Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances

Wilson Wong
The University of Western Australia, Australia

Wei Liu
The University of Western Australia, Australia

Mohammed Bennamoun
The University of Western Australia, Australia

Senior Editorial Director:	Kristin Klinger
Director of Book Publications:	Julia Mosemann
Editorial Director:	Lindsay Johnston
Acquisitions Editor:	Erika Carter
Development Editor:	Joel Gamon
Production Editor:	Sean Woznicki
Typesetters:	Jennifer Romanchak and Mike Brehm
Print Coordinator:	Jamie Snively
Cover Design:	Nick Newcomer

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: <http://www.igi-global.com/reference>

Copyright © 2011 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher. Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Ontology learning and knowledge discovery using the Web: challenges and recent advances / Wilson Wong, Wei Liu and Mohammed Bennamoun, editors.

p. cm.

Includes bibliographical references and index.

ISBN 978-1-60960-625-1 (hardcover) -- ISBN 978-1-60960-626-8 (ebook) 1. Ontologies (Information retrieval) 2. Data mining. I. Wong, Wilson, 1981- II. Liu, Wei, 1972 July 14- III. Bennamoun, M. (Mohammed)
TK5105.88815.O587 2011
006.3'12--dc22

2010043008

British Cataloguing in Publication Data

A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 7

Learning SKOS Relations for Terminological Ontologies from Text

Wei Wang

University of Nottingham Malaysia Campus, Malaysia

Payam M. Barnaghi

University of Surrey, UK

Andrzej Bargiela

University of Nottingham Jubilee Campus, UK

ABSTRACT

The problem of learning concept hierarchies and terminological ontologies can be divided into two sub-tasks: concept extraction and relation learning. The authors of this chapter describe a novel approach to learn relations automatically from unstructured text corpus based on probabilistic topic models. The authors provide definition (Information Theory Principle for Concept Relationship) and quantitative measure for establishing “broader” (or “narrower”) and “related” relations between concepts. They present a relation learning algorithm to automatically interconnect concepts into concept hierarchies and terminological ontologies with the probabilistic topic models learned. In this experiment, around 7,000 ontology statements expressed in terms of “broader” and “related” relations are generated using different combination of model parameters. The ontology statements are evaluated by domain experts and the results show that the highest precision of the learned ontologies is around 86.6% and structures of learned ontologies remain stable when values of the parameters are changed in the ontology learning algorithm.

INTRODUCTION

Concept hierarchies used in most of the contemporary digital libraries are created and maintained

manually. One of the most subtle problems with the manual approach is that the engineering and subsequent maintenance processes are time-consuming. The manually created hierarchies are also prone to suffer from obsolescence as research

DOI: 10.4018/978-1-60960-625-1.ch007

advances (e.g., invention of new techniques). Consequently, emergence of new terminologies in various research areas can not be easily reflected in such static concept hierarchies. It is also difficult to construct concept hierarchies with broader and deeper coverage, which can help users on query suggestion and expansion, browsing, and navigation. Unsupervised approaches for learning knowledge are perceived as promising approaches to alleviate these problems, minimising human involvement and effort.

The Semantic Web (Berners-Lee, Hendler, & Lassila, 2001) has been bringing more research attention towards knowledge acquisition using automated approaches. There have been a number of existing works that aim to learn different types of ontologies from unstructured text corpus using techniques from Natural Language Processing (Hearst, 1992, Cimiano & Staab, 2004, Cimiano, Pivk, Schmidt-Thieme, & Staab, 2005), Information Extraction (Cunningham, 2005, Cimiano & Völker, 2005, Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004), and Machine Learning (clustering and classification) (Maedche, Pekar, & Staab, 2002, Biemann, 2005). An important and plausible assumption is that given sufficient amount of text in a domain, coverage of knowledge in that domain can be ensured (Cimiano, 2006). Although learned ontologies are less accurate than those created manually, the advantages of being inexpensive, time-saving, and resistant to obsolescence make the automated approaches attractive, especially in domains where semi-structured data is not available or cannot be directly transformed to structured form.

A concept hierarchy or topic hierarchy can be viewed as a simple form of a terminological ontology in which concepts are not only organised using more general/specific relations, but also other types of relations, such as “related” (introduction on the “related” and “broader” relations can be found in Section “SKOS Ontology Model” and “Information Theory Principle for Concept Relationship”) defined in the SKOS ontology model¹.

In this chapter we explore a novel approach for learning terminological ontologies with respect to the SKOS model using Latent Dirichlet Allocation (Blei, Ng, & Jordan, 2003, Steyvers & Griffiths, 2005). The main objective is to establish “broader” and “related” relations between concepts using an unsupervised approach. The learned ontologies can be used to extend and expand existing topic hierarchies deployed in digital libraries and search engines, for different purposes such as facilitating search, browsing, query suggestion, and document annotation. The rest of the chapter is organised as follows. Section “Related Work” provides a short introduction on ontology categorisation and ontology learning tasks. It also gives an overview of existing methods for ontology learning from unstructured text. Section “Introduction to Latent Dirichlet Allocation” presents some background knowledge on the Latent Dirichlet Allocation, in particular the generative process, model representation and parameter estimation of the model. Section “Learning Relations in Terminological Ontologies” elaborates our approach for learning “broader” and “related” relations to construct terminological ontologies. We focus on the definition of concept relationship principle and an iterative algorithm for organising concepts into ontologies with the SKOS relations. Section “Experiment” describes the experiment conducted on a dataset which consists of abstracts of publications in the Semantic Web research area. Using the ontology learning algorithm with different parameters, around 7,000 ontology statements are generated. Results of evaluation in terms of recall, precision and F1 measures are demonstrated in Section “Evaluation”. Section “Discussion and Future Work” concludes the chapter and describes the future work.

RELATED WORK

In this section we first provide some background knowledge on ontology categorisation and on-

tology learning processes in order to define the specific problem that our approach aims to solve. We briefly review prevailing ontology learning approaches which use techniques from Natural Language Processing (Hearst, 1992, Cimiano & Staab, 2004, Cimiano et al., 2005), Information Extraction (Cunningham, 2005, Cimiano & Völker, 2005, Kiryakov et al., 2004), Machine Learning (Maedche et al., 2002), and simple Data Co-occurrence Analysis (Sanderson & Croft, 1999, Diederich & Balke, 2007) (A detailed survey on ontology learning methods can also be found in (Biemann, 2005)).

ONTOLOGY CATEGORISATION

The philosophical word “ontology” is borrowed by computer science or more recently by the Semantic Web to denote a specification of conceptualisation (Gruber, 2003). The word has become one of the most fundamental concepts in the Semantic Web research. The following provides a taxonomy for types of ontology² which receives consensus.

1. **Formal Ontology:** a terminological ontology whose categories are distinguished by axioms and definitions stated in logic or in some computer-oriented language that could be automatically translated to logic.
2. **Prototype-based Ontology:** a terminological ontology whose categories are distinguished by typical instances or prototypes rather than by axioms and definitions in logic.
3. **Terminological Ontology:** an ontology whose categories are not fully specified by axioms and definitions.

A formal ontology explicitly defines mechanisms (i.e., axioms) for logical inference, whereas a prototype-based ontology consists of categories which are formed by collecting instances extensionally. In a terminological ontology, concepts

can be organised in a tree-like structure using subsumption or “broader” (or “narrower”³) relations as well as other types of relations (e.g., “related” in the SKOS model). The ACM classification tree⁴ and Medical Subject Headings (MeSH)⁵ are two well-known examples of concept hierarchy. One limitation with tree-based representation is that (“fuzzy”) relations to more specific concepts cannot be modelled. The presence of the additional relations is one of the major differences between a terminological ontology and a concept hierarchy.

SKOS Ontology Model

SKOS (Simple Knowledge Organisation System) provides a model for expressing the basic structure and content of concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies, folