication (Zeng, Liu, Lai, Zhou, & Zhao, 2014). Later dos Santos et al. also used CNN but combined with a novel pairwise ranking loss function to reduce the impact of artificial classes, which achieved the state-of-the-art result in SemEval 2010 Task 8. Meanwhile, there are other studies exploring multiple variations of recurrent neural network for relation classification, including bidirectional-LSTM (Zhang, Zheng, Hu, & Yang, 2015), hierarchical LSTM (Y. Xu et al., 2016), bidirectional tree-structured LSTM (Miwa & Bansal, 2016). Miwa and Bansal also found that LSTM-based RNN models are generally outperformed by CNN models for relation classification, due to the limited capacity of capturing linguistic structure information in neural architecture. Several works also rebuilt various neural networks on the shortest dependency path between two entities nodes and yield competitive results (Y. Liu et al., 2015; K. Xu, Feng, Huang, & Zhao, 2015; Y. Xu et al., 2015, 2016).

3.2.2.3 Joint Entity and Relation Extraction

Entity recognition and relation extraction are two highly related tasks in natural language processing, since given entity types will help to identify the semantic relation between a pair of

annotated entities and given possible relations between entities will help to predict entity types. For example, if two entities have hyponymy or synonymy relation, they must belong to the same entity type. If two entities are in different types, there is no possible that they have hyponymy or synonymy relation. Therefore, a joint model for entity recognition and relation extraction will enhance the performance of both tasks.

The goal of end-to-end entity recognition and relation extraction is to identify entity mentions from unstructured texts and predict possible semantic relations between pairs of entities in the same sentence. Most previous approaches use pipeline framework to solve this problem, which decomposes the task into two separate components: entity mention detection and relation classification. There is one big drawback with such pipelined methods that it prohibits the interactions between related components and ignores cross-task dependencies. Errors in entity recognition are propagated to relation extraction without any chance to modify, even if the context information surrounding a pair of entities strongly implies a specific relation.

Several works have attempted on joint entity recognition and relation extraction to address problems in pipelined approaches. Roth and Yih proposed a joint inference for entity and relation extraction by linear programming (Roth & Yih, 2004, 2007), which first trained a set of entity and relation classifiers based on local features to get classification probabilities, and then globally optimized over suggestions of the classifiers by integer linear programming (ILP). Classification probabilities were used to build the objective function and requirements on entity types for specific relations were formulated as constraints. It has been shown that global inference improves stand-alone learning for entity and relation extraction. One limitation is that their work failed to solve the task in an end-to-end manner, because it assumed that entity boundaries were given and the joint model only predict entity types and relation classes. Yang and Cardie applied similar ILP framework to joint inference opinion-related entities and relations for the task of opinion extraction (B. Yang & Cardie, 2013). The only difference is that they use CRF-based sequence labelling to replace local entity classifier and thus eliminate the assumption that entity boundaries were given. Although these works witnessed the advantage of joint model on entity recognition and relation extraction, ILP-based global inference relied on local models separately learned for each task without integrating related tasks in a unified learning process.

Some other researches applied probabilistic graphical models for joint extraction of entities and relations (X. Yu & Lam, 2010; Singh, Riedel, Martin, Zheng, & McCallum, 2013), which constructed a joint model by combining all variables and factors of each individual task into a single graphical model and solved related subtasks in a joint inference process. However, this work also assumed that entity boundaries were given and failed to achieve end-to-end extraction goal. Recently, Li et al. proposed to formulate the joint entity and relation extraction task as a structured predication problem (Q. Li & Ji, 2014). First, each sentence is modelled as a graph where entities are nodes and relations are directed edges, and then linear model is used to

predict the most probable graph structure based on multiple local and global features. Feature weights were estimated in the structured perceptron learning framework and the process of weighs estimation is to extract entities and relations simultaneously in a joint model. However, this method requires large number of hand-crafted local and global features, which reduce the generality of the method for entity and relation extraction in different domains.

Collobert et al. proposed a unified neural network architecture (Collobert et al., 2011) and applied it to multi-task learning in natural language processing, for example part-of-speech tagging, chunking and named entity recognition. One basic assumption is that features trained for one task could be useful for other related tasks. The unified neural architecture leverages this assumption: related tasks share common representation layers and each task has a specific function layer. However, this architecture has not been applied to entity recognition and relation extraction. Khashabi first attempted to solve end-to-end entity and relation extraction with neural network models in a pipeline approach (Khashabi, 2013), which built recursive neural network based on syntactic parsing trees and train the network for entity recognition and relation classification separately. This work regarded each node (constituency) as a candidate entity and thus help to decide entity boundaries in sentences. In this chapter, we will propose a joint neural network model for end-to-end entity and relation extraction and train the model for multi-task learning in a unified process.

3.2.3 Concept Hierarchy Generation

Concept hierarchy, such as WordNet ontology and Yahoo! Directory, is a natural way to organize human knowledge and has been manually created in the past decades. However, human design suffers from heavy workload and low efficiency. This motivates studies on concept hierarchy generation methods. To build a concept hierarchy, the most important part is to identify hyponymy relation (also called "is-a" relation) between two entities (or concepts).

Hyponymy relation is one basic type of semantic relations that has been widely used in the construction of taxonomy, ontology and knowledge base. Given two concepts c_1 and c_2, if the semantic field of c_2 contains c_1, we say that the hyponym c_1 is in an "is-a" relation with its hypernym c_2. This section reviews some typical and important work on "is-a" relation identification.

Traditional approaches to identifying "is-a" relations can be generally divided into two categories: pattern-based methods and statistic-based methods. Pattern based methods mainly rely on linguistic techniques (e.g. lexical analysis and syntactic analysis) to acquire "is-a" patterns, for example "A such as B" or "A is one kind of B", and match them within given documents to identify paired hypernym-hyponym entities. Hearst first introduced lexico-syntactic patterns and used bootstrapping to discover new patterns (Hearst, 1992). In the following studies, patterns could be manually designed (Kozareva & Hovy, 2010) or automatically generated (Snow,

Jurafsky, & Ng, 2005; Navigli, Velardi, & Faralli, 2011). Pattern-based methods are simple to implement, but suffer from low precision and coverage, because fixed surface-level pattern matching could not adapt to flexible and variable structures in natural language.

Statistic-based methods compensate for the low coverage of pattern-based methods and identify "is-a" relations by calculating semantic relatedness between entities using a large variety of features, including co-occurrence features, entity-related context features and syntactic dependency parsing features (Turney & Pantel, 2010). These methods mostly rely on the distributional inclusion hypothesis (*DIH*) (Geffet & Dagan, 2005), which assumes that hypernyms have broader contexts than hyponyms. Specifically, a concept c_1 entails a concept c_2 if in any context that concept c_1 is used so can be concept c_2, which means if c_2 is a hypernym concept of c_1, then a significant number of distributional features of c_1 are included in the features of c_2. For example, if c_1 is cat and c_2 is animal, most features of cat are included in the features of animal, but at least some features of animal do not apply to cat. For instance, the term rights is strongly associated with animal, but not so much for cat (Z. Yu, Wang, Lin, & Wang, 2015). Methods based on DIH differ in the calculation of semantic relatedness between entities for "is-a" relation identification, such as calculating the number of common features shared by hypernym entity and hyponym entity (Weeds, Weir, & McCarthy, 2004), calculating the number of unique features possessed by hyponym entity (Lenci & Benotto, 2012) and a measure of average precision derived from information retrieval (Kotlerman, Dagan, Szpektor, & Zhitomirsky-Geffet, 2010).

There are three main problems with these statistic-based methods: (1) DIH hypothesis does not hold for all pairs of entities with "is-a" relation, since hyponyms could have some unique features incompatible with their hypernyms. For example, American is a hypernym of Obama, but Obama definitely has some unique features like black man that do not apply to American; (2) The measures of semantic relatedness could not distinguish hyponymy relation from synonymy relation and part-and-whole relation; (3) The statistic-based methods suffer from low accuracy and heavily rely on feature selection.

Recently, wording embeddings (Bengio, Ducharme, Vincent, & Jauvin, 2003) have been widely used in many other NLP tasks and proved effective in capturing both linguistic and semantic relations between words. However, directly using co-occurrence based wording embeddings could not attain desirable results for hyponymy relation identification, because co-occurrence based representation learning could only make similar words have similar embeddings and thus have no means to reveal "is-a" relation.

Fu et al. discovered that word embeddings could preserve hypernym-hyponym relationship (Fu et al., 2014), for example $v(laborer) - v(carpenter) \approx v(actor) - v(clown)$ which is similar to the famous semantic example $v(king) - v(queen) \approx v(man) - v(woman)$ discussed in (Mikolov, Yih, & Zweig, 2013). According to this observation, they proposed a hypernym-

hyponym identification method based on word embeddings, which leaned a linear transition matrix Φ mapping words to their hypernyms. Specifically, given a word x and its hypernym y, such that $v(y) = \phi \times v(x)$, and thus the problem is transformed to a typical linear regression problem. Tan et al. simplified this work by replacing the linear transition matrix Φ with an "is-a" vector $v(is-a)$ and captured the hypernym-hyponym relationship by this vector (Tan, Gupta, & van Genabith, 2015), for example $v(clown) \times v(is-a) \approx v(actor)$. However, it is not effective to directly use co-occurrence based wording embeddings for the purpose of capturing hypernym-hyponym relationship. Moreover, they only learned through the pairs of words without considering the context information between them. The context information in the sentence has been shown effective to identify hyponymy relation in texts (Levy, Remus, Biemann, & Dagan, 2015; Anh, Tay, Hui, & Ng, 2016).

Different from the above two methods, Yu et al. proposed a supervised distance-margin neural network which directly learning embeddings from a set of extracted hypernym-hyponym word pairs (Z. Yu et al., 2015), instead of learning the representations from word co-occurrence. They applied such term embeddings as features to SVM classifier to predict positive hypernymy pairs. However, this method heavily relied on the pre-extracted hypernym-hyponym pairs for training. If a pair of hypernymy terms is not in the training set, it failed to predicate the hypernymy relation due to the unknown term embeddings. Besides, this method also ignored the contextual information between hypernym and hyponym words which could be an important indicator for the hypernymy relation identification.

3.3 Scientific Concept Hierarchy Generation

The micro scientific resource space consists of three dimensions: Task, Process and Material, which respectively describe the research problem, methodology and data. Constructing the concept hierarchies in each micro dimension needs to extract the three types of entities and identify hyponymy and synonymy relations between a pair of entities.

3.3.1 Methodology

The entity recognition and relation extraction are two highly related tasks in natural language processing. Given entity types will help to improve the accuracy of relation identification and given possible relations between paired entities will also prompt the inference on entity types. For example, a pair of entities with the hyponymy relation must be in the same type. Thus, a joint model for entity recognition and relation extraction will enhance both tasks.

Currently, most approaches use pipeline framework to solve these two tasks separately, which prohibits the interactions between related tasks and ignores cross-task dependencies. Errors in entity recognition are propagated to relation extraction without any chance to revise, even if the context information surrounding a pair of entities strongly implies a specific relation.

This chapter proposes a joint neural network model, called JER-Tree-LSTM, to simultaneously extract entities and relations from scientific articles in an end-to-end manner. Specifically, the joint neural network model first learns vector representations for nodes of constituent in a constituency tree and then performs soft-max classification for entity type prediction and learns a transition matrix to transform hyponym embeddings to hypernym embeddings. Finally, a supervised SVM classifier is trained to classify relations between a pair of entities based on entity embeddings and the transition matrix.

3.3.2 Joint Entity and Relation Extraction Model

The basic idea behind the joint neural network model is that embeddings trained for one task could be useful for other related tasks. In the joint model tasks of interests share the basic representation layers and each individual task possesses a separate functional layer. Figure 3.2 shows the general framework of the joint model, where the shared representation layer learns unified vector representations for the input multi-channel features. It consists of a multi-channel embedding layer and a Tree-LSTM layer. In the task of entity recognition, hidden vectors are mapped and classified into entity types through the projection layer. In the task of relation extraction, hyponym embeddings are transformed into their corresponding hypernym embeddings through the transformation layer.

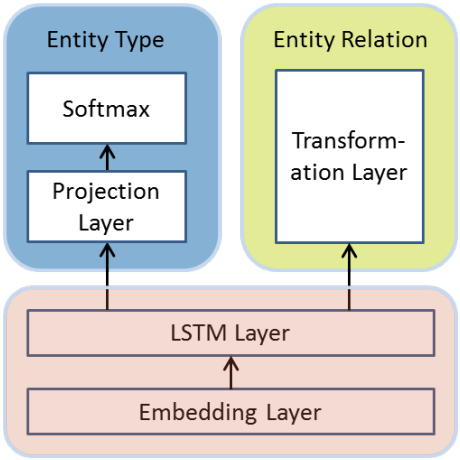


Figure 3.2: The framework of the joint entity and relation extraction model

Figure 3.3 demonstrates the tree-structured neural network architecture of the joint entity and relation extraction model based on the constituency tree. The example sentence is extracted from an article (Green, Behabtu, Pasquali, & Adams, 2009) in the field of material science in SemEval 2017 Task 10. Figure 3.3-a shows the annotation result on this sentence. It contains an "is-a" relation between a pair of Process entities, that is, *CVD technique* is a hypernym entity of *HiPco process*. It also contains an instance of nested entities that *HiPco* itself is a Material entity but also attends in a Process entity of *HiPco Process*. The constituency tree of this sentence is shown in Figure 3.3-b.

The joint entity and relation extraction model builds the tree-structured neural network

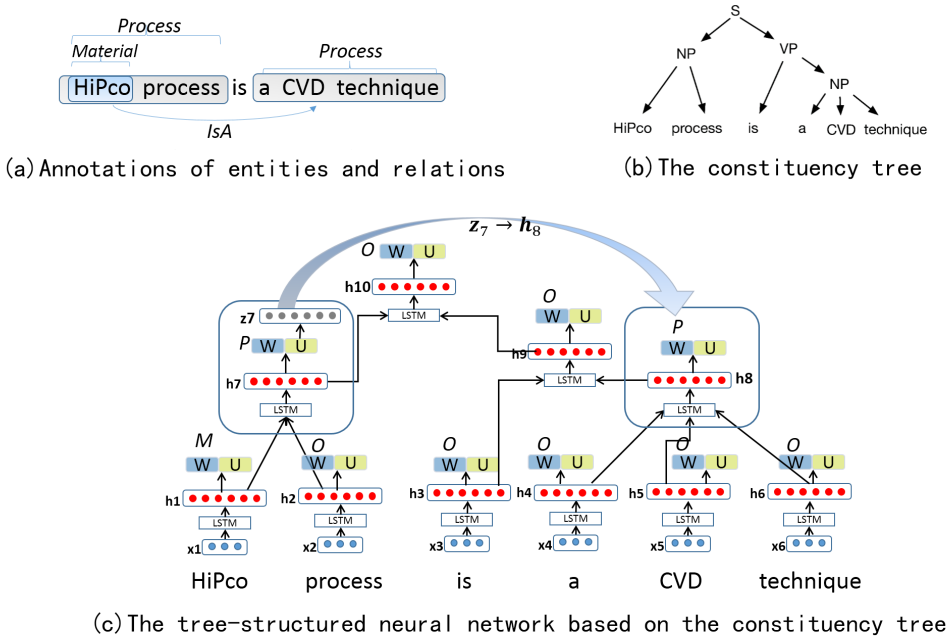


Figure 3.3: The joint neural network model based on the constituency tree.

according to the constituency tree in Figure 3.3-c, where leaf nodes correspond to each word in the sentence and $\mathbf{x}$ are word feature vectors in multi-channel embedding layer and fed into Tree-LSTM layer to generate hidden state vector $\mathbf{h}$. The joint extraction model stacks a projection layer and a transformation layer based on the Tree-LSTM output vector $\mathbf{h}$. W and U are neural network connection weights in task-specific functional layers, thus represent the projection layer and transformation layer respectively. Figure 3.4 shows the concrete neural network on single constituency tree node.

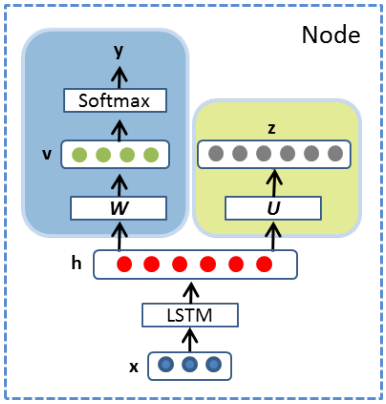


Figure 3.4: The neural network on the constituency tree node.

The objective function of the joint neural network model consists of two parts: (1) minimize the cross-entropy error for entity recognition; (2) maximize the distant margin between positive and negative instances for relation extraction. The details of each layers structure and function will be discussed in the following subsections.

3.3.2.1 Basic Representation Layers

(1) Multi-channel embedding layer This work applies five types of features to build the joint neural network model for entity and relation extraction, including character-level embedding, token-level embedding, part-of-speech, chunk and capitalization features. We call these five types of features as multi-channel features and concatenate them in the multi-channel embedding layer to represent each token.

Token-level Embedding $\boldsymbol{x^w}$: Each token represented by a one-hot vector in a fixed vocabulary is mapped into a dense vector space by looking up in a word embedding table $\boldsymbol{L^w}$. Assume that the vocabulary size is V and the word embedding size is d_w, then the word embedding table is of size $V \times d_w$. Due to the large difference in the word usage between scientific articles and news reports, we build scientific corpus and learn word embeddings for scientific literatures. This corpus is built with full journal articles in ScienceDirect database using SemEval 2017 Task 10 corpus as seeds and expanding based on citations. Word embeddings are learned by word2vec model on this corpus.

Character-level Embedding $\boldsymbol{x^b}$: The character-level embedding for a token is derived from its character sequence using bidirectional LSTM and concatenating the forward and backward outputs as the character-level embedding $\boldsymbol{x^b}$. Each character is mapped to a real-valued vector according to a character embedding table. Assume that the character vocabulary size is V^b and the character embedding size is d_b, then the character embedding table $\boldsymbol{L^b}$ is of size $V^b \times d_b$. The character embedding table is initialized randomly and updated with training. Introducing character-level embeddings into the joint entity and relation extraction model could help handle the occurrence of out-of-vocabulary words like terminologies or formulas in scientific documents. Moreover, it could make morphological variations of the same stem share similar representations.

Part-of-speech Feature $\boldsymbol{x^p}$: Word embeddings alone may not be enough to capture linguistic and semantic properties of words, or even conflict with some specific context, thus we add part-of-speech embedding into multi-channel features. Penn Treebank provides part-of-speech tag set including 48 different POS tags. Assume that the POS embedding size is d_{pos}, then the POS embedding table $\boldsymbol{L^p}$ is of size $48 \times d_{pos}$. The POS embedding table is initialized randomly and tuned during training.

Chunk Feature $\boldsymbol{x^c}$: Besides POS feature, chunk feature is also important to entity recognition, thus we use Illinois Chunker to perform shallow parsing on each sentence and obtain chunk feature for each word. The chunk embedding table contains 23 different chunk tags and each is represented by a real-valued vector of size d_{chunk} initialized at random. The dimensionality of the chunk embedding table is $23 \times d_{chunk}$.

Capitalization Feature $\boldsymbol{x^i}$: The capitalization feature of words in scientific documents is particularly important to scientific entity recognition. We encode the capitalization feature of each word using a 4-dimension one-hot vector. Each dimension corresponds to one of the

following cases: (1) all letters within a word are in capital; (2) the first letter is in capital; (3) all letters are lower; and (4) any letter in a word except the first one is in capital.

We concatenate the above five embeddings as the output of the multi-channel embedding layer to represent each word in a sentence, which is formulated as the following:

$$\boldsymbol{x} = [\boldsymbol{x^w}L^w, \boldsymbol{x^b}L^b, \boldsymbol{x^p}L^p, \boldsymbol{x^c}L^c, \boldsymbol{x^i}] \in \boldsymbol{R}^{d_w+d_b+d_{pos}+d_{chunk}+4} \tag{3.1}$$

where x is the final output vector of the multi-channel embedding layer, $\boldsymbol{x^w}$,$\boldsymbol{x^b}$,$\boldsymbol{x^p}$,$\boldsymbol{x^c}$,$\boldsymbol{x^i}$ are one-hot vectors corresponding to each feature, $\boldsymbol{L^w}$,$\boldsymbol{L^b}$,$\boldsymbol{L^p}$,$\boldsymbol{L^c}$ are embedding tables and $d_w, d_b, d_{pos}, d_{chunk}$ denote the size of each feature embedding. The output vector $\boldsymbol{x}$ will serve as the input of the next Tree-LSTM layer.

(2) Tree-LSTM layer It has been shown that contextual information is important for the entity relation identification (Levy et al., 2015; Anh et al., 2016). The joint entity and relation extraction model exploits recurrent neural network (RNN) to model entity-related contextual information in a sentence. Compared with feedforward neural networks, RNN is more suitable for modelling sequential data with unlimited length due to the recurrent connections, which could enable to compress and store history information in a low-dimension vector.

Specifically, at any time step t, the hidden state vector $\boldsymbol{h_t}$ stores the information from the beginning to the current time step, which is derived from the current input $\boldsymbol{x_t}$ and the previous hidden state vector $\boldsymbol{h_{t-1}}$. Formally, the update function is given by equation 3.2:

$$\boldsymbol{h_t} = tanh(W\boldsymbol{x_t} + U\boldsymbol{h_{t-1}} + \boldsymbol{b}) \tag{3.2}$$

where W and U are neural connection weights for the input and recurrent connections, $\boldsymbol{b}$ is a bias term for hidden state vectors and $tanh$ is a non-linear activation function.

However, RNNs are hard to train with the gradient back-propagation algorithm through time due to the well-known problem of gradient vanishing or exploding (Hochreiter, 1998), that is when the gradient of the error function is propagated back through the network on a long sequence, it may probably get to decay or grow exponentially and leads to the failure of training.

One approach called Long Short-Term Memory ($LSTM$) was proposed to overcome the gradient vanishing or exploding during back propagation through long-sequence recurrent network (Hochreiter, 1998) by introducing selective gating mechanism and memory unit to modify the network architecture. Recently, a large number of LSTM-RNN variants have been proposed and applied successfully in many NLP applications. In this subsection, we will briefly introduce LSTM and several LSTM variants (Zaremba & Sutskever, 2014).

Concretely, at any time step t, LSTM recurrent neural network unit consists of a set of d-dimension vector components, including three adaptive gates $\boldsymbol{i_t}$, $\boldsymbol{f_t}$ and $\boldsymbol{o_t}$, a memory cell $\boldsymbol{c_t}$ and a hidden state $\boldsymbol{h_t}$. The connectivity structure of a LSTM unit is depicted in Figure 3.5. The equation 3.3 lists the update for each component, where $\boldsymbol{i_t}$, $\boldsymbol{f_t}$ and $\boldsymbol{o_t}$ are the input gate, forget gate and output gate respectively. Each gate is derived from the current input $\boldsymbol{x_t}$ and previous

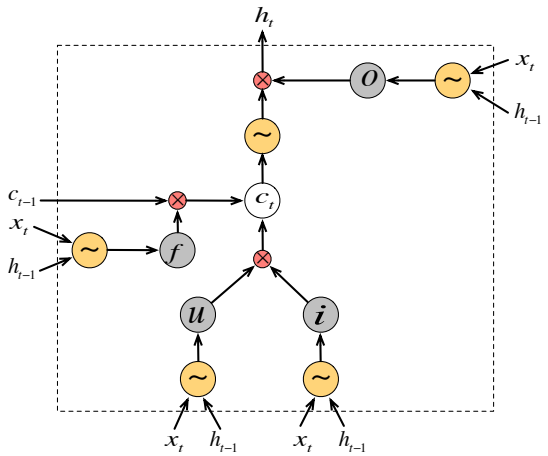


Figure 3.5: The LSTM unit

hidden state $\boldsymbol{h_{t-1}}$ using the sigmoid function to make it fall into $[0, 1]$. They respectively control the extent to which LSTM memory cells accept the current input, keep from the previous state and output to the current hidden state. $\boldsymbol{c_t}$ is the current memory cell which is a combination of the candidate content $\boldsymbol{u_t}$ and the previous cell content $\boldsymbol{c_{t-1}}$ weighted by the input gate $\boldsymbol{i_t}$ and the forget gate $\boldsymbol{f_t}$. The output of the current LSTM unit $\boldsymbol{h_t}$ is updated by first applying non-linear activation function tanh on $\boldsymbol{c_t}$ and then weighted by output gate $\boldsymbol{o_t}$.

$$
\begin{aligned}
\boldsymbol{i_t} &= \sigma(W^{(i)}\boldsymbol{x_t} + U^{(i)}\boldsymbol{h_{t-1}} + \boldsymbol{b}^{(i)}) \\
\boldsymbol{f_t} &= \sigma(W^{(f)}\boldsymbol{x_t} + U^{(f)}\boldsymbol{h_{t-1}} + \boldsymbol{b}^{(f)}) \\
\boldsymbol{o_t} &= \sigma(W^{(o)}\boldsymbol{x_t} + U^{(o)}\boldsymbol{h_{t-1}} + \boldsymbol{b}^{(o)}) \\
\boldsymbol{u_t} &= tanh(W^{(u)}\boldsymbol{x_t} + U^{(u)}\boldsymbol{h_{t-1}} + \boldsymbol{b}^{(u)}) \\
\boldsymbol{c_t} &= \boldsymbol{i_t} \times \boldsymbol{u_t} + \boldsymbol{f_t} \times \boldsymbol{c_{t-1}} \\
\boldsymbol{h_t} &= \boldsymbol{o_t} \times tanh(\boldsymbol{c_t})
\end{aligned}
\tag{3.3}
$$

In recent years, the LSTM-based recurrent neural network mainly has the following three variants:

Bidirectional LSTM (Graves, Jaitly, & Mohamed, 2013): The improvement of bidirectional LSTM is that it contains two parallel LSTMs: one runs on the input sequence and the other runs on the reverse sequence. At each time step, the hidden state of the bidirectional LSTM concatenates the forward and backward hidden states, which enables to capture both history and future information.

Multilayer LSTM (Sutskever, Vinyals, & Le, 2014): Multilayer LSTM-based recurrent neural networks are more powerful in sequence representation and learning, however the increasing in the number of parameters in the multilayer architecture not only burden the training process but also increase the model complexity. In multilayer LSTMs, the hidden state of an LSTM unit in the l layer is served as the input to the LSTM unit in the $l + 1$ layer.

Tree-Structured LSTM (Tai, Socher, & Manning, 2015): Natural language is considered to have syntactic properties, which means words in closer grammatical relationship are com-

bined into phrases according to the syntactic tree rather than the order of words. Since chain-structured LSTMs are insufficient to capture such syntactic properties, tree-Structure LSTM is a better alternative solution to compose a sentence representation from its sub-constituents in a given syntactic tree. In (Tai et al., 2015), tree LSTMs exhibit a more powerful capacity in learning semantic representations for sentences than chain-based LSTMs.

The joint entity and relation extraction model exploits tree-structured LSTM for advanced feature representation. In Figure 3.3-c, the tree-LSTM layer builds on a given constituency tree in Figure 3.3-b, where each leaf node takes the multi-channel feature embeddings $\boldsymbol{x}$ as input and then tree-LSTM units recursively derive each phrase and sentence representation $\boldsymbol{h}$ from its child-constituent vector representations. Tree LSTM can fully capture the compositionality of words in a sentence and learn semantic representations for nodes in a constituency tree.

Next we introduce the details of each component in tree LSTM unit and their update functions. As in standard LSTM units, each tree LSTM unit also consists of three adaptive gates $\boldsymbol{i_j}$,$\boldsymbol{f_j}$ and $\boldsymbol{o_j}$, a memory cell $\boldsymbol{c_j}$ and a hidden state $\boldsymbol{h_j}$. The difference between tree LSTM unit and standard LSTM unit is that the updates of the gates and memory cell rely on the hidden states of all child nodes. Besides, a standard LSTM unit has a single forget gate that controls the extent to which the previous state is forgotten, while a tree LSTM unit sets one forget gate for each individual child node. This setup allows to control the extent to which the memory cell selectively keep from each child node.

The equation 3.4 gives the update for each component vector. Give a constituency tree, let $C(j)$ denote the children set of node j. $\widetilde{\boldsymbol{h_j}}$ is the temp hidden state of node j that sums over all childrens hidden states. The input gate $\boldsymbol{i_j}$ and output gate $\boldsymbol{o_j}$ are both derived from $\widetilde{\boldsymbol{h_j}}$, while the forget gate $\boldsymbol{f_{jk}}$ for each child node k is computed using the corresponding child hidden state $\boldsymbol{h_k}$. The memory cell $\boldsymbol{c_j}$ combines the memory cell candidate $\boldsymbol{u_j}$ and arbitrarily many child memory cells $\boldsymbol{c_k}$ weighted by the input gate $\boldsymbol{i_j}$ and forget gates $\boldsymbol{f_{jk}}$.

$$
\begin{aligned}
\widetilde{\boldsymbol{h_j}} &= \sum_{k \in C(j)} \boldsymbol{h_k} \\
\boldsymbol{i_j} &= \sigma(W^{(i)}\boldsymbol{x_j} + U^{(i)}\widetilde{\boldsymbol{h_j}} + \boldsymbol{b}^{(i)}) \\
\boldsymbol{f_{jk}} &= \sigma(W^{(f)}\boldsymbol{x_j} + U^{(f)}\boldsymbol{h_{tk}} + \boldsymbol{b}^{(f)}) \\
\boldsymbol{o_j} &= \sigma(W^{(o)}\boldsymbol{x_j} + U^{(o)}\widetilde{\boldsymbol{h_j}} + \boldsymbol{b}^{(o)}) \\
\boldsymbol{u_j} &= tanh(W^{(u)}\boldsymbol{x_j} + U^{(u)}\widetilde{\boldsymbol{h_j}} + \boldsymbol{b}^{(u)}) \\
\boldsymbol{c_j} &= \boldsymbol{i_j} \times \boldsymbol{u_j} + \sum_{k \in C(j)} \boldsymbol{f_{jk}} \times \boldsymbol{c_k} \\
\boldsymbol{h_j} &= \boldsymbol{o_j} \times tanh(\boldsymbol{c_j})
\end{aligned}
\tag{3.4}
$$

Figure 3.6 depicts the connectivity structure of a tree LSTM unit. The rectangle represents a tree LSTM unit for one node. Node 1 is the parent and nodes 2 and 3 are children of node 1.

The joint entity and relation extraction model decides the entity boundaries with the help

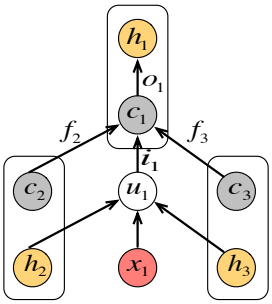


Figure 3.6: The tree-structured LSTM unit

of the constituency tree. For the task of entity recognition, there is a basic assumption that an entity should be a subsequence of words in a sentence. Moreover, it requires words in the subsequence must share a common parent in the corresponding constituency tree and this parent node should not span any other words except the words in the subsequence. Nodes in a constituency tree would be possible entities and we therefore turn to tree-structured LSTM recurrent neural network to learn vector representations for each node in a constituency tree.

In addition, there is another consideration of tree LSTM in the joint model that the occurrence of nested entities is quite common in scientific documents, where a single word could attend in multiple entities with different types. Figure 3.7 shows an example of nested entities from a paper (Paper ID: S0032386109005485) in SemEval 2017 Task 10 corpus. This example contains two different nested entities: one is a Material entity contained in a Task entity and the other is two Material entities contained in a Process entity. According to the statistics on SemEval 2017 Task 10 corpus, the nested entities account for 8.78% in the whole dataset. Specifically, they occupy 9.23% in the training set and 7.51% in the test set. Traditional sequence labelling approach for entity recognition based on *BIO* or *BILOU* tagging schema could only assign one entity type for each single word, which fails to solve nested entity extraction in scientific documents. Tree-structured LSTM model provides an attractive option for joint entity and relation extraction due to its capacity in handling nested entities. It allows to predict different entity types for nodes containing a same word.

Figure 3.7: An example of nested entities.

3.3.2.2 Task-specific Functional Layers

In the joint model, each related task has a separate functional layer. For entity recognition, the projection layer performs mapping and soft-max classification for entity type prediction. For

relation identification, the transformation layer learns a transition matrix to transform hyponym entity embeddings to their hypernym embeddings. In Figure 3.3 and Figure 3.4, W and U are connection weight matrices in the projection layer and transformation layer respectively.

(1) The projection layer for entity extraction The joint model regards the entity extraction problem as an entity type classification problem which classifying nodes in a given constituency tree into 4 categories: Task, Process, Material and None. Task represents a class of research task or problem related entities; Process represents a class of method or process related entities; Material represents a class of data or material related entities and None represents the node is not in any of the above categories.

Like other entity type classification systems, the feature vectors $\boldsymbol{h}$ (hidden states in the tree LSTM layer) of each node in a constituency tree are fed to a softmax classifier in the projection layer whose output is the prior probability distribution $\widehat{\boldsymbol{y}}$ over entity types as shown in Figure 3.4. The projection layer is formulated by Equation 3.5:

$$\widehat{\boldsymbol{y}} = softmax(\boldsymbol{W}\boldsymbol{h} + \boldsymbol{b}^{\boldsymbol{W}}) \tag{3.5}$$

where $\boldsymbol{h}$ is a d-dimension vector and $\boldsymbol{W}$ is the projection matrix of size $4 \times d$. The final output of the projection layer is $\widehat{\boldsymbol{y}}$ whose dimensionality equals to the number of entity types. Each entry can be interpreted as the score of the corresponding entity type. The objective function for entity recognition is to minimize the cross-entropy between the ground-truth vector and the projection layer output $\widehat{\boldsymbol{y}}$.

(2) The transformation layer for relation extraction The joint model relies on the transformation layer for hypernym-hyponym relation extraction, which learns a transition matrix $\boldsymbol{U}$ to transform hyponym entities to their corresponding hypernym entities. Specifically, as shown in Figure 3.4, hyponym embeddings $\boldsymbol{h}$ (hidden states in the tree LSTM layer) are transformed to their hypernym embeddings $\boldsymbol{z}$ through a fully connected transformation layer formulated by equation 3.6:

$$\boldsymbol{z} = tanh(\boldsymbol{U}\boldsymbol{h} + \boldsymbol{b}^{\boldsymbol{U}}) \tag{3.6}$$

where $\boldsymbol{h}$, $\boldsymbol{z}$ are d-dimension vectors and $\boldsymbol{U}$ is the $d \times d$ transition matrix mapping hyponym vectors to hypernym vectors.

The transformation layer associates hyponym embeddings and hypernym embeddings and learns the transition matrix by maximizing the distance margin between positive and negative relation instances using a pair-wise training strategy.

3.3.2.3 Learning Objective

The joint model builds a unified neural network to extract entities and relations simultaneously from scientific documents, thus the learning objective for the joint model incorporates both parts.

We first introduce notations for entities and relations in a scientific document. The joint model takes sentences as input. For a given sentence s of length l, s contains k entities denoted by $Entity(s) = \{e_1, e_2, ..., e_k\}$ and each entity e_j consists of several continuous words in s. Besides the entities, s contains m_h pairs of hypernymy relations denoted by $Hyper(s) = \{hyp_1, hyp_2, ..., hyp_{m_h}\}$. Each hypernymy relation consists of a pair of entities, denoted as $hyp_i = (e_{ha} \rightarrow e_{hb})$. The hypernymy relations are directional, where the first entity e_{ha} is a hypernym entity and the second entity e_{hb} is a hyponym entity. In addition, the sentence s may also contain m_s pairs of synonymy relations denoted by $Synon(s) = \{syn_1, syn_2, ..., syn_{m_s}\}$. Each synonymy relation consists of an unordered entity pair, denoted as $syn_i = (e_{sa}, e_{sb})$. Assuming that there are N_s nodes in the constituency tree with regarding to sentence s, each entity in the sentence s could be mapped to one of N_s nodes in the tree. As for entity recognition, each node in the tree would be assigned to one entity class: Task, Process, Material or None.

The final objective function J in the joint model is an average of sentence loss J_s on all sentences in the training set with $L2$ regularization on connection weights $\boldsymbol{W}$ and $\boldsymbol{U}$ parameterized by λ, given by equation 3.7:

$$J = \frac{1}{|S|} \sum_s J_s + \frac{\lambda}{2} [\sum_{i,j} W_{i,j}^2 + \sum_{i',j'} U_{i',j'}^2] \tag{3.7}$$

For a specific sentence s, the loss function J_s is composed of two parts given by equation 3.8. One is a cross entropy loss denoted as $J_(s, e)$ for the entity recognition task and the other is a pairwise distance-margin loss for the relation extraction task. $J_{s,hyp}$ and $J_{s,syn}$ represent the loss for hypernymy relations and synonymy relations in sentence s respectively. In equation 3.8, α is a balance factor between the two tasks.

$$J_s = \alpha J_{s,e} + (1 - \alpha) \times (J_{s,hyp} + J_{s,syn}) \tag{3.8}$$

Specifically, the loss $J_{s,e}$ for entity recognition is to calculate the average cross entropy between the ground-truth label vector $\boldsymbol{y_n}$ and the projection layer output $\widehat{\boldsymbol{y_n}}$ for all nodes in the constituency tree. The calculation is given by equation 3.9:

$$\begin{aligned} J_{s,e} &= -\frac{1}{N_S} \sum_n logP(y_n | subtree(n)) \\ &= -\frac{1}{N_S} \sum_n Cross\_Entropy(\boldsymbol{y_n}, \widehat{\boldsymbol{y_n}}) \\ &= -\frac{1}{N_S} \sum_n \sum_c y_{n,c} \times log(\widehat{y}_{n,c}) \end{aligned} \tag{3.9}$$

where $subtree(n)$ represents a subtree rooted at node n. $y_{n,c}$ and $\widehat{y}_{n,c}$ denote the c^{th} component of y_n and $\widehat{\boldsymbol{y_n}}$ corresponding to an entity class c.

As for relation extraction, the joint model aims to identify two types of relations: hypernymy relations and synonymy relations. For any synonymy relation $syn = (e_{sa}, e_{sb})$, the joint model tries to learn embeddings such that $\boldsymbol{h}(e_{sa})$ is close to $\boldsymbol{h}(e_{sb})$, where $\boldsymbol{h}(e_{sa})$ and $\boldsymbol{h}(e_{sb})$ are hidden state vectors of a pair of synonymy entities e_{sa} and e_{sb}.

Similarly, for any hypernymy relation $hyp = (e_{ha} \rightarrow e_{hb})$, the learning objective of the joint model is to make $\boldsymbol{h}(e_{ha})$ close to $\boldsymbol{z}(e_{hb})$, where $\boldsymbol{h}(e_{ha})$ is the hidden state vector of the hypernym entity e_{ha} and $\boldsymbol{z}(e_{hb})$ is the output vector of the transformation layer for the hyponym entity e_{hb}. The transformation layer learns to transform hyponym embeddings to their corresponding hypernym embeddings. The formulation of $J_{s,hyp}$ and $J_{s,syn}$ will be detailed next.

We choose 1-norm distance as the distance measure between a pair of entities denoted as δ, thus the distance between paired synonymy entities and between paired hypernymy entities is defined as equation 3.10:

$$\begin{aligned} \delta_s(syn_i) &= \|\boldsymbol{h}(e_{sa}) - \boldsymbol{h}(e_{sb})\|_1 \\ \delta_h(hyp_i) &= \|\boldsymbol{h}(e_{ha}) - \boldsymbol{z}(e_{hb})\|_1 \end{aligned} \tag{3.10}$$

$\delta_s(syn)$ is called the synonymy entity distance in relation syn and $\delta_h(hyp)$ is called the hypernymy entity distance in relation hyp.

Based on the definition of entity distance in relations, the pairwise distance-margin loss function in the joint model for hypernymy and synonymy relation identification represented by $J_{s,hyp}$ and $J_{s,syn}$ in sentence s is given by equation 3.11:

$$\begin{aligned} J_{s,hyp} &= \sum_{hyp_i} max(0, \delta_h(hyp_i) - \delta_h(hyp_i^-) + \epsilon) \\ J_{s,syn} &= \sum_{syn_i} max(0, \delta_s(syn_i) - \delta_s(syn_i^-) + \epsilon) \end{aligned} \tag{3.11}$$

where ϵ is the margin. $J_{s,hyp}$ and $J_{s,syn}$ respectively sum over the hypernymy relation set and the synonymy relation set in sentence s, trying to maximize the margin of entity distance between positive and negative relation instances. hyp_i and syn_i represent a positive hypernymy relation instance and a positive synonymy relation instance respectively, while hyp_i^- and syn_i^- represent negative relation instances. $\delta_h(hyp_i)$ denotes the entity distance in a positive hypernymy relation instance, while $\delta_h(hyp_i^-)$ denotes the entity distance in a negative hypernymy instance. $\delta_s(syn_i)$ denotes the entity distance in a positive synonymy relation instance, while $\delta_s(syn_i^-)$ denotes the entity distance in a negative synonymy instance. The goal of training is to guarantee $\delta_h(hyp_i)$ is smaller than $\delta_h(hyp_i^-)$ and $\delta_s(syn_i)$ is smaller than $\delta_s(syn_i^-)$ by a certain margin ϵ.

The joint model employs pairwise training and we generate a negative relation instance for each corresponding positive instance. Taking the hypernymy relation as example, for a positive hypernymy instance $hyp_i = (e_{ha} \rightarrow e_{hb})$ in sentence s, we replace either e_{ha} or e_{hb} to generate a corresponding negative instance hyp_i^- which is in the form of $e_{ha}^- \rightarrow e_{hb}$ or $e_{ha} \rightarrow e_{hb}^-$. We search for nodes in the constituency tree and choose one (except for e_{ha}) that is most likely to be the hypernym of e_{hb} in the tree as e_{ha}^- and one (except for e_{hb}) that is most likely to be the hyponym of e_{ha} as e_{hb}^-. The selection of e_{ha}^- and e_{hb}^- is given by equation 3.12:

$$\begin{aligned} e_{ha}^- &= argmin_{e'_{ha} \in Tree(s), e'_{ha} \neq e_{ha}} \delta_h(hyp(e'_{ha} \rightarrow e_{hb})) \\ e_{hb}^- &= argmin_{e'_{hb} \in Tree(s), e'_{hb} \neq e_{hb}} \delta_h(hyp(e_{ha} \rightarrow e'_{hb})) \end{aligned} \tag{3.12}$$

The negative instance hyp_i^- is either $hyp(e_{ha}^- \rightarrow e_{hb})$ or $hyp(e_{ha} \rightarrow e_{hb}^-)$ with a smaller hypernymy entity distance, given by equation 3.13:

$$hyp_i^- = argmin_{hyp}\{\delta_h(hyp(e_{ha}^- \rightarrow e_{hb})), \delta_h(hyp(e_{ha} \rightarrow e_{hb}^-))\} \tag{3.13}$$

Similarly, the negative synonymy relations are produced in the same way as the hypernymy counterparts.

3.3.3 Supervised Relation Classification

To identify the relationship between pairs of entities, we employ a supervised relation classifier which takes the entity embedding vectors in the joint neural network model as input for relation classification. Specifically, we use two Support Vector Machines (*SVMs*) (Cortes & Vapnik, 1995) to respectively predict hypernymy and synonymy relations.

The joint neural network learns embeddings to encode both hypernymy and synonymy relationship, which ensures the entity distance of any positive relations is smaller than their negative counterparts. The embeddings has the following two properties:

$$\begin{aligned} \delta_h(hyp_i) &< \delta_h(hyp_i^-) \\ \delta_s(syn_i) &< \delta_s(syn_i^-) \end{aligned} \tag{3.14}$$

Therefore, given an entity pair (e_a, e_b) in a sentence, we create the input feature for the SVM classifiers by concatenating the hidden state vectors $\boldsymbol{h}(e_a)$ and $\boldsymbol{h}(e_b)$ to capture both entity features. In addition, we also add the offset vector to the input feature according to the definition of synonymy entity distance and hypernymy entity distance. As for the hypernymy classification, we concatenate the offset vector $\boldsymbol{h}(e_a) - \boldsymbol{z}(e_b)$ to capture hypernymy relation features, while for the synonymy classification, we concatenate $\boldsymbol{h}(e_a) - \boldsymbol{h}(e_b)$ to capture synonymy relation features.

Assuming that h is a d-dimension vector, for an entity pair (e_a, e_b), the input feature vector for the hypernymy SVM classifier is a $3d$-dimension vector $\langle \boldsymbol{h}(e_a), \boldsymbol{h}(e_b), \boldsymbol{h}(e_a) - \boldsymbol{z}(e_b) \rangle$ and the input feature vector for the synonymy SVM classifier is a vector $\langle \boldsymbol{h}(e_a), \boldsymbol{h}(e_b), \boldsymbol{h}(e_a) - \boldsymbol{h}(e_b) \rangle$ of the same size $3 \times d$.

3.3.4 Concept Hierarchy Generation

The goal of entity and relation extraction from scientific documents is to construct the hierarchical coordinate systems in micro dimensions in a scientific resource space. The resource space model requires coordinates in each dimension should be organized in a tree-structure hierarchy, however, in real applications scientific entities fail to comply with tree-structured organization, which means a single entity may associate with several parent entities.

For example in Figure 3.1, the entity "ILP-based Supervised Method" is derived from two parents "ILP-based Method" and "Supervised Method", which breaks the single-parent rule in

tree organization. Such case is ordinary in scientific documents. Thus, we relax the requirement of building tree-structured coordinate systems in resource space model to keep the rationality and completeness of hypernymy relations between scientific concepts.

A directed acyclic graph (DAG) is more suitable to organize scientific concepts. However, the concept graph constructed using pairwise hypernym-hyponym relations may contain cycles and redundant edges, thus it usually needs some post-processing operations to guarantee a simple DAG structure. First, we need to detect cycles and delete the weakest edge that representing the weakest hypernymy relation with the largest entity distance. Second, we delete edges that could be derived using the transitivity of hypernymy relations to further simplify the DAG. Finally, a DAG-structured concept hierarchy is generated as the micro-dimensions in a scientific resource space.

3.4 Experiment and Results

3.4.1 Datasets

The experiments use the SemEval 2017 Task 10 scientific corpus[1] which consists of 500 journal articles from ScienceDirect open access publications evenly distributed in the computer science, material science and physics domains. The corpus has been annotated with mention-level key phrases and semantic relations between them. The full text of the articles and their additional metadata are also provided in xml format.

Table 3.1: Statistics of Semeval 2017 Task-10 dataset.

Scientific E/R Dataset Analysis		Train Set	Dev Set	Test Set	Data Set
Number of Articles		350	50	100	500
Entity Summary	Total Number of Entities	6684	1162	2052	9898
	Nested Entity Number	617	98	154	869
	Nested Entity Percentage	9.23%	8.43%	7.51%	8.78%
Relation Summary	Total Number of Relations	669	168	207	1044
	Cross-sent Rel Number	26	7	8	41
	Cross-sent Rel Percentage	3.9%	4.2%	3.8%	3.92%

The whole dataset is divided into 3 parts: 350 documents are kept for training, 50 for development and 100 for testing. Table 3.1 shows the statistics of SemEval 2017 Task 10 dataset. The total number of mention-level entities is 9898, of which the nested entities account for 8.78% in the dataset. Specifically, they occupy 9.23% in the training set and 7.51% in the test set. The total number of relations is 1044 and the cross-sentence relations take up 3.92%.

Due to the large difference in the word usage between scientific articles and news reports, we build a scientific corpus to learn word embeddings for scientific literature. This corpus contains

[1]https://scienceie.github.io/

6021 journal articles, which are collected from ScienceDirect open access database by expanding the SemEval 2017 Task 10 corpus (500 articles) based on citation relations. The full articles contain 27.8M words and the vocabulary size is 140K. On this corpus, we learn 200-dimension word embeddings using word2vec model, serving as the input of the multi-channel embedding layer.

3.4.2 Experiment Setup

The experiments include two parts: the first part is to check the effectiveness of our joint neural network model JER-Tree-LSTM in entity recognition and the second part is to show its effectiveness in relation extraction.

Table 3.2: Experiment plan

Tasks	Comparative Methods	
Entity Recognition (ER)	Baseline Method	CRF
	State-of-the-art NN-based Methods	LSTM-RNN
		RNN-CRF
		RNN-GCN
		Tree LSTM
	SemEval 2017 Task 10	Top 3 Systems
	Our System	JER-Tree-LSTM
Relation Extraction (RE)	SemEval 2017 Task 10	Top 3 Systems
	Hypernym RE	(Fu, 2014)
		(Yu, 2015)
	Our System	JER-Tree-LSTM

As for entity recognition, the joint model JER-Tree-LSTM is compared with three types of systems listed in Table 3.2: (1) the traditional sequence labelling baseline CRF (Lafferty et al., 2001); (2) the state-of the-art neural network based entity recognition methods, including chain LSTM based methods (LSTM-RNN and RNN-CRF) (Huang et al., 2015), Tree LSTM methods (Tai et al., 2015) and graph convolutional neural network RNN-GCN on the constituency tree (Cetoli, Bragaglia, O'Harney, & Sloan, 2017); (3) the top 3 systems for scientific key phrases identification in SemEval 2017 Task 10 (Augenstein et al., 2017).

As for relation extraction, our system JER-Tree-LSTM is compared with two types of systems listed in Table 3.2: (1) the state-of-the-art Hypernymy relation extraction (*Hypernymy RE*) systems based on representation learning approaches, including (Fu et al., 2014) and (Z. Yu et al., 2015); (2) the top 3 systems for relation extraction in SemEval 2017 Task 10 (Augenstein et al., 2017). Micro-F1 is used as the evaluation measure to evaluate the performance of systems on scientific entity and relation extraction tasks.

3.4.3 Parameter Configuration

In the experiments, we use the development set for parameter tuning and fix the best parameter set for model testing. In CRF, LSTM-RNN, RNN-CRF and RNN-GCN, the *BIO* schema is employed, where B- prefix represents the first token of an entity, I- indicates other tokens inside an entity and O marks none entity. In Tree-LSTM and JER-Tree-LSTM, each node in a constituency tree is directly classified into 4 entity categories: Task, Process, Material and None.

Table 3.3: Configuration of parameters.

	Parameters	Values
Embeddings	d_w	200
	d_b	25
	d_{pos}	10
	d_{chunk}	10
	d_i	4
Hidden Dims	d_{w-lstm}	100
	d_{c-lstm}	25
Hyper-parameters	balance factor α	0.8 decrease to 0.2
	learning rate η	0.005
	dropout p	0.7

The parameters generally fall into three classes and are configured as in Table 3.3:

1. Dimensions of embeddings: the word embedding dimension d_w is 200; the character embedding dimension d_b is 25; the dimensions of POS embeddings d_{pos} and chunk embeddings d_{chunk} are 10; the dimension of capitalization feature d_i is 4.

2. Dimensions of hidden layers: the dimensions of hidden state vectors in character-level LSTMs d_{c-lstm} and word-level LSTMs d_{w-lstm} are set to 25 and 100 respectively. The dimension of Tree-LSTM hidden states is also set to 100.

3. Hyper-parameters: the neural network is optimized by SGD with a learning rate η of 0.005; the dropout p is set to 0.7 in our experiments to prevent neural network models from overfitting; the balance factor α dynamically adjusts the weights placing between entity recognition and relation extraction when computing the objective function in the joint model. α starts with a value of 0.8, then after 50 epochs it decreases 0.1 every 10 epochs until it reaches to 0.2. This indicates that the joint model first learns to identify entities, then turns to learn relations between entities.

3.4.4 Experimental Results

We evaluate our joint neural network model JER-Tree-LSTM with regarding to both entity recognition and relation extraction.

3.4.4.1 Results on Entity Recognition

Table 3.4 reports the results of our joint neural network model JER-Tree-LSTM for scientific entity recognition and compares it with other systems on development set and test set. According to the results, we observe that:

1. The joint model JER-Tree-LSTM performs better than the traditional sequence labelling baseline model CRF with a nearly 11% increase of the Micro-F1 measure on the test set.

2. The tree models JER-Tree-LSTM (40.64%) and Tree-LSTM (39.39%) achieve the improvements over the linear-chain RNN models, such as LSTM-RNN (34.94%), RNN-GCN (33.59%) and RNN-CRF (38.49%), which demonstrates the advantage of the constituency tree for scientific entity recognition due to its capacity in identifying nested entities.

3. The joint model JER-Tree-LSTM outperform the standard Tree-LSTM model by 3.17% in terms of the Micro-F1 score on test set and this shows the advantage of the joint model framework compared with the pipeline one.

4. Compared with the top 3 systems in SemEval 2017 Task 10, JER-Tree-LSTM achieves promising results, whose Micro-F1 score outperforms the 3^{rd} system and the 2^{nd} system with relative improvements of 6.64% and 1.64% respectively. As for the best system, the joint model JER-Tree-LSTM is slightly inferior to the 1^{st} system, since it collected a large number of scientific terms as entity gazetteer features from the web and freebase, and generated the final model as an ensemble of 15 entity models, while our joint model only employs simple lexical and constituency parsing features.

By analysing the results in Table 3.4, we can also get some additional insights to understand different performances of RNN variants. According to Table 3.4, RNN-CRF outperforms LSTM-RNN (10.16% relative), which shows the importance of the CRF layer in chain LSTMs for entity recognition. Besides, the Micro-F1 score decreases from 34.94% in LSTM-RNN to 33.59% in RNN-GCN when adding dependency parsing features to chain LSTM. The large number of equations and non-standard terms in scientific documents decrease the accuracy of the dependency parsing, which produce incorrect dependency features leading to inferior entity recognition results.

Table 3.5 shows the influence of different feature configurations of our joint model JER-Tree-LSTM on scientific entity recognition task. We investigate 4 types of features by sepa-

Table 3.4: Micro-F1 results of scientific entity recognition

Comparative Methods		Micro-F1 (Dev)	Micro-F1 (Test)
Baseline Method	CRF	31.21	29.00
State-of-the-art NN-based Methods	RNN	41.56	34.94
	RNN-GCN	38.12	33.59
	RNN-CRF	44.96	38.49
	Tree LSTM	46.92	39.39
SemEval 2017 Task-10 Participants	Best System	N/A	44
	2^{nd} System	N/A	39
	3^{rd} System	N/A	34
Our System	JER-Tree-LSTM	47.39	40.64

rately removing or changing one type of feature embeddings, including pretrained word embeddings (pretrain_emb), character-level embeddings (char-lstm), chunk embeddings (chunk) and part-of-speech embeddings (pos). Table 3.5 reports the precision (P), recall (R) and Micro-F1 scores on the test set by removing one type of feature embeddings each time. We observe that all above feature embeddings could improve the performance of JER-Tree-LSTM on scientific entity recognition to different extent, among which the pretrained word embeddings and character-level embeddings are most effective features that improve the Micro-F1 measure by 13.3% and 7.4% respectively. Additionally, part-of-speech embeddings and chunk embeddings also make improvements to the joint model JER-Tree-LSTM by 4.4% and 3.8% respectively.

Table 3.5: Impacts of feature embeddings on entity recognition.

Model	P	R	F1
JER-Tree-LSTM	38.62	42.88	40.64
No pretrain_emb	32.49	40.01	35.86
No char-lstm	40.54	35.46	37.83
No chunk	41.92	36.71	39.14
No pos	40.19	37.71	38.91

We also provide some typical case analysis on the results of JER-Tree-LSTM for scientific entity recognition in Figure 3.8 to best understand the advantages and limitations of our model. Figure 3.8 shows typical correct and error entities identified by our joint model. Green and red respectively mark correct and false entities predicted by the system. Blue labels golden entities that our system fails to pick out.

The joint model can correctly identify nested entities and non-standard word entities shown in Figure 3.8. For example, it could correctly pick out a nested Material entity “CPA pill” from a Task entity “Measuring and analysing the hold time of the CPA pill”, since the constituency tree provides a feasible labelling method for nested candidates. Besides, the joint model could identify non-standard word entities that containing numbers or special tokens, such

Correct Cases	
Overlapped Entities	[Measuring and analysing the hold time of the [CPA pill]$_{\text{Material}}$]$_{\text{Task}}$ allows the [thermal boundary resistance]$_{\text{Process}}$ within the pill to be assessed.
Non-standard word entities	This may be caused by [“rolling effect”]$_{\text{Process}}$ made by [Al_2O_3 nanoparticles]$_{\text{Material}}$ on the surface of oxide coating.
Error Cases	
Confusion of Process/Material	[SPS]$^{\text{Process}}_{\text{Material}}$ has been utilized in several studies to [retain the nanostructure of aluminum alloy powders during consolidation]$_{\text{Process}}$.
	Other models use [SWEs]$^{\text{Material}}_{\text{Process}}$ but focus on the use of multi resolution grids or irregular mesh.
Wrong Adjective boundaries	The system is divided into cubic regions, each [particle centre]$_{\text{Material}}$ is within one zone, and [potential contacting particles]$_{\text{Material}}$ are within the same or next-door zones.
	It has been shown that the most efficient forms of energy transfer between the two occurs when there is a [neighbouring carotenoid species]$_{\text{Material}}$.
Wrong of-NP boundaries	The sentence stress feedback system was devised to [predict and detect the sentence stress]$_{\text{Task}}$ of any practice sentence]$_{\text{Task}}$.
	This paper [develops and examines a simplified approach]$_{\text{Task}}$ for the on-site use of [transient response analysis]$_{\text{Process}}$ and [discusses the potential advantages]$_{\text{Process}}$ of the technique as a tool for the [assessment of the corrosion condition]$_{\text{Task}}$ of steel]$_{\text{Task}}$ in reinforced concrete structures.

Figure 3.8: Cases analysis of entity extraction results

as $Al_2O_3 nanoparticles$. Introducing character-level embeddings into the joint model could enhance the word representation leaning using word morphology and shape information and help model out-of-vocabulary words in scientific documents.

The error cases identified by our joint model JER-Tree-LSTM could be generally divided into three classes shown in Figure 3.8:

1. The model could be easily confused about Process entities and Material entities due to similar context between these two types of entities. For example, the Material entity “SPS” is predicted as a Process and the Process entity “SWEs” is predicted as a Material.

2. A second type of common errors happens to noun phrase entities starting with adjective words, the model prefers to miss the adjective words and predict wrong boundaries. For example, the model ignores *potential* and *neighbouring* when identifying Material entities.

3. Another common error type involves falsely predicting *of*-NP entities. The joint model system tends to excessively contain *of*-NP contents especially when detecting Task entities. For example, the system made mistakes when deciding the end for entities overlapped with *of*-NPs in Figure 3.8.

3.4.4.2 Results on Relation Extraction

We also investigate the effectiveness of our joint model on relation extraction task. Table 3.6 shows the Micro-F1 measure results of relation extraction on the test set of SemEval 2017 Task 10 among our system JER-Tree-LSTM and several other state-of-the-art systems.

First, we compare our system JER-Tree-LSTM with two Hypernymy relation extraction (*Hypernymy RE*) systems based on representation learning approaches. According to the results

Table 3.6: Micro-F1 results of relation extraction.

Comparative Methods		Micro-F1 (Test)
SemEval 2017 Task-10 Participants	Best System	0.28
	2^{nd} System	0.21
	3^{rd} System	0.2
Hypernym RE	(Fu, 2014)	0.08
	(Yu, 2015)	0.12
Our System	JER-Tree-LSTM	0.20

in Table 3.6, JER-Tree-LSTM outperforms the two Hypernymy RE systems with 12% improvement over (Fu, 2014) (Fu et al., 2014) and 8% improvement over (Yu, 2015) (Z. Yu et al., 2015). Fu et al. learn a linear transition matrix and Yu et al. directly learn hypernym-hyponym embeddings. They both learn from pairs of words and ignore the context information. The joint model applies Tree-LSTM to encode whole sentence information, thus improve the accuracy of relation classification.

We also compare JER-Tree-LSTM with the top 3 relation extraction systems in SemEval 2017 Task 10. As shown in Table 3.6, JER-Tree-LSTM gets almost the same Micro-F1 score with the 3^{rd} system (0.20) and the 2^{nd} system (0.21), which proves the joint model can achieve competitive performances with state-of-the-art systems on scientific relation extraction. However, the best system performs better than other systems, because it uses external resources (Wikipedia and Freebase) to enhance scientific hypernym-hyponym features in relation models and generated the final model as an ensemble of 8 relation models.

4 Prototype System of Scientific Resource Space

4.1 System Overview

The prototype system of scientific resource space uses the resource space model to organize massive scientific literature resources and provides daily browsing, retrieval and summarization services for users. The scientific resource space organizes scientific resources according to two types of dimensions: the macro-dimensions based on the metadata of scientific articles (e.g. Year, Author and Category etc.) and the micro-dimensions based on the content of scientific articles (e.g. Problem, Methodology and Data etc.).

First, the prototype system enables easy browsing on scientific articles by extracting different types of key phrases (scientific entities). The system first identifies key phrases that describing research task, process and material respectively and then highlights them in different colours, so as to help users capture key information in texts especially when people have adapted to fast skimming reading.

Second, according to the division of dimensions, the prototype system supports two types of scientific information retrieval. One is metadata retrieval based on the macro dimensions and the other is content retrieval based on the micro dimensions. Users search for scientific papers by either providing filters on macro dimensions or selecting specific content on micro dimensions.

Apart from these, the prototype system also provides summarization service to compare differences between document groups, which helps to solve some practical problems in scientific information retrieval. For example, it can facilitate the comparison on different methods or on different research problems.

The prototype system enables users to explore a scientific resource space to get acquaintance with the development of a specific domain. For example, by exploring the task dimension users could learn the categorization of research problems or tasks. Also, users could know some general technique routes or methods for addressing a specific problem by exploring the process dimension.

In this chapter, we implement the prototype system by applying research results of this thesis to modifying the category hierarchy in the macro dimension and building concept hierarchies in the micro dimensions, so that we can automatically generate a complete scientific resource space to help researchers organize and utilize scientific literature efficiently.

4.2 Function Design

The main functions of the prototype system could be generally divided into the following two types: space exploring functions and resources retrieval functions.

Space exploring functions mainly provide a series of operations on the dimensions and coordinates in a scientific resource space, including dimension exhibition, hierarchy modification and coordinate adjustments, detailed as follows:

1. *Dimension exhibition operations* support multi-dimensional scaling hierarchical visualization of a scientific resource space, which provides a unified view of a resource space. Users can first select a dimension and then zoom in and out on categories of interest. It also provides a tree view operation for hierarchical coordinates which allows users to navigate the tree along a specific path.

2. *Hierarchy modification operations* provide a series of operations to modify a category hierarchy, such as cross-branch move, pull-up, merge and split, which corresponding to the elementary operations of category hierarchy maintenance in the global phase and local phase introduced in Chapter 2 of this thesis.

3. *Coordinate adjustment operations* involve the addition and deletion of a given coordinate.

Resource browsing and retrieving functions mainly provide a series of operations with regarding to resource services, listed as follows:

1. *Resource retrieving functions* include two types of retrieval. One is metadata retrieval and the other is content retrieval. The metadata retrieval mainly uses the four macro dimensions (Year, Publication, Type and Category) to select scientific articles. As for content retrieval, users search for content of interest in two ways. One is to retrieve documents containing query keywords in title or body by keyword matching techniques. The other is that users select coordinates in the three micro dimensions (Task, Process and Material) and use coordinates as filters to jointly select documents.

2. *Resource browsing functions* support two views of browsing: single resource view and global resource view. The single resource view provides users an easy way to read a single scientific document. It is characterized by the ability to distinguish different types of keywords in different colours. Red indicates task keywords, blue labels process keywords, and green marks material keywords. It facilitates the process of reading and information seeking in texts. The global resource view provides users with a way to explore the resource distribution in a scientific resource space.

3. *Summarization functions* provide the comparative summarization service for documents in different categories, which help users to capture key information quickly and better un-

derstand the differences between categories. For example, it can facilitate the comparison between different methods applied to a same problem or between different problems solved by a same method.

4.3 User Interface Design

The user interface of the prototype system consists of three main pages: the home page of a scientific resource space shown in Figure 4.1, the main page of space exploring shown in Figure 4.2 and the page of resource browsing shown in Figure 4.4.

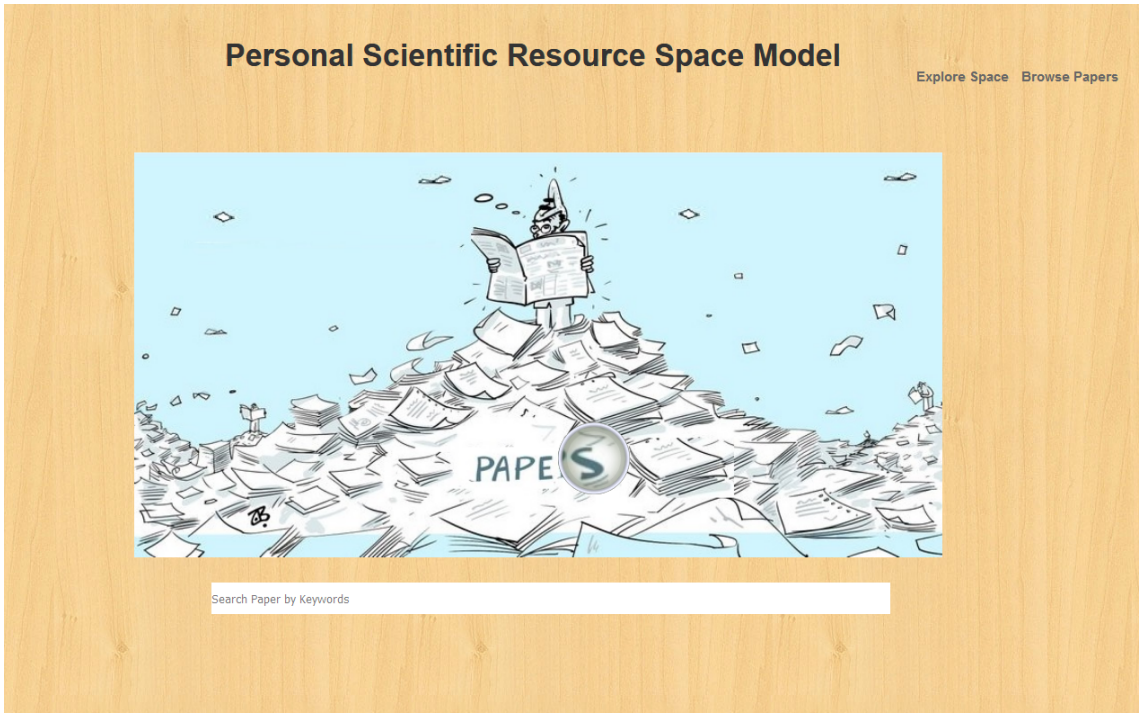


Figure 4.1: The home page of scientific resource space.

The home page of a scientific resource space is designed as Figure 4.1, which contains three functional elements: (1) a text box in the middle of the page receives the user input of the query keywords in order to activate the content retrieval function based on the keyword matching; (2) a hyperlink button "Explore Space" links the home page to the main page of space exploring; (3) a hyperlink button "Browse Papers" links the home page to the page of paper browsing.

The space exploring page in Figure 4.2 displays the multi-dimensional visualization of a whole scientific resource space and enables the space exploring functions. The page is divided into two parts. The left part uses the sunburst chart to show a unified view of dimensions and coordinates in a resource space, including four macro dimensions and three micro dimensions with their hierarchical coordinate systems. Coordinates of the same level are represented by one ring with the innermost circle as the top of the hierarchy. When users click at a specific coordinate, the sunburst chart will focus on this selected coordinate and rearrange the sunburst chart. The right part uses the tree diagram to show a coordinate tree. When users click at a node in the tree, the child nodes will automatically expand or shrink, which allows users to

navigate the tree along paths of interest. The root of the tree will change with the selected coordinate in the left sunburst chart.

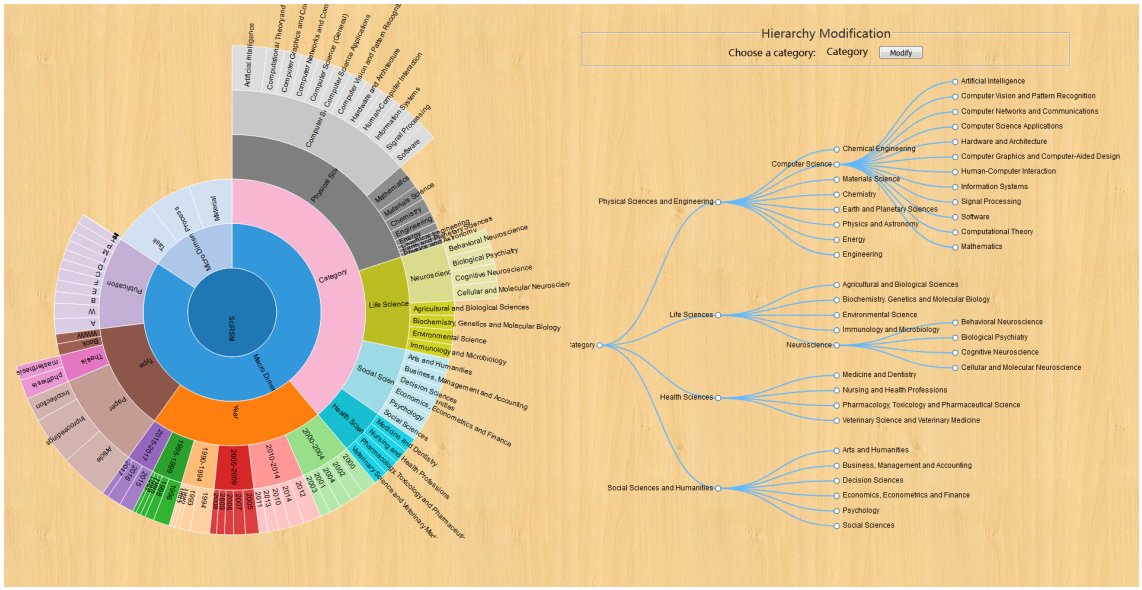


Figure 4.2: The main page of space exploring.

In addition, another important function that the space exploring page supports is to modify the category hierarchy, the ScienceDirect taxonomy on the macro dimension of Category in a scientific resource space. The hierarchy maintenance is activated by the button of "Modify" in the right side of the page. The hierarchy is modified according to the AMHC approach proposed in Chapter 2 of this thesis. When AMHC approach is finished, it will show up the modification page.

The hierarchy modification page shows the optional modification operations for users to select as shown in Figure 4.3. It includes three types of operations: (1) Pull-up operation: pull up category Mathematics to its upper level; (2) Merge operation: merge category Artificial Intelligence and category Computer Vision and Pattern Recognition; (3) Split operation: split category Artificial Intelligence into several finer subcategories. Users can accept or reject each modification suggestion. If users accept a modification operation, the related category names in the right tree diagram will be shown in bold. If users click the reject button, the category names will be restored originally. The apply button activates all selected modification operations and meanwhile the tree diagram shows the modified category hierarchy.

When users double click on a specific category node in the tree diagram, it will go into the corresponding category and display the resource browsing page of this category. For example, if users double click on a leaf node of Natural Language Processing, it will jump to the resource browsing page of the category Natural Language Processing shown in Figure 4.4.

Figure 4.4 shows the resource browsing page in the prototype system of scientific resource space, which is also divided into two parts. The left side is a navigation bar and the right side is a single resource view. The navigation bar displays the resources (scientific articles) in subcategories in separate blocks. Taking the category *Natural Language Processing* for exam-

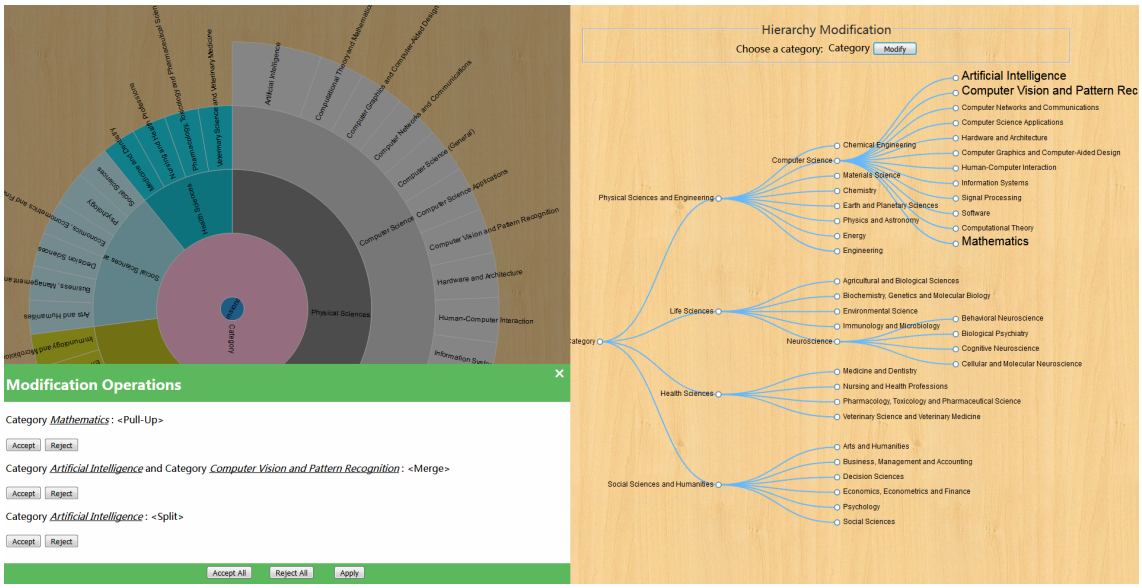


Figure 4.3: The hierarchy modification page.

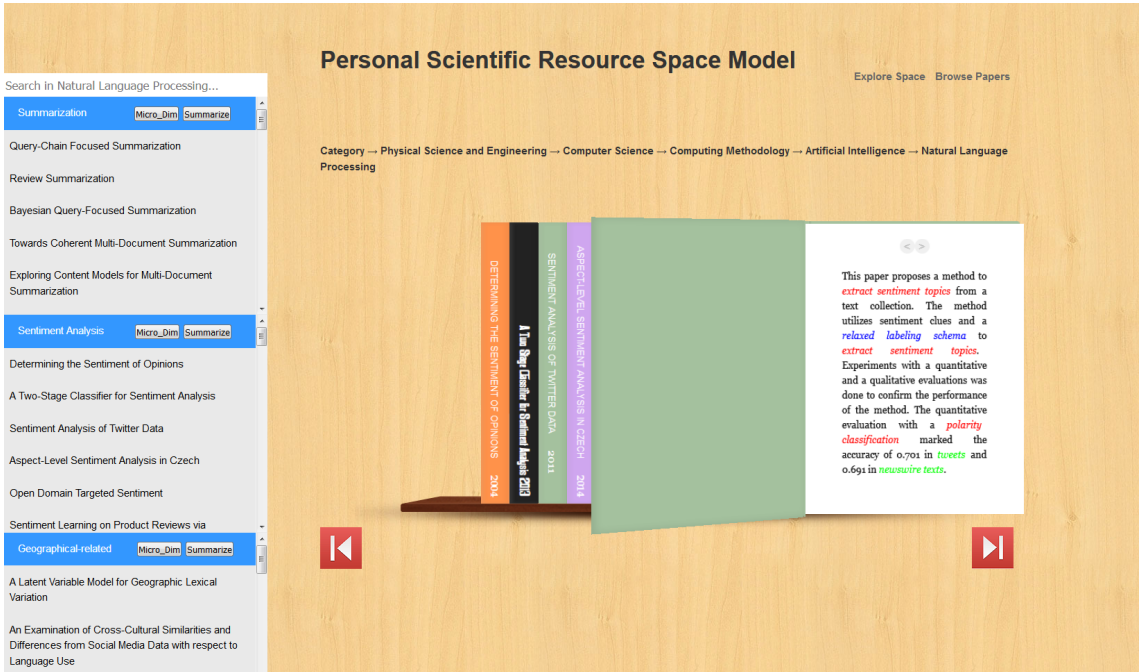


Figure 4.4: The resource browsing page.

ple, the resources can be further classified into three subcategories: *Summarization*, *Sentiment Analysis* and *Geographical-related NLP tasks*. The navigation bar lists the scientific articles in three subcategories, where each subcategory is in one block. When users select a specific article in the navigation bar, the right resource view will open it and show the content of this articles. In the navigation bar, each subcategory has two buttons: Micro-Dim button and Summarize button. The Micro-Dim button is to trigger the generation of a 3-dimensional micro space for this subcategory and the Summarize button is to generate a summary of articles in this subcategory. Figure 4.5 shows the user interface of the 3-dimensional micro space for the category Summarization triggered by the Micro-Dim button. The micro space enables manipulations, such as rotation, scaling and focus, to provide a global resource view for browsing a category. Figure 4.6 shows the user interface of the summary for the category Summarization triggered by the Summarize button.

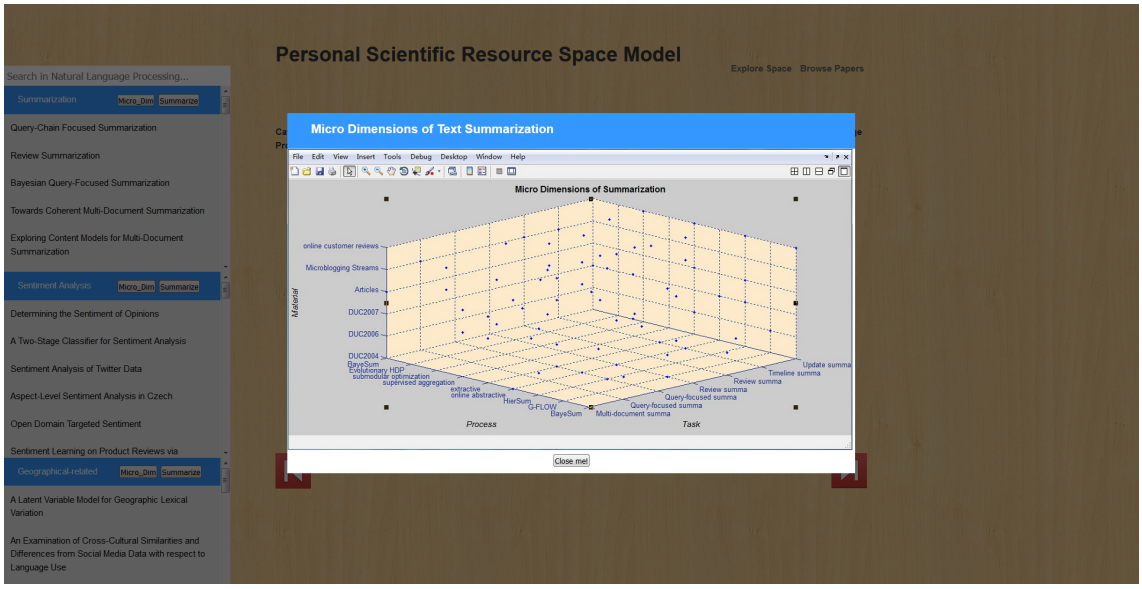


Figure 4.5: The micro space page triggered by the Micro-Dim button.

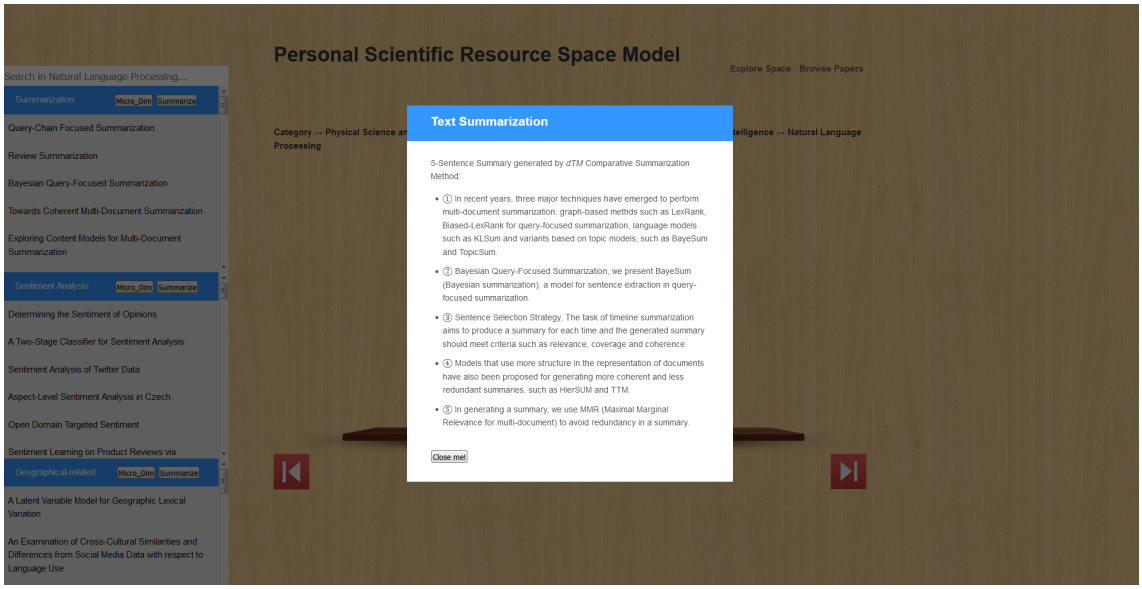


Figure 4.6: The summary page triggered by the Summarize button.

The right side of a resource browsing page in Figure 4.4 is a resource view which displays articles in bookshelves. The red forward (Next) and backward (Previous) buttons are used to update a batch of articles and a batch consists of 12 articles. When users click on an article in the navigation bar, the selected article will be displayed by a single resource view.

The single resource view provides users an easy way to read a single scientific article, which characterized by the ability to distinguish different types of keywords in different colours. In Figure 4.4, red indicates task or problem keywords (e.g. extract sentiment topics, polarity classification and timeline summarization etc.), blue labels process or method keywords (e.g. G-FLOW, BayeSum and HDP etc.), and green marks material or data keywords (e.g. DUC dataset, tweets and newswire texts etc.). This facilitates reading and understanding of running texts in people's daily lives.

Figure 4.7 and Figure 4.8 show the 3-dimensional micro spaces for the category Sentiment Analysis and category Geographical-related NLP Tasks respectively based on the concept hierarchy generation approach proposed in Chapter 3 of this thesis.

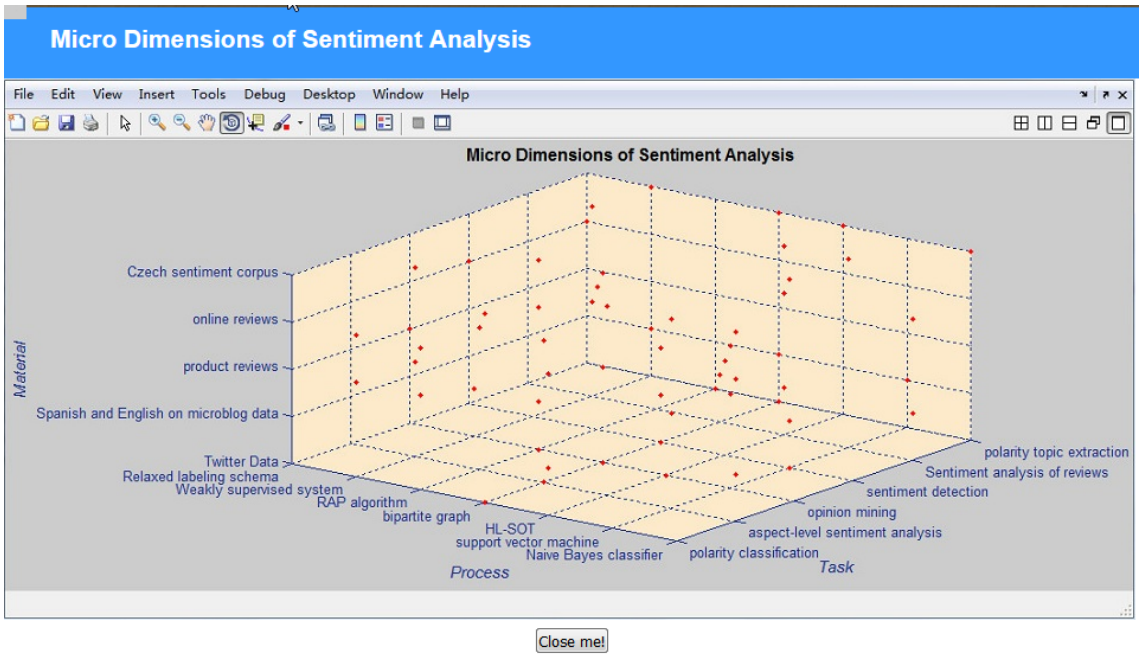


Figure 4.7: The 3-dimensional micro-space of Sentiment Analysis category.

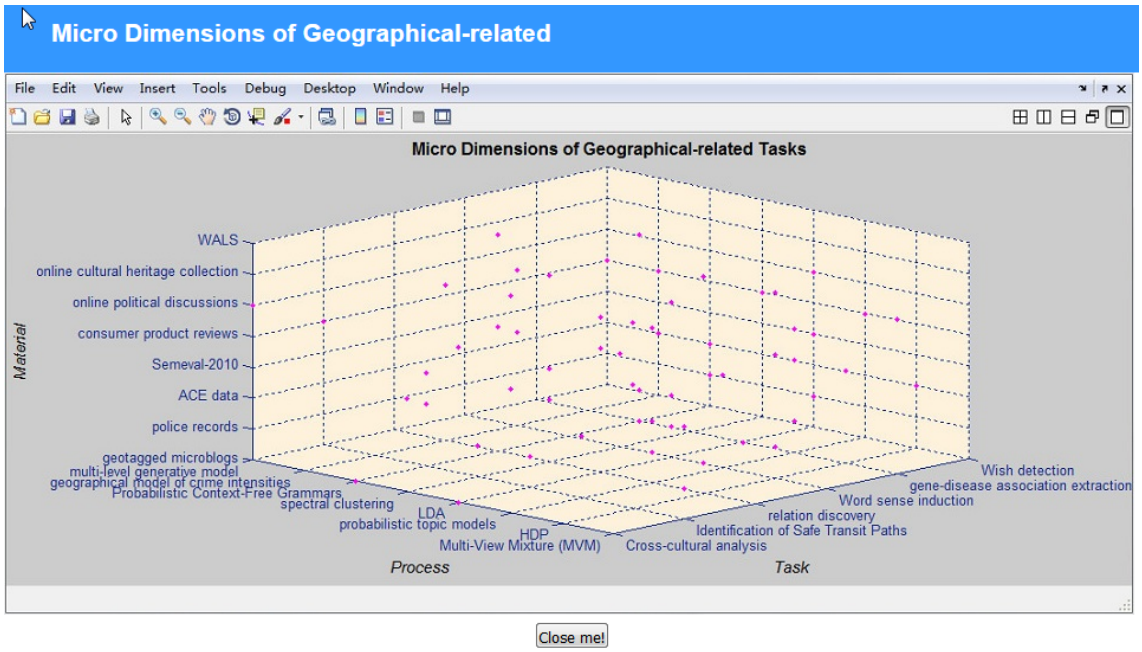


Figure 4.8: The 3-dimensional micro-space of Geographical-Related category.

5 Comparative Summarization Service in Scientific Resource Space

5.1 Motivations of Scientific Comparative Summarization

With the rapidly expanding of disciplines, the boundaries between different subjects are becoming increasingly blurred. The interconnected nature of real-world applications brings more cross-field research problems leading to a much closer relationship between research subjects. Real-world challenges require researchers to quickly get acquainted with knowledge in other areas. Another reason of absorbing knowledge from different subjects is to acquire a comprehensive understanding of some general models, theories and technologies to inspire their study.

Comparative summarization for scientific articles has real applications in scientific information retrieval. It can facilitate the comparison between different technique routes or between different research problems. For example, imagine a requirement from a researcher in summarization area who is familiar with topic models wants to focus his research on opinion summarization. His current interest would be on the specific knowledge of sentiment analysis and how topic model helps with sentiment analysis, while the common background knowledge, such as topic model and basic NLP technologies, would be undesired. The real-word demanding is hard to satisfy by generic text summarization methods due to the difficult in removing the common background knowledge, which encourages the study of comparative summarization for scientific papers between multiple subjects.

Therefore, the comparative summarization aims to summarize the differences among document groups (D. Wang, Zhu, Li, & Gong, 2012). Apparently, the core is to compare different topics and find unique characteristics for each document group. Our intent is to apply differential topic model (dTM) to comparative summarization and the intuition behind is that we want to model group-specific topics to capture unique word usage for summarizing the distinctness of a group.

5.2 Comparative Summarization based on Coordinate Partition

A scientific resource space provides a series of services on scientific articles to help users get quick access to useful information. Text summarization is to generate a short and concise summary that conveys the most important ideas from an original document, which enables

readers more easily to get general information of interest. This chapter focuses on summarization service and proposes scientific comparative summarization based on the concept of *coordinate partition* in resource space model, which help users better understand the differences between document groups.

Reviewing the definition of *coordinate partition* in section 1.2, we know that for any coordinate C on an axis X' in a resource space, resources defined by coordinate C can be partitioned by another axis X other than X', and the coordinate partition on C produces n classes corresponding to n coordinates $\{C_1, C_2, ..., C_n\}$ on axis X. The coordinate C is called the original coordinate and the axis X is called the partition axis.

In a scientific resource space, the summary based on the concept of *coordinate partition* is produced by first choosing an original coordinate and a partition axis, and then conducting the coordinate partition to classify the resources under the original coordinate into several categories on the partition axis, and finally generating a summary for each category. Reconsidering the example in Section 5.1, the comparative summarization service based on coordinate partition in scientific resource space can easily generate the satisfactory summary by first choosing the "Topic Model" (in the Process axis) as the original coordinate and then using Task dimension as the partition axis to partition resources under the original coordinate, finally the summary generated for "Sentiment Analysis" (a coordinate in the Task axis) is what the researcher is interested in. This summary includes contents that how topic model is applied to sentiment analysis and meanwhile excludes contents of background knowledge on topic models.

The summarization based on the *coordinate partition* in a scientific resource space is a multi-document summarization for comparing the differences between categories. However, it is difficult to use generic multi-document summarization methods to produce the category summary based on the *coordinate partition* in a scientific resource space, because generic summarization methods summarize important information that is delivered in most of documents. When summarizing with the generic summarization methods, sentences talking about the common theme are likely to be selected, which leads to the occurrence of common information in each category summary.

The summarization service based on the coordinate partition in a scientific resource space aims at comparing different categories and capturing the distinctness of each category to form comparative summaries. Thus, it requests particular comparative summarization methods that try to contain more unique content on category-specific themes and reduce content on common themes. This chapter explores differential topic models to generate comparative summaries for scientific articles based on coordinate partition in a scientific resource space.

5.3 Related Work

This section introduces the related work on automatic text summarization and topic models for documents comparison.

5.3.1 Multi-document Summarization

Existing multi-document summarization can be either extractive or abstractive. An extractive summarization method is to select important sentences from original documents and then concatenate them into a shorter form expressing the gist of the original documents. "Important" content is defined as frequent or favourably positioned content. Sentences are scored based on their statistical features to reflect their importance. In contrast, abstractive summarization methods try to understand the main concepts and then construct sentences whose fragments expressing the concepts come from different source sentences. Abstractive technique uses linguistic methods to analyse and interpret texts and then finds new concepts and expressions to best describe it. In this way, it can generate a new shorter summary that conveys the most important ideas from the original documents.

Our work focuses on the extractive techniques which involve in assigning saliency scores to sentences and extracting high-scored sentences in a greedy manner to construct a summary (Radev, Jing, & Budzikowska, 2000; Mihalcea & Tarau, 2004; Wan, Yang, & Xiao, 2007; Cai, Li, Ouyang, & Yan, 2010).

The centroid-based method, first proposed by (Radev et al., 2000), is one of the most popular extractive summarization methods. It uses cluster centroids to produce summaries. Documents are represented by $TF \times IDF$ vector. The first step is to use agglomerative clustering algorithm to group documents on the same event and then compute the centroids for each cluster. Centroids can be regarded as pseudo-documents (bag-of-words) that statistically represent a cluster of documents. The second step is to use the centroid to measure the topical centrality of each sentence in a cluster. Two metrics are defined: *cluster-based relative utility* which measures how relevant a particular sentence is to the topic of the entire cluster, and *cross-sentence informational subsumption* which measures the redundancy of selected sentences. MEAD is an implementation of the centroid-based extractive summarization approach developed by (Radev, Jing, Styś, & Tam, 2004). Due to the absence of the complicated language generation process, the centroid-based approach is technically simple and thus always served as a baseline method for experimental comparison.

Topic-driven summarization is a special task for multi-document summarization. It requires combining query-relevance with information-novelty when generating summaries. Carbonell and Goldstein pioneered the study of this task by proposing the maximal marginal relevance (*MMR*) measure (Carbonell & Goldstein, 1998), which rewards relevant sentences and penalizes redun-

dant ones by a linear combination of two similarities: Sentence-Query similarity and Sentence-Sentence Similarity. MMR strives to reduce redundancy while maintaining query relevance in sentence selection. For summarization, they select top ranking sentences by MMR and organize the sentences in their original order within the documents. The experimental results have shown that MMR performs better with longer documents due to the reduction of content repetition. Topic-driven summarization has its appealing prospect that generating summaries taking user preference into consideration, because different users need different summarization of the same document.

Graph-based ranking techniques such as TextRank (Mihalcea & Tarau, 2004) and LexPageRank (Erkan & Radev, 2004) have been widely used in extractive summarization. TextRank and LexPageRank resemble to HITS (Page, Brin, Motwani, & Winograd, 1999) and PageRank (Kleinberg, 1999) to rank sentences. A mutual rank algorithm is proposed to simultaneously summarize documents and extract keywords (Wan et al., 2007). A bi-gram based supervised method is proposed for extractive document summarization in ILP framework (C. Li, Qian, & Liu, 2013). CollabRank (Wan & Xiao, 2009) uses a collaborative approach to extract key phrases in a single document. A reinforcement approach to multi-document summarization by simultaneously ranking and clustering sentences is proposed in (Cai et al., 2010).

Extractive techniques may not be effective due to the lack of deep understanding of texts, while abstractive methods understand concepts and merge facts from different sentences, thus they are more likely to produce summaries resembling to human-written counterparts. However, researches on this route are still immature and less popular due to the difficulties in deep text analysis and understanding. We summarize the following four main methods: (1) Information fusion based methods (Filippova & Strube, 2008; Filippova, 2010; Banerjee, Mitra, & Sugiyama, 2015) generate new sentences of common information by multi-sentence fusion; (2) Information extraction based methods (Genest & Lapalme, 2012; Bing et al., 2015; W. Li, 2015) generate new sentence through information extraction techniques; (3) Sentence paraphrasing based methods (Nenkova, 2008; Siddharthan, 2011) try to improve quality of summary by noun phrases rewriting and co-reference resolution; (4) Sequence-to-sequence learning based methods (Nallapati, Zhou, Santos, Gulcehre, & Xiang, 2016; Gu, Lu, Li, & Li, 2016; See, Liu, & Manning, 2017) model the summarization process as an end-to-end sequence generation process based on large training corpus.

A multi-dimensional summarization methodology was proposed to transform the paradigm of traditional summarization research through multi-disciplinary fundamental exploration on semantics, dimension, knowledge, computing and cyber-physical society (Zhuge, 2016).

5.3.2 Comparative Summarization

Unlike the generic summarization that summarizes the common information in document collection, the comparative summarization aims to summarize the differences among document groups. Wang et al. proposed a discriminative sentence selection method to generate summary by selecting sentences in a greedy manner to minimize the generalized variance of a covariance matrix using a multivariate normal model (D. Wang et al., 2012). Shen and Li proposed a method by building the sentence graph for each document group and extracting a complementary minimum dominating set on each graph to form a discriminative summary (Shen & Li, 2010).

5.3.3 Update Summarization

The most similar task to comparative summarization is update summarization, which aims to detect and summarize novel information in a document set B under the assumption that users have already learnt the documents in set A, where documents in A chronologically precede the documents in B. The update summarization has been well studied. Most existing methods solve it as a redundancy removal problem by adding functionality to remove redundant sentences using filtering rules (Fisher & Roark, 2008), Maximal Marginal Relevance (Boudin, El-Bèze, & Torres-Moreno, 2008), or graph-based algorithms (Shen & Li, 2010; W. Li, Wei, Lu, & He, 2008).

More related to this thesis is the work of a topic-model based update summarization approach *DualSum* (Delort & Alfonseca, 2012), which learns a general background distribution across the corpus and a document-specific distribution for each document, but also learns two collection-specific distributions for each pair of update collection and base collection: the joint topic distribution and the update topic distribution. We modify *DualSum* as a baseline for evaluation in Section 5.5.2.

5.3.4 Topic Models for Documents Comparison

The other type of related work is the comparison of documents. Most existing studies for this goal focus on topic models to discover common and specific themes among document collections, referred to as cross-collection topic models (Paul, 2009). This idea was first explored with an initial topic model *PLSI* (Zhai, Velivelli, & Yu, 2004), and later improved with *LDA* topic model (Blei, 2012) which inspires our *dTM-Dirichlet* model. There are a number of real-world applications extending cross-collection topic models in different scenarios (Ahmed & Xing, 2010; P. Li, Wang, Gao, & Jiang, 2011). For example, Paul and Girju employed cross-collection LDA (*cc-LDA*) for cross-cultural analysis of blogs and forums (Paul & Girju, 2009) and later they proposed a two-dimensional topic-aspect model (*TAM*) to jointly discover topics and aspects in scientific literature (Paul & Girju, 2010). The common idea behind these cross-collection topic models is that using latent topics capture the common and unique word usage among document

collections. Cross-collection topic models neglect the correlations between each collection-specific topic and the common background topic, thus make it insufficient to capture differential word usage. More importantly, the correlations are the essence of the differential topic models.

5.4 Comparative Summarization based on Differential Topic Models

Comparative summarization aims at summarizing the differences among document groups (D. Wang et al., 2012). The core is to compare different topics and find unique characteristics for each document group. The main motivation of our method is to apply differential topic models (*dTM*) to comparative summarization and model the group-specific topics to capture the unique word usage for characterising documents in the same group.

We first propose a probabilistic generative model *dTM-Dirichlet* to model the group-specific word distributions to capture the unique word usage for each document group. However, *dTM-Dirichlet* is not a truly differential topic model and it suffers from the problems of high inference cost, over-parameterization and lack of sparsity. Evolving from the idea of *SAGE* (Eisenstein, Ahmed, & Xing, 2011), we develop *dTM-SAGE* to make the word probability distributions for each document group to share a common background word distribution and explicitly models how words are used differently in each group from the background word distribution.

To generate *dTM*-based comparative summaries, we propose two sentence-scoring methods to measure the sentence discriminative capacity and a greedy sentence selection method to select the most distinguished sentences, which meets the requirements of summarization service in scientific resource space.

5.4.1 Differential Topic Models

The differential topic models are developed for comparative summarization. We first develop a simple probabilistic generative model, *dTM-Dirichlet*. Evolved from *dTM-Dirichlet*, *dTM-SAGE* is developed by modelling the correlations as additive relation between the group-specific deviations and a background word distribution, which enables to capture more salient group-specific words and bypass the problems of high inference cost, over-parameterization and lack of sparsity.

To illustrate *dTM*, we first define some notations to express a document corpus C. Let G be the number of groups in the corpus, M_g be the number of papers in group g and $N_{g,m}$ be the number of words in paper m. A word $w_{g,m,n}$ representing the n^{th} word in paper m of group g is a discrete observed variable, defined to be an item in the vocabulary list of the whole corpus.

5.4.1.1 *dTM-Dirichlet* Model

dTM-Dirichlet model is a simplified version of cross-collection LDA (*ccLDA*) (Paul, 2009) for comparing multiple text collections. *dTM-Dirichlet* builds two types of word model. One is for each document group g, in which there is a group-specific content word model ϑ_g that emits discriminative words for the group g. The other type is a superset of group-independent word models $\varphi_k (k = 1, ..., K)$ that generates either background words shared by all document groups or salient words occurring in several documents of different groups. Reconsidering the scenario in Section 5.1, the group-independent word model represents two classes of words, i.e. the background words like topic model that are shared by almost all papers; and the salient words like NP chunk and dependency parsing that only occur in several papers of different groups.

We focus on the group-specific word model for comparative summarization. Since background words and salient words provide no group-specific knowledge, they are not distinguished in *dTM-Dirichlet*. Following probabilistic topic models, we assume that word models φ_k and ϑ_g are multinomial distributions over words, drawn from uniform Dirichlet distribution (Dir) with priors α_φ and α_ϑ.

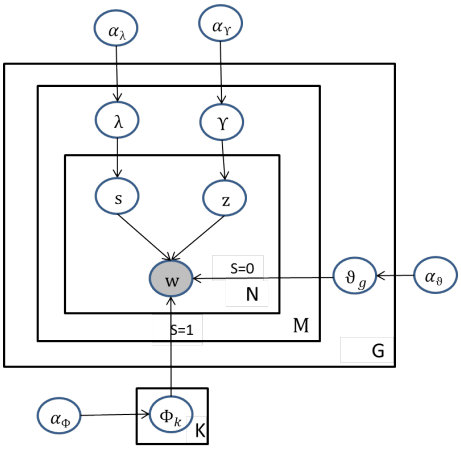


Figure 5.1: dTM-Dirichlet Model Graph Representation.

As shown in Figure 5.1, *dTM-Dirichlet* associates each document a topic distribution $\gamma_{g,m} \sim Dir(\alpha_\gamma)$, and the topic assignment variable $z_{g,m,n}$ for each word in the document thus can be multinomially sampled from $\gamma_{g,m}$, denoted as $z_{g,m,n} \sim Multi(\gamma_{g,m})$. Besides a topic variable $z_{g,m,n}$, each word is also assigned with a binary variable $s_{g,m,n}$ that indicates whether the word is a group-independent topic word ($s_{g,m,n} = 1$) or a group-specific content word ($s_{g,m,n} = 0$). Each document has a group-specific word controller $\lambda_{g,m} \sim Beta(\alpha_\lambda)$, which reflects the proportion of group-specific content in a document. The binary variable $s_{g,m,n}$ is sampled from a Bernoulli test with the probability of $\lambda_{g,m}$.

Formally, the generative process of *dTM-Dirichlet* model for a corpus C divided into G document groups is shown in Table 5.1. When $s_{g,m,n} = 1$, the sample of word $w_{g,m,n}$ is from the

group-independent topic word distribution $\varphi_k(k = 1, ..., K)$ which is identical to LDA. When $s_{g,m,n} = 0$, the sample of word $w_{g,m,n}$ is directly drawn from the group-specific content word distribution ϑ_g that is independent from the document's topic distribution $\gamma_{g,m}$.

As for the time complexity of the *dTM-Dirichlet* model, conventional Gibbs sampling methods for inference requires $O(K)$ operations per sample where K is the number of group-independent topics in the model. Thus the time complexity is $O(G \times M \times N \times K)$ where G is the number of groups, M is the average number of documents in each group and N is the average number of words in each document. FastLDA proposed an method which draws equivalent samples but requires on average significantly less then K operations per sample (Porteous et al., 2008).

Table 5.1: The generative process of dTM-Dirichlet.

1. For each topic k, where $1 \leq k \leq K$
   a. Draw $\Phi_k \sim Dir(\alpha_\Phi)$
2. For each document group g, where $1 \leq g \leq G$
   a. Draw $\vartheta_g \sim Dir(\alpha_\vartheta)$
   b. For each document m in group g, where $1 \leq m \leq M_g$
      1) Draw $\lambda_{g,m} \sim Beta(\alpha_\lambda)$
      2) Draw $\gamma_{g,m} \sim Dir(\alpha_\gamma)$
      3) For each word n, where $1 \leq n \leq N_{g,m}$
         a) Draw $s_{g,m,n} \sim Bern(\lambda_{g,m})$
         b) If $s_{g,m,n} = 1$ (a group-independent topic word)
            A. Draw a topic assignment $z_{g,m,n} \sim \gamma_{g,m}$
            B. Draw a word $w_{g,m,n} \sim \Phi_{z_{g,m,n}}$
         c) If $s_{g,m,n} = 0$ (a group-specific content word)
            A. Draw a word $w_{g,m,n} \sim \vartheta_g$

dTM-Dirichlet uses group-specific word distributions to capture the differential lexicon usage of document groups. However, *dTM-Dirichlet* is not a truly differential topic model, which requires the development of *dTM-SAGE* for comparative summarization.

5.4.1.2 *dTM-SAGE* Model

When generating topics for multiple document collections, LDA-style generative models associate a multinomial distribution with each document group, which is the same as how we model the group-specific content words in *dTM-Dirichlet* model.

In contrast, Sparse Additive Generative model (*SAGE*) (Eisenstein et al., 2011) provides an alternative way to LDA by endowing each document group with a model of the deviation in log-frequency from a constant background distribution, which brings three advantages: (1) a sparsity-inducing prior can be applied to limit the number of terms whose probability diverges

from the background term frequencies; (2) multi-facets latent variables can be easily combined by adding each facet component together to reduce the inference cost; (3) it is redundant to learn unique probabilities for high-frequency background words of each group. Modelling the deviation of each group-specific word distribution cancels the relearn process for the background words.

We propose *dTM-SAGE* which explicitly models the deviation in log-frequency of each group-specific word distribution from a background lexical distribution. *dTM-SAGE* also builds word models for group-independent topic words and group-specific content words. The group-independent topic words consist of background topic words and salient topic words.

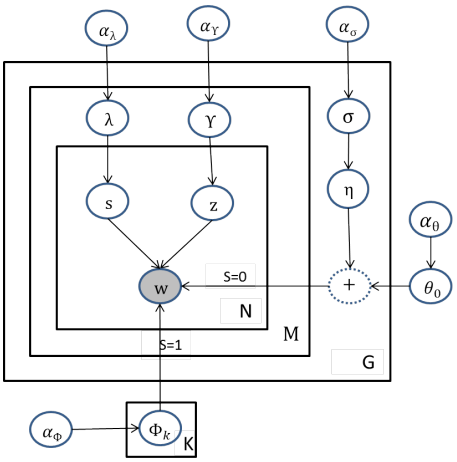


Figure 5.2: dTM-SAGE Model Graph Representation.

dTM-SAGE models two types of group-independent words separately: as shown in Figure 5.2, the salient topic words captured by $\varphi_k(k = 1, ..., K)$ and the background topic words captured by ϑ_0. The word models φ_k and ϑ_0 are multinomial distributions drawn from uniform Dirichlet prior with parameter α_φ and α_ϑ. To enable ϑ_0 to capture real background topic words shared by all document groups, we replace the constant background distribution in *SAGE* with a latent distribution learnt by MAP estimation using a Newton optimization.

The major difference between *dTM-SAGE* and *dTM-Dirichlet* is how the group-specific topics are generated. In Figure 5.2, each document group g has a group component vector η_g representing the deviations in log-frequencies from the background distribution ϑ_0. The group-specific topic is represented by log frequency deviations rather than word probabilities. Given the background distribution ϑ_0 and the group component vector η_g, the group-specific topic distribution ϑ_g for each word in a document in the group g, denoted by $\vartheta_g \propto exp(\vartheta_0 + \eta_g)$, is computed by equation 5.1:

$$p(w|\vartheta_0, \eta_g) = exp(\vartheta_0 + \eta_g) / \sum_v exp(\vartheta_{0,v} + \eta_{g,v}) \tag{5.1}$$

where g indexes the group component vector and v indexes the term in the corpus vocabulary.

Following *SAGE*, we ignore covariance between terms. For each term v, $\eta_{g,v}$ is drawn from a zero-mean Gaussian distribution $N(0, \sigma_{g,v})$, where the variance $\sigma_(g, v)$ is drawn from the Expo-

nential distribution parameterized by α_σ. The compound model $\int N(\eta; 0, \sigma) Exponential(\sigma; \alpha_\sigma) d\sigma$ is equivalent to a zero-mean Laplace prior on η which has the capacity of inducing sparsity and meanwhile permitting large degrees of deviations.

In *dTM-SAGE*, ϑ_0, η_g and σ are treated as latent variables. We use MAP to estimate ϑ_0, η_g and develop variational inference on σ. The generative process of *dTM-SAGE* is shown in Table 5.2. See Appendix A for more inference details.

As for the time complexity of the *dTM-SAGE* model, the number of operations for each document m in group g is $O(N^*_{g,m} \times K)$, where $N^*_{g,m}$ is the unique number of words in the document and K is the number of group-independent topics in the model. Thus the time complexity of the mean-field variational inference for the *dTM-SAGE* model is $O(G \times M \times N^* \times K)$ where G is the number of groups, M is the average number of documents in each group and N^* is the average number of unique words in each document.

Table 5.2: The generative process of dTM-SAGE.

1. Draw $\vartheta_0 \sim Dir(\alpha_\vartheta)$
2. For each topic k, where $1 \leq k \leq K$
   a. Draw $\Phi_k \sim Dir(\alpha_\Phi)$
3. For each document group g, where $1 \leq g \leq G$
   a. For each term v, where $1 \leq v \leq V$
      1) Draw $\sigma_{g,v} \sim Exponential(\alpha_\sigma)$
      2) Draw $\eta_{g,v} \sim N(0, \sigma_{g,v})$
   b. Set $\boldsymbol{\theta_g} \propto exp(\boldsymbol{\vartheta_0} + \boldsymbol{\eta_g})$
   c. For each document m in group g, where $1 \leq m \leq M_g$
      1) Draw $\lambda_{g,m} \sim Beta(\alpha_\lambda)$
      2) Draw $\gamma_{g,m} \sim Dir(\alpha_\gamma)$
      3) For each word n, where $1 \leq n \leq N_{g,m}$
         a) Draw $s_{g,m,n} \sim Bern(\lambda_{g,m})$
         b) If $s_{g,m,n} = 1$ (a group-independent topic word)
            A. Draw a topic assignment $z_{g,m,n} \sim \gamma_{g,m}$
            B. Draw a word $w_{g,m,n} \sim \Phi_{z_{g,m,n}}$
         c) If $s_{g,m,n} = 0$ (a group-specific content word)
            A. Draw a word $w_{g,m,n} \sim \vartheta_g$

5.4.2 Comparative Summary Generation

To summarize differences among document groups, we rely on group-specific topics ϑ_g to select most discriminative sentences for summary generation. This section introduces the sentence scoring and the sentence selection techniques developed for *dTM*-based comparative summariza-

tion.

5.4.2.1 Sentence Scoring

Both *dTM-Dirichlet* and *dTM-SAGE* model the group-specific word distributions ϑ_g to capture the unique content in each document group. For *dTM-SAGE*, we can also get a corpus background topic distribution ϑ_0 that reflects the common themes shared by all groups. To measure the sentence discriminative capacity, we develop two sentence scoring methods: one is based on the word discriminative scores and the other measures the difference of the probabilities that a sentence is generated from a group-specific topic distribution and the background topic distribution.

(1) Sentence scoring based on the word discriminative scores Given a set of group-specific word distributions ϑ_g $(1 \leq g \leq G)$, we define the calculation of the word discriminative score $DSW(v, g)$ of a term v to a group g in equation 5.2:

$$DSW(v, g) = \sum_{g' \neq g} (\vartheta_{g,v} - \vartheta_{g',v}) / (\sqrt{\sum_{g} \vartheta_{g,v}^2} + \epsilon) \tag{5.2}$$

where ϵ is a small number (set to 0.05) to avoid the error of division by zero. Larger value of the word discriminative score indicates more discriminative ability the word has. The intuition is that a word more likely to occur in a particular group and less likely to occur in other groups tends to be more discriminative.

Thus, the discriminative capacity of a sentence s to a group g $DCS\_dsw(s, g)$ is the average over the word discriminative scores, computed as equation 5.3:

$$DCS\_dsw(s, g) = \sum_{w \in s} DSW(w, g) / len(s) \tag{5.3}$$

(2) Sentence scoring based on the sentence generation probability The other method to measure the discriminative capacity of a sentence relies on the likelihood that the sentence is generated from a group-specific distribution and the background topic distribution. Its design is motivated by the idea that a word is more discriminative if it occurs more often in a group-specific topic and occurs rarely in the shared background topic.

Given a topic-word distribution ϑ, the probability of a sentence s generated from ϑ is computed by equation 5.4:

$$logP(s|\vartheta) = \sum_{w \in s} log\vartheta_w \tag{5.4}$$

Given a set of group-specific word distributions ϑ_g $(1 \leq g \leq G)$ and a background topic distribution ϑ_0, the discriminative capacity of a sentence s to a group g, represented by $DCS\_dgp(s, g)$, is calculated as the difference of sentence generative probabilities in equation 5.5:

$$DCS\_dgp(s, g) = ulogP(s|\vartheta_g) - (1 - u)logP(s|\vartheta_0) \tag{5.5}$$

where u is a balance factor trading off between group-specific words and background words.

5.4.2.2 Sentence Selection

To select discriminative sentences to form group summary, we use different sentence selection methods according to sentence scoring techniques.

For the sentence scoring based on the word discriminative scores, we first rank the sentences according to the sentence discriminative capacity score DCS_dsw. Then we select a sentence with the highest score if it satisfies the redundancy constraint that indicated by a cosine similarity threshold (empirically set to 0.8).

For the scoring based on difference sentences generative probabilities, suppose that we have a set of candidate sentences S to form a summary for group g and we want to select k sentences denoted as S_k. A greedy sentence selection schema is proposed to build S_k by iteratively choosing a j^{th} sentence that currently has the maximum sentence discriminative capacity score DCS_dgp, formulated by equation 5.6:

$$s_j^* = \underset{s_j \in S \setminus S_{j-1}}{\arg\max}\ DCS\_dgp(s, g) \tag{5.6}$$

In order to discourage redundancy, after select one sentence, we update the group-specific topic distribution ϑ_g by setting $\vartheta_{g,w} \propto \vartheta_{g,w}^2$ for each word w in the selected sentence s_j^*. Sentences are selected in this manner until reaching the summary limit.

5.5 Experiment and Results

5.5.1 Data Collection and Annotation

Comparative summarization is not a new task. However, to our best knowledge there is no public benchmark data set available. For collecting experiment data, we choose three tasks in NLP: summarization (*SUMMA*), sentiment analysis (*SA*) and geographical NLP tasks (*GEO*) to form three document groups. To make different groups share more salient themes, we focus on papers using probabilistic topic models.

Table 5.3: Information of dataset

Group	Keywords		$\|D\|$	$\|S\|$
	Title	Plain Text		
SUMMA	summarization	topic model	35	6636
SA	Sentiment	topic model	45	10239
GEO	N/A	topic model, geographical	49	8249

We collect 129 papers in total for the three groups from ACL Anthology Searchbench, which provides semantic, full text and bibliographic search for 28,000 papers in the ACL Anthology. For each group, we search with two types of keyword filters: plain text filter and title filter. Table 5.3 shows the general information of each document group, including the keywords, the number of documents |D| and the number of sentences |S|. To pre-process the dataset, we exclude

all tables, figures and formulas, remove stop words, perform stemming with Porter Stemmer, and prune words less than 5 times across the corpus. There are 3720 tokens after pre-processing.

We hire three PhD students in Aston University to annotate the dataset. After reading papers in each group, each annotator is asked to first pick out all discriminative sentences in each paper and then write reference summaries delivering the major differences for each group. Additional instructions are given to annotators: Each reference summary should be no more than 300 words; and the discriminative sentences should enable the judgment of which group the paper belongs to. Equipped with the annotated dataset, two parts of evaluations are performed: evaluation of differential topic models and evaluation of the summarization methods.

5.5.2 Evaluations on *dTM*

In this section, we compare *dTM-Dirichlet* and *dTM-SAGE* with other three topic models in terms of model perplexity and topic coherences listed in Table 5.4: (1) standard LDA topic model, which we run across the corpus and perform Newton optimization to update hyper-parameters; (2) *SAGE*, which a sparse additive generative model proposed in (Eisenstein et al., 2011), and the non-parametric Jeffreys prior make it parameter-free; (3) the variant of *DualSum*, which is proposed for update summarization (Delort & Alfonseca, 2012) and revised to perform comparative summarization by replacing pairs of collection-specific distributions with group-specific distributions. We implement the variant of *DualSum*, *dTM-Dirichlet* and *dTM-SAGE* models. Experimental settings are detailed below.

Settings for the variant of *DualSum*. The dirichlet priors for word distributions are empirically set to 0.1 and $\alpha_\lambda = (2.0, 2.0, 1.0)$ to encourage more words generated from the group-specific distributions and document-specific distributions.

Settings for *dTM-Dirichlet*. The dirichlet priors for word distributions α_ϑ and α_φ are set to 0.1. For other papramenters, we set the number of group-independent topics $K = 20$, the prior for the topic distribution $\alpha_\gamma = 50/K$, and the prior for the group-specific word controller $\alpha_\lambda = 2.0$. $Beta(2.0, 2.0)$ yields equal probabilities that words sampling from the group-specific distribution and the group-independent distributions.

Settings for *dTM-SAGE*. Parameters are set the same as those in *dTM-Dirichlet*: $\alpha_\vartheta = \alpha_\varphi = 0.1$, $K = 20$, $\alpha_\gamma = 50/K$ and $\alpha_\lambda = 2.0$. The variational distribution of the variance σ is $Gamma(\widetilde{a}, \widetilde{b})$ which is initialized as $\widetilde{a} = 10.0$ and $\widetilde{b} = 5.0$. The initialization for ϑ_0 and η are from the Uniform distribution $U(0, 1)$ and the Normal distribution $N(0, 0.5)$ respectively.

First, we investigate the model perplexity. Perplexity is a general measure for evaluating the generative ability of a probabilistic topic model. We compute the perplexity on a held-out test set, 20% of the original dataset. Note that we calculate the perplexity for all models except the variant of*DualSum*, since it models the document-specific distribution for each document and thus there is no natural way to assign probability to a new document. For the variant of

DualSum, we train the model on the whole dataset and report the results on the test set, though it by no means can reflect the generalization capacity of the model.

Perplexity results are listed in the first row in Table 5.4, from which we can see that the perplexity scores decrease by 7% and 13% respectively between *dTM-Dirichlet* and standard LDA and between *dTM-SAGE* and standard *SAGE*. The better results of differential topic models over the standard ones are due to the discrimination between group-specific topics and group-independent topics. Both *SAGE* methods outperform their counterparts of the Dirichlet-multinomial, because the sparsity-inducing prior enables *SAGE* to control sparsity adaptively without over-fitting (Eisenstein et al., 2011).

To check the quality of the generated group-specific topics, we investigate various topic coherence measures. The intuition behind the topic coherence measures is that words clustering into a single topic tend to co-occur in the same document. It has been previously verified that topic coherence score is highly correlated with human-judged topic coherence in many works. We rely on Palmetto library (Röder, Both, & Hinneburg, 2015), an online open source implementation, which offers a framework to calculate many coherence measures within a reference corpus of the English Wikipedia.

In our experiment, we compare three widely-used coherence scores over the five topic models: (1) C_A (Aletras & Stevenson, 2013), which is the pairwise comparison of the top words based on a context window of size 5; (2) C_V (Röder et al., 2015), which is a one-set segmentation of the top words based on a sliding window of size 110; (3) C_UCI (Newman, Lau, Grieser, & Baldwin, 2010), which is the pointwise mutual information (*PMI*) of all word pairs of the top words based on a sliding window of size 10.

We focus on the group-specific topics. For each group-specific topic-word distribution we get a word list containing the top-20 words and calculate the coherence scores for each word list. The topic coherence results in Table 5.4 are the average coherence scores of the three group word lists. The coherence scores are calculated within two reference corpus: the English Wikipedia (*Wiki*) and the original dataset (*Intra*). Table 5.5 shows the top 10 words selected by *SAGE*, *dTM-SAGE* and *dTM-Dirichlet* for the group SUMMA.

Table 5.4: Comparisons on perplexity and topic coherences of different models.

Measures	LDA	SAGE	Variant of DualSum	dTM-Dirichlet	dTM-SAGE
Perplexity	2218.37	2177.29	1564.04	2052.78	1891.10
C_A (Wiki)	0.098	0.143	0.130	0.138	0.147
C_V (Wiki)	0.321	0.334	0.344	0.360	0.355
C_UCI (Wiki)	-2.116	-1.917	-1.272	-1.495	-0.905
C_UCI (Intra)	-0.895	-0.849	-0.662	-0.661	-0.608

Main observations from Table 5.4 are concluded as follows:

1. The three differential topic models generally perform better than the standard *LDA* and

SAGE models on all coherence measures, which shows the advantage of our *dTM* models for distinguishing group-specific words and group-independent words;

2. *dTM-SAGE* consistently performs the best among all the five models in terms of C_A and C_UCI with the increase at least by 6.5% over *dTM-Dirichlet* and 8.2% over the variant of *DualSum*, which shows the advantage of *dTM-SAGE* in accurately ranking the group-specific words due to the essence of the differential word model;

3. *dTM-Dirichlet* outperforms the variant of *DualSum* with C_A and C_V, however, it performs nearly the same or even worse when measured with C_UCI.

Table 5.5: Top 10 words selected by different models.

SAGE	dTM-Dirichlet	dTM-SAGE
sentence, topic, query document, summary, word, generative, model, vertice, distribution	sentence, summary, document, topic, rouge, extract, score, select, multi, system	sentence, rouge, ilp, duc, tac, summary, timeline, lexrank, redundant, mead

In addition, words selected by *dTM-SAGE* (like rouge, lexrank, redundant) in Table 5.5 are more informative and discriminative than words selected by *SAGE* and *dTM-Dirichlet*.

5.5.3 Evaluations on Summarization

To evaluate the quality of the generated summaries, we compare our *dTM*-based comparative summarization methods with five other typical methods under ROUGE metrics (C.-Y. Lin & Hovy, 2003). Further, to check the discriminative ability of the comparative summaries, following the evaluation method of (D. Wang et al., 2012), we investigate the precision of the discriminative sentence selection.

In our experiment, we implement three types of summarization methods: (1) Generic baseline methods, including the centroid-based method (Radev et al., 2004), the graph-based method LexPageRank (Erkan & Radev, 2004) and the MMR-based method (Carbonell & Goldstein, 1998); (2) State-of-the-art comparative summarization methods, including the discriminative sentence selection (DSS) method (D. Wang et al., 2012) and the complementary dominating set (*CDS*) method (Shen & Li, 2010); (3) topic model based comparative summarization methods, which combine four different topic models with two sentence scoring strategies *DCS_dsw* and *DCS_dgp* defined in Section 5.4.2, including the basic *LDA* (dsw), the variant of *DualSum* (dsw), *dTM-Dirichlet* (dsw), *dTM-SAGE* (dsw) and *dTM-SAGE* (dgp). For each group, we select 20 sentences to form the final summary.

First, we examine the precision of the discriminative sentence selection. For each group we have 20 sentences in a summary and count how many sentences belong to the annotated

Table 5.6: Comparison of Rouge scores and precisions.

	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4	Precision
Baselines					
Centroid	0.23084	0.01867	0.21739	0.05672	0.383
LexPageRank	0.25334	0.02092	0.23822	0.06767	0.417
MMR	0.28272	0.02817	0.26333	0.08094	0.433
State-of-the-arts					
DSS	0.30898	0.03766	0.29346	0.09239	0.600
CDS	0.31749	0.03717	0.29047	0.09340	0.549
State-of-the-arts					
Basic LDA (dsw)	0.29812	0.03625	0.27940	0.08865	0.517
Variant of DualSum (dsw)	0.37445	0.04584	0.34542	0.11245	0.650
dTM-Dirichlet (dsw)	0.33024	0.06047	0.31388	0.12363	0.700
dTM-SAGE (dsw)	0.39173	0.06800	0.35764	0.12716	0.717
dTM-SAGE (dgp)	**0.42266**	**0.08801**	**0.38519**	**0.16205**	**0.750**

discriminative sentence set. Comparisons of the precision results of discriminative sentence selection by different methods are listed in the last column in Table 5.6. From the precision results, we find that: (1) our *dTM*-based comparative summarization methods can select over 70% discriminative sentences, which significantly outperform the state-of-the-art methods with a nearly 20% increase on the precision score; (2) All generic summarization methods perform rather worse due to different concerns on summarization resulting in the lack of discriminative ability of summaries.

We use ROUGE-1.5.5 toolkit to evaluate the quality of generated summaries by comparing them with human-written reference summaries. In our experiment, we limit the length of all summaries to 250 words and report the average ROUGE scores (F-Scores) on various summarization methods in Table 5.6.

According to Table 5.6, the following conclusions can be drawn:

1. Our *dTM*-based comparative summarization methods perform significantly better (paired t-test with $p<0.05$) than all the baselines, which demonstrates that targeting at a different goal for summarizing the general information among document groups, generic summarization methods are less applicable for comparative summarization, though by removing redundancy, MMR performs better than the other two baselines but still lags behind other summarization methods specifically proposed for comparative summarization;

2. Our *dTM-SAGE* comparative summarization methods significantly outperform (paired t-test with $p<0.05$) the other two state-of-the-art comparative summarization methods, which shows that summarizing differences by extracting group-specific topics is more effective than directly summarizing at the sentence level;

3. Both *dTM-SAGE* methods achieve better ROUGE scores than *dTM-Dirichlet*, which is ascribe to the advantage of a differential word model contributing to more informative and discriminative group-specific topics;

4. For *dTM-SAGE*, the greedy sentence selection schema based on DCS_dgp is more effective than simply ranking sentence with DCS_dsw.

We show an example of the summary generated for the group SUMMA by our *dTM-SAGE* and *dTM-Dirichlet* in Table 5.7. Looking into the summaries, we find that all sentences in both summaries are related to summarization but different in the degree of their discriminative ability. Apparently, the summary generated by *dTM-SAGE* is more specific and unique to summarization, while the summary generated by *dTM-Dirichlet* still contains some general information about topic models in sentence 2 and sentence 5.

Another observation is that the summary of *dTM-SAGE* tends to contain more salient group-specific terms that may not occur in most of group documents but still possess high discrimination, like "query-focused", "MMR" and "HierSUM". In contrast, the summary by *dTM-Dirichlet* covers more background group-specific words, like "summarization" and "MDS". Although these background group-specific terms are discriminative for the group, they are relatively less informative than the salient terms for the purpose of summarization.

5.6 Comparative Summarization in Scientific Resource Space

In this section, we use the dataset in Section 5.5.1 to construct a scientific resource space. The micro-dimensions in the scientific resource space is shown in Figure 5.3, where new research problems such as "Summarization", "Sentiment Analysis" and "Geo-related tasks" are inserted as coordinates on Task dimension, the method related coordinates such as "Topic Model" are inserted on Process dimension and the data set related coordinates such as "DUC" and "TAC" are inserted on Material dimension.

The coordinate partition is performed by first choosing the Topic Model as the original coordinate and then using Task dimension as the partition axis to partition resources under the original coordinate. The categories produced by the coordinate partition operation correspond to the three research problems "Summarization", "Sentiment Analysis" and "Geo-related tasks" that addressed by a same technique route "Topic Model". The comparative summary generated for each category by *dTM-SAGE* method is shown in Figure 5.4 - 5.6.

Figure 5.5 shows the comparative summary generated for the category of "Sentiment Analysis". This summary includes contents unique to sentiment analysis and how topic models are applied to sentiment analysis such as Joint Sentiment Topic model (JST), Aspect and Sentiment Unification model ($ASUM$) and Topic Sentiment Mixture (TSM). Meanwhile it excludes contents of background knowledge on topic models. Beside, this summary contains multi-aspect

Table 5.7: 5-sentence summary generated by dTM-Dirichlet and dTM-SAGE.

Summary by dTM-Dirichlet. 1. Most of the existing multi-document summarization methods decompose the documents into sentences and work directly in the sentence space using a term-sentence matrix. 2. Bayesian sentences-based topic model, every sentences in a document is assumed to be associated to a unique latent topic. 3. While previous MDS systems have focused primarily on salience and coverage but not coherence, G-Flow generates an ordered summary by jointly optimizing coherence and salience. 4. Markov Random Walk Model (MRW) Graphs methods have been successfully applied to weighting sentences for generic and query-focused summarization. 5. The topic distributions are used to get the sentence scores and rank sentences.
Summary by dTM-SAGE. 1. In recent years, three major techniques have emerged to perform multi-document summarization: graph-based methds such as LexRank, Biased-LexRank for query-focused summarization, language models such as KLSum and variants based on topic models, such as BayeSum and TopicSum. 2. Bayesian Query-Focused Summarization, we present BayeSum (Bayesian summarization), a model for sentence extraction in query-focused summarization. 3. Sentence Selection Strategy, The task of timeline summarization aims to produce a summary for each time and the generated summary should meet criteria such as relevance, coverage and coherence. 4. Models that use more structure in the representation of documents have also been proposed for generating more coherent and less redundant summaries, such as HierSUM and TTM. 5. In generating a summary, we use MMR (Maximal Marginal Relevance for multi-document) to avoid redundancy in a summary.

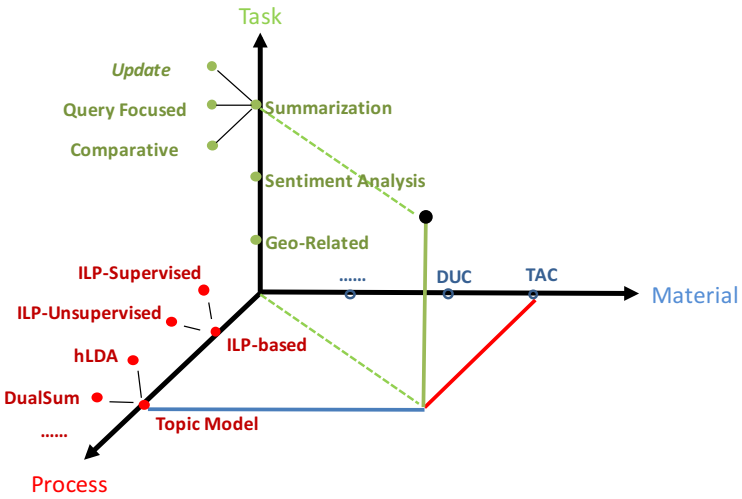


Figure 5.3: The micro-dimensions of scientific resource space generated on comparative summarization dataset.

Text Summarization

5-Sentence Summary generated by *dTM* Comparative Summarization Method:

- ① In recent years, three major techniques have emerged to perform multi-document summarization: graph-based methds such as LexRank, Biased-LexRank for query-focused summarization, language models such as KLSum and variants based on topic models, such as BayeSum and TopicSum.
- ② Bayesian Query-Focused Summarization, we present BayeSum (Bayesian summarization), a model for sentence extraction in query-focused summarization.
- ③ Sentence Selection Strategy, The task of timeline summarization aims to produce a summary for each time and the generated summary should meet criteria such as relevance, coverage and coherence.
- ④ Models that use more structure in the representation of documents have also been proposed for generating more coherent and less redundant summaries, such as HierSUM and TTM.
- ⑤ In generating a summary, we use MMR (Maximal Marginal Relevance for multi-document) to avoid redundancy in a summary.

Close me!

Figure 5.4: The comparative summary of Summarization category.

Sentiment Analysis

5-Sentence Summary generated by *dTM* Comparative Summarization Method:

- ① Sentiment analysis or opinion mining involves examine opinions about the products and aspects (price, food) in reviews or tweets.
- ② Sentiment Analysis is on three levels document level, phrase level, and aspect level Sentiment Analysis.
- ③ Unsupervised learning and supervised classification (NB, SVM) methods are used to determine the sentiment polarity of each aspect category.
- ④ Typical Topic models for sentiment analysis are Joint sentiment topic model (JST), Aspect and Sentiment Unification model (ASUM), Topic Sentiment Mixture (TSM) for unsupervised joint sentiment topic detection.
- ⑤ Data sets are movie reviews, in camera domain and multi-domain sentiment data set from IMDB and amazon.com.

Close me!

Figure 5.5: The comparative summary of Sentiment Analysis category.

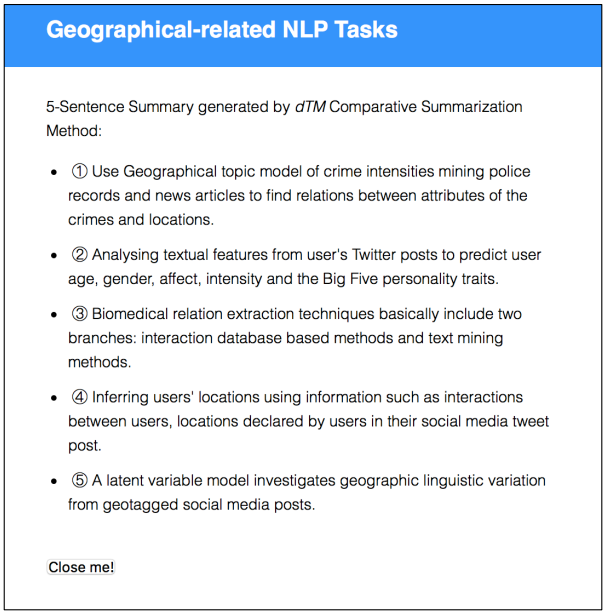


Figure 5.6: The comparative summary of Geographical-Related category.

information of sentiment analysis, including problem description, sub-problem division, typical methods and data sets, which fully meets the demands of the application scenario described in Section 5.1.

6 Conclusions

6.1 Thesis Summary

The number of scientific literature resources has been increasing exponentially, which sharply contradicts people's limited reading time. In the face of the explosive growth of massive scientific literature resources, the lack of effective models organizing resources reduces the efficiency in the acquisition and utilization of scientific literature. How to efficiently organize and manage the vast amount of scientific literature has become an important problem in the field of computer science.

This thesis exploits the resource space model (RSM) to organize scientific literature resources. The multi-dimensional hierarchical coordinate structure of a resource space naturally supports multi-facet resource browsing and hierarchical query, which should hopefully satisfy users' demands on scientific information acquisition. We combine the characteristics of scientific resources with RSM and propose scientific resource space. A scientific resource space consists of two types of dimensions: macro-dimensions and micro-dimensions. Macro-dimensions describe the metadata of scientific articles, while micro-dimensions semantically describe the fine-grained contents of scientific articles.

Automatic construction of a scientific resource space is the main research problem of this thesis. We propose the construction methods for macro-dimensions and micro-dimensions respectively. We study the comparative summarization for generating summaries based on coordinate partition in a scientific resource space. In addition, we design a prototype system based on scientific resource space which can help researchers query and browse scientific resources. The main contribution of this thesis automates the construction process of a scientific resource space and provides a series of services to help users get useful information accurately and efficiently.

Firstly, automatic maintenance of the category hierarchy is proposed for the construction of macro-dimensions. The category hierarchy in the macro-dimension needs to evolve dynamically so as to satisfy the dynamic requirements of the organization and management of resources. We propose an automatic maintenance method to modify the original category hierarchy according to the hierarchical clustering of resources. Experimental results on Reuters-21578, 20Newsgroups, DMOZ datasets show the effectiveness of this method.

Next, in terms of the construction of micro-dimensions, we propose a joint entity and relation extraction model based on deep neural network to extract three types of entities (Task, Process

and Material) and two types of relations (Hyponymy and Synonymy) from scientific articles to build concept hierarchy in the micro-dimensions. The entity and relation tasks share basic representation layers in the unified neural network framework to prompt performance of each other. Experimental results on SemEval 2017 Task 10 dataset show the effectiveness of the joint model on entity recognition and relation extraction tasks.

Last, based on the above automatic construction methods, we implement a prototype system of the scientific resource space, which provides a series of services on scientific articles. We focus on the scientific summarization service and propose a new comparative summarization method based on differential topic models. It solves the problem of generating comparative summaries based on the coordinate partition in a scientific resource space. The comparative summary points out differences between different categories. For example, it can facilitate the comparison between different methods applied to a same problem or between problems addressed by a same technique route.

6.2 Future Work

This thesis takes an initial research on the organization and management of large scientific resources based on resource space model. It involves many research areas, such as natural language processing, machine learning and databases. This thesis focuses on the automatic construction of a scientific resource space and solves several problems during the construction process. Followings are some key research points for the future plan:

1. Design complete resource operations in scientific resource space. The main objects in a scientific resource space are scientific literature resources. The resource space model needs to provide a series of complete operations to enable resource query, modify and update. Query capability and expressive power of a scientific resource space lay on the foundation of the completeness of resource operations.

2. Study physical storage mechanism for scientific resource space. The multi-dimensional hierarchical characteristics of the scientific resource space require special storage mechanism to guarantee the efficiency of resource retrieval. Traditional spatial indexing methods rely on a linear order of coordinates on each dimension, so that Euclidean distance can be used to measure the similarity of resources. Similar resources are stored in a near area to ensure efficient retrieval. However, coordinates in a scientific resource space represent classification semantics and usually have hierarchical relationships other than linear order.

3. Combine with probabilistic resource space model. The probabilistic resource space model allows users to organize resources with uncertainty. In this thesis, scientific articles are classified into definite categories, however in many cases, it is hard to determine whether an article falls into a certain category or not, especially for interdisciplinary ones. Thus, it is necessary to incorporate probability with the resource space model to operate resources with uncertainty.

References

Aggarwal, C. C., Gates, S. C., & Yu, P. S. (1999). On the merits of building categorization systems by supervised clustering. In *Proceedings of the fifth acm sigkdd international conference on knowledge discovery and data mining* (pp. 352–356).

Ahmed, A., & Xing, E. P. (2010). Staying informed: supervised and semi-supervised multi-view topical analysis of ideological perspective. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 1140–1150).

Aletras, N., & Stevenson, M. (2013). Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th international conference on computational semantics (iwcs 2013)–long papers* (pp. 13–22).

Anh, T. L., Tay, Y., Hui, S. C., & Ng, S. K. (2016). Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 403–413).

Augenstein, I., Das, M., Riedel, S., Vikraman, L., & McCallum, A. (2017). Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*.

Banerjee, S., Mitra, P., & Sugiyama, K. (2015). Multi-document abstractive summarization using ilp based multi-sentence compression. In *Ijcai* (pp. 1208–1214).

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137–1155.

Bing, L., Li, P., Liao, Y., Lam, W., Guo, W., & Passonneau, R. J. (2015). Abstractive multi-document summarization via phrase selection and merging. *arXiv preprint arXiv:1506.01597*.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Boudin, F., El-Bèze, M., & Torres-Moreno, J.-M. (2008). A scalable mmr approach to sentence scoring for multi-document update summarization. *Coling 2008: Companion volume: Posters*, 23–26.

Bunescu, R. C., & Mooney, R. J. (2005). A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 724–731).

Cai, X., Li, W., Ouyang, Y., & Yan, H. (2010). Simultaneous ranking and clustering of sentences: a reinforcement approach to multi-document summarization. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 134–142).

Carbonell, J., & Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international acm sigir conference on research and development in information retrieval* (pp. 335–336).

Cetoli, A., Bragaglia, S., O'Harney, A., & Sloan, M. (2017). Graph convolutional networks for named entity recognition. *arXiv preprint arXiv:1709.10053*.

Chang, C.-C., & Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, *2*(3), 27.

Chen, J., Ji, D., Tan, C. L., & Niu, Z. (2005). Unsupervised feature selection for relation extraction. In *Companion volume to the proceedings of conference including posters/demos and tutorial abstracts.*

Chiu, J. P., & Nichols, E. (2015). Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*.

Chuang, S.-L., & Chien, L.-F. (2004). A practical web-based approach to generating topic hierarchy for text segments. In *Proceedings of the thirteenth acm international conference on information and knowledge management* (pp. 127–136).

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493–2537.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Delort, J.-Y., & Alfonseca, E. (2012). Dualsum: a topic-model based approach for update summarization. In *Proceedings of the 13th conference of the european chapter of the association for computational linguistics* (pp. 214–223).

de Waard, A., Buitelaar, P., & Eigner, T. (2009). Identifying the epistemic value of discourse segments in biology texts. In *Proceedings of the eighth international conference on computational semantics* (pp. 351–354).

Dhillon, I. S., & Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine learning*, *42*(1-2), 143–175.

Eisenstein, J., Ahmed, A., & Xing, E. P. (2011). Sparse additive generative models of text.

Erkan, G., & Radev, D. R. (2004). Lexpagerank: Prestige in multi-document text summarization. In *Proceedings of the 2004 conference on empirical methods in natural language processing.*

Filippova, K. (2010). Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 322–330).

Filippova, K., & Strube, M. (2008). Sentence fusion via dependency graph compression. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 177–185).

Fisher, S., & Roark, B. (2008). Query-focused supervised sentence ranking for update summaries. In *Tac.*

Fu, R., Guo, J., Qin, B., Che, W., Wang, H., & Liu, T. (2014). Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (Vol. 1, pp. 1199–1209).

Geffet, M., & Dagan, I. (2005). The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd annual meeting on association for computational linguistics* (pp. 107–114).

Genest, P.-E., & Lapalme, G. (2012). Fully abstractive approach to guided summarization. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2* (pp. 354–358).

Gillick, D., Brunk, C., Vinyals, O., & Subramanya, A. (2015). Multilingual language processing from bytes. *arXiv preprint arXiv:1512.00103*.

Graves, A., Jaitly, N., & Mohamed, A.-r. (2013). Hybrid speech recognition with deep bidirectional lstm. In *Automatic speech recognition and understanding (asru), 2013 ieee workshop on* (pp. 273–278).

Green, M. J., Behabtu, N., Pasquali, M., & Adams, W. W. (2009). Nanotubes as polymers. *Polymer*, *50*(21), 4979–4997.

Griffiths, T. L., Jordan, M. I., Tenenbaum, J. B., & Blei, D. M. (2004). Hierarchical topic models and the nested chinese restaurant process. In *Advances in neural information processing systems* (pp. 17–24).

Gu, J., Lu, Z., Li, H., & Li, V. O. (2016). Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Guha, S., Rastogi, R., & Shim, K. (1998). Cure: an efficient clustering algorithm for large databases. In *Acm sigmod record* (Vol. 27, pp. 73–84).

GuoDong, Z., Jian, S., Jie, Z., & Min, Z. (2005). Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics* (pp. 427–434).

Hasegawa, T., Sekine, S., & Grishman, R. (2004). Discovering relations among named entities from large corpora. In *Proceedings of the 42nd annual meeting on association for computational linguistics* (p. 415).

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on computational linguistics-volume 2* (pp. 539–545).

Hirohata, K., Okazaki, N., Ananiadou, S., & Ishizuka, M. (2008). Identifying sections in scientific

abstracts using conditional random fields. In *Proceedings of the third international joint conference on natural language processing: Volume-i.*

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107–116.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Jain, A. K., & Dubes, R. C. (1988). Algorithms for clustering data.

Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the acl 2004 on interactive poster and demonstration sessions* (p. 22).

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, *32*(8), 68–75.

Khashabi, D. (2013). On the recursive neural networks for relation extraction and entity recognition.

King, B. (1967). Step-wise clustering procedures. *Journal of the American Statistical Association*, *62*(317), 86–101.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, *46*(5), 604–632.

Kotlerman, L., Dagan, I., Szpektor, I., & Zhitomirsky-Geffet, M. (2010). Directional distributional similarity for lexical inference. *Natural Language Engineering*, *16*(4), 359–389.

Kozareva, Z., & Hovy, E. (2010). A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 1110–1118).

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lenci, A., & Benotto, G. (2012). Identifying hypernyms in distributional semantic spaces. In *Proceedings of the first joint conference on lexical and computational semantics-volume 1: Proceedings of the main conference and the shared task, and volume 2: Proceedings of the sixth international workshop on semantic evaluation* (pp. 75–79).

Levy, O., Remus, S., Biemann, C., & Dagan, I. (2015). Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 970–976).

Li, C., Qian, X., & Liu, Y. (2013). Using supervised bigram-based ilp for extractive summarization. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)* (Vol. 1, pp. 1004–1013).

Li, P., Wang, Y., Gao, W., & Jiang, J. (2011). Generating aspect-oriented multi-document summarization with event-aspect model. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 1137–1146).

Li, Q., & Ji, H. (2014). Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (Vol. 1, pp. 402–412).

Li, T., Zhu, S., & Ogihara, M. (2007). Hierarchical document classification using automatically generated hierarchy. *Journal of Intelligent Information Systems*, *29*(2), 211–230.

Li, W. (2015). Abstractive multi-document summarization with semantic information extraction. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1908–1913).

Li, W., Wei, F., Lu, Q., & He, Y. (2008). Pnr 2: ranking sentences with positive and negative reinforcement for query-oriented update summarization. In *Proceedings of the 22nd international conference on computational linguistics-volume 1* (pp. 489–496).

Liakata, M., Saha, S., Dobnik, S., Batchelor, C., & Rebholz-Schuhmann, D. (2012). Automatic recognition of conceptualization zones in scientific articles and two life science applications. *Bioinformatics*, *28*(7), 991–1000.

Liakata, M., Teufel, S., Siddharthan, A., & Batchelor, C. (2010). Corpora for the conceptualisation and zoning of scientific papers. In *Lrec*.

Lin, C.-Y., & Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology-volume 1* (pp. 71–78).

Lin, J., Karakos, D., Demner-Fushman, D., & Khudanpur, S. (2006). Generative content models for structural analysis of medical abstracts. In *Proceedings of the workshop on linking natural language processing and biology: Towards deeper biological literature analysis* (pp. 65–72).

Liu, T.-Y., Yang, Y., Wan, H., Zeng, H.-J., Chen, Z., & Ma, W.-Y. (2005). Support vector machines classification with a very large-scale taxonomy. *Acm Sigkdd Explorations Newsletter*, *7*(1), 36–43.

Liu, Y., Wei, F., Li, S., Ji, H., Zhou, M., & Wang, H. (2015). A dependency-based neural network for relation classification. *arXiv preprint arXiv:1507.04646*.

Mann, W. C., & Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, *8*(3), 243–281.

Marcu, D. (2000). The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational linguistics*, *26*(3), 395–448.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing.*

Mikolov, T., Yih, W.-t., & Zweig, G. (2013). 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* (pp. 746–751).

Minka, T. (2000). *Estimating a dirichlet distribution.* Technical report, MIT.

Miwa, M., & Bansal, M. (2016). End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Nallapati, R., Zhou, B., Santos, C. d., Gulcehre, C., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Navigli, R., Velardi, P., & Faralli, S. (2011). A graph-based algorithm for inducing lexical taxonomies from scratch. In *Ijcai* (Vol. 11, pp. 1872–1877).

Nawaz, R., Thompson, P., McNaught, J., & Ananiadou, S. (2010). Meta-knowledge annotation of bio-events. In *Lrec* (pp. 2498–2507).

Nenkova, A. (2008). Entity-driven rewrite for multi-document summarization.

Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics* (pp. 100–108).

Nicholas, D., & Clark, D. (2012). 'reading'in the digital environment. *Learned Publishing*, *25*(2), 93–98.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The pagerank citation ranking: Bringing order to the web.* (Tech. Rep.). Stanford InfoLab.

Paisley, J., Wang, C., Blei, D. M., & Jordan, M. I. (2015). Nested hierarchical dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(2), 256–270.

Paul, M. (2009). Cross-collection topic models: Automatically comparing and contrasting text. *Urbana*, *51*, 61801.

Paul, M., & Girju, R. (2009). Cross-cultural analysis of blogs and forums with mixed-collection topic models. In *Proceedings of the 2009 conference on empirical methods in natural language processing: Volume 3-volume 3* (pp. 1408–1417).

Paul, M., & Girju, R. (2010). A two-dimensional topic-aspect model for discovering multi-faceted topics. *Urbana*, *51*(61801), 36.

Phan, X.-H., & Nguyen, C.-T. (2007). Gibbslda++: Ac/c++ implementation of latent dirichlet allocation (lda). *Tech. rep.*.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., & Welling, M. (2008). Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th acm sigkdd international conference on knowledge discovery and data mining* (pp. 569–577).

Puzicha, J., Hofmann, T., & Buhmann, J. M. (2000). A theory of proximity based clustering: Structure detection by optimization. *Pattern Recognition*, *33*(4), 617–634.

Qian, L., Zhou, G., Kong, F., Zhu, Q., & Qian, P. (2008). Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd international conference on computational linguistics-volume 1* (pp. 697–704).

Radev, D. R., Jing, H., & Budzikowska, M. (2000). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 naacl-anlpworkshop on automatic summarization-volume 4* (pp. 21–30).

Radev, D. R., Jing, H., Styś, M., & Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management*, *40*(6), 919–938.

Ravenscroft, J., Oellrich, A., Saha, S., & Liakata, M. (2016). Multi-label annotation in scientific articles-the multi-label cancer risk assessment corpus. In *Lrec.*

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the eighth acm international conference on web search and data mining* (pp. 399–408).

Roth, D., & Yih, W.-t. (2004). *A linear programming formulation for global inference in natural language tasks* (Tech. Rep.). ILLINOIS UNIV AT URBANA-CHAMPAIGN DEPT OF COMPUTER SCIENCE.

Roth, D., & Yih, W.-t. (2007). Global inference for entity and relation identification via a linear programming formulation. *Introduction to statistical relational learning*, 553–580.

Ruch, P., Boyer, C., Chichester, C., Tbahriti, I., Geissbühler, A., Fabry, P., ... Veuthey, A.-L. (2007). Using argumentation to extract key sentences from biomedical abstracts. *International journal of medical informatics*, *76*(2-3), 195–200.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513–523.

See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, *27*(3), 379–423.

Shen, C., & Li, T. (2010). Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 984–992).

Siddharthan, A. (2011). Text simplification using typed dependencies: a comparison of the robustness of different generation strategies. In *Proceedings of the 13th european workshop on natural language generation* (pp. 2–11).

Singh, S., Riedel, S., Martin, B., Zheng, J., & McCallum, A. (2013). Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 workshop on automated knowledge*

base construction (pp. 1–6).

Sneath, P. H., & Sokal, R. R. (1973). *Numerical taxonomy. the principles and practice of numerical classification.*

Snow, R., Jurafsky, D., & Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In *Advances in neural information processing systems* (pp. 1297–1304).

Socher, R., Huval, B., Manning, C. D., & Ng, A. Y. (2012). Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 1201–1211).

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. In *Kdd workshop on text mining* (Vol. 400, pp. 525–526).

Sun, A., & Lim, E.-P. (2001). Hierarchical text classification and evaluation. In *Data mining, 2001. icdm 2001, proceedings ieee international conference on* (pp. 521–528).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Tan, L., Gupta, R., & van Genabith, J. (2015). Usaar-wlv: Hypernym generation with deep neural nets. In *Proceedings of the 9th international workshop on semantic evaluation (semeval 2015)* (pp. 932–937).

Tang, L., Zhang, J., & Liu, H. (2006). Acclimatizing taxonomic semantics for hierarchical content classification. In *Proceedings of the 12th acm sigkdd international conference on knowledge discovery and data mining* (pp. 384–393).

Tenopir, C., Mays, R., & Wu, L. (2011). Journal article growth and reading patterns. *New Review of Information Networking*, *16*(1), 4–22.

Teufel, S., Carletta, J., & Moens, M. (1999). An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of the ninth conference on european chapter of the association for computational linguistics* (pp. 110–117).

Teufel, S., & Kan, M.-Y. (2011). Robust argumentative zoning for sensemaking in scholarly documents. In *Advanced language technologies for digital libraries* (pp. 154–170). Springer.

Teufel, S., & Moens, M. (2002). Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, *28*(4), 409–445.

Teufel, S., Siddharthan, A., & Batchelor, C. (2009). Towards discipline-independent argumentative zoning: evidence from chemistry and computational linguistics. In *Proceedings of the 2009 conference on empirical methods in natural language processing: Volume 3-volume 3* (pp. 1493–1502).

Thompson, P., Nawaz, R., McNaught, J., & Ananiadou, S. (2011). Enriching a biomedical event

corpus with meta-knowledge annotation. *BMC bioinformatics*, *12*(1), 393.

Tratz, S., & Hovy, E. (2010). A taxonomy, dataset, and classifier for automatic noun compound interpretation. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 678–687).

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, *37*, 141–188.

Wan, X., & Xiao, J. (2009). Graph-based multi-modality learning for topic-focused multi-document summarization. In *Ijcai* (pp. 1586–1591).

Wan, X., Yang, J., & Xiao, J. (2007). Manifold-ranking based topic-focused multi-document summarization. In *Ijcai* (Vol. 7, pp. 2903–2908).

Wang, D., Zhu, S., Li, T., & Gong, Y. (2012). Comparative document summarization via discriminative sentence selection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *6*(3), 12.

Wang, M. (2008). A re-examination of dependency path kernels for relation extraction. In *Proceedings of the third international joint conference on natural language processing: Volume-ii.*

Ware, M., & Mabe, M. (2015). The stm report: An overview of scientific and scholarly journal publishing.

Weeds, J., Weir, D., & McCarthy, D. (2004). Characterising measures of lexical distributional similarity. In *Proceedings of the 20th international conference on computational linguistics* (p. 1015).

Xu, K., Feng, Y., Huang, S., & Zhao, D. (2015). Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv preprint arXiv:1506.07650*.

Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., & Jin, Z. (2016). Improved relation classification by deep recurrent neural networks with data augmentation. *arXiv preprint arXiv:1601.03651*.

Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., & Jin, Z. (2015). Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1785–1794).

Yang, B., & Cardie, C. (2013). Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)* (Vol. 1, pp. 1640–1649).

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international acm sigir conference on research and development in information retrieval* (pp. 42–49).

Yu, X., & Lam, W. (2010). Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Proceedings of the 23rd international conference*

on computational linguistics: Posters (pp. 1399–1407).

Yu, Z., Wang, H., Lin, X., & Wang, M. (2015). Learning term embeddings for hypernymy identification. In *Ijcai* (pp. 1390–1397).

Yuan, Q., Cong, G., Sun, A., Lin, C.-Y., & Thalmann, N. M. (2012). Category hierarchy maintenance: a data-driven approach. In *Proceedings of the 35th international acm sigir conference on research and development in information retrieval* (pp. 791–800).

Zaremba, W., & Sutskever, I. (2014). Learning to execute. *arXiv preprint arXiv:1410.4615*.

Zelenko, D., Aone, C., & Richardella, A. (2003). Kernel methods for relation extraction. *Journal of machine learning research*, *3*(Feb), 1083–1106.

Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of coling 2014, the 25th international conference on computational linguistics: Technical papers* (pp. 2335–2344).

Zhai, C., Velivelli, A., & Yu, B. (2004). A cross-collection mixture model for comparative text mining. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining* (pp. 743–748).

Zhang, S., Zheng, D., Hu, X., & Yang, M. (2015). Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th pacific asia conference on language, information and computation* (pp. 73–78).

Zhao, Y., Karypis, G., & Fayyad, U. (2005). Hierarchical clustering algorithms for document datasets. *Data mining and knowledge discovery*, *10*(2), 141–168.

Zhuge, H. (2004). Resource space model, its design method and applications. *Journal of Systems and Software*, *72*(1), 71–81.

Zhuge, H. (2007). *The web resource space model* (Vol. 4). Springer Science & Business Media.

Zhuge, H. (2016). *Multi-dimensional summarization in cyber-physical society*. Morgan Kaufmann.

Zhuge, H., & Xing, Y. (2012). Probabilistic resource space model for managing resources in cyber-physical society. *IEEE Transactions on Services Computing*, *5*(3), 404–421.

Appendices

A Inference on Variables in *dTM-SAGE*

Generally, we take MAP (maximum a posterior) estimation for the background word distribution $\boldsymbol{\vartheta_0}$ and the group component vectors $\boldsymbol{\eta}$ and develop variational inference techniques for all other variables.

In *dTM-SAGE*, the lower bound L with regarding to $\boldsymbol{\vartheta_0}$, $\boldsymbol{\eta}$ and $\boldsymbol{\sigma}$ is:

$$\begin{aligned} L = logP(\boldsymbol{\vartheta_0}|\alpha_\vartheta) + \sum_g \sum_m \sum_n E_Q[logP(w_{g,m,n}|s_{g,m,n}=0, \boldsymbol{\vartheta_0}, \boldsymbol{\eta_g})] \\ + \sum_g E_Q[logP(\boldsymbol{\eta_g}|0, \boldsymbol{\sigma_g}] + \sum_g E_Q[logP(\boldsymbol{\sigma_g}|\alpha_\sigma] - \sum_g E_Q[logQ(\boldsymbol{\sigma_g})] \end{aligned} \tag{A.1}$$

Maximize L with respect to $\boldsymbol{\vartheta_0}$:

$$L(\boldsymbol{\vartheta_0}) = \sum_v (\alpha_\vartheta^v - 1) * log\vartheta_0^v + \sum_g \sum_m \sum_n \widetilde{\lambda}_{gmn}^0 * \{\vartheta_0^{w_{g,m,n}} - log(\sum_j^V exp(\eta_g^j + \vartheta_0^j))\} \tag{A.2}$$

By Assuming $T(v) = \frac{exp(\eta_g^v + \vartheta_0^v)}{\sum_j^V exp(\eta_g^j + \vartheta_0^j)}$, taking derivatives with respect to ϑ_0^v:

$$\frac{\partial L}{\partial \vartheta_0^v} = \frac{\alpha_\vartheta^v - 1}{\vartheta_0^v} + \sum_g \sum_m \sum_n \widetilde{\lambda}_{gmn}^0 * \{I(w_{g,m,n} = v) * (1 - T(v)) + I(w_{g,m,n} \neq v) * (-T(v))\} \tag{A.3}$$

We use Newton-Raphson method to optimize $\boldsymbol{\vartheta_0}$. First, we derive the Hessian matrix by setting:

$$\begin{aligned} H_{vv}(\boldsymbol{\vartheta_0}) = & \frac{d^2 L}{(d\vartheta_0^v)^2} = -\frac{\alpha_\vartheta^v - 1}{(\vartheta_0^v)^2} + \sum_g \sum_m \sum_n \widetilde{\lambda}_{gmn}^0 * (T(v)^2 - T(v)) \\ H_{vv'}(\boldsymbol{\vartheta_0}) = & \frac{d^2 L}{d\vartheta_0^v d\vartheta_0^{v'}} = \sum_g \sum_m \sum_n \widetilde{\lambda}_{gmn}^0 * T(v)T(v') \end{aligned} \tag{A.4}$$

After getting Hessian matrix, we invert it with Sherman-Morrison formula and compute the Newton step:$\triangle \boldsymbol{\vartheta_0} \triangle \boldsymbol{\vartheta_0} = H^{-1}(\boldsymbol{\vartheta_0}) \bigtriangledown_{\boldsymbol{\vartheta_0}} L(\boldsymbol{\vartheta_0})$.

Same procedure on $\boldsymbol{\eta}$:

$$\begin{aligned} \frac{\partial L}{\partial \eta_g^v} = -\eta_g^v * [(\widetilde{\alpha}_g^v - 1)\widetilde{b}_g^v]^{-1} + \sum_g \sum_m \sum_n \widetilde{\lambda}_{gmn}^0 * \{I(w_{g,m,n} = v) * (1 - T(v)) \\ + I(w_{g,m,n} \neq v) * (-T(v))\} \end{aligned} \tag{A.5}$$

$$\begin{aligned} H_{vv}(\boldsymbol{\eta_g}) = & \frac{d^2 L}{(d\eta_g^v)^2} = -[(\widetilde{\alpha}_g^v - 1)\widetilde{b}_g^v]^{-1} + \sum_g \sum_m \sum_n \widetilde{\lambda}_{gmn}^0 * (T(v)^2 - T(v)) \\ H_{vv'}(\boldsymbol{\eta_g}) = & \frac{d^2 L}{d\eta_g^v d\eta_g^{v'}} = \sum_g \sum_m \sum_n \widetilde{\lambda}_{gmn}^0 * T(v)T(v') \end{aligned} \tag{A.6}$$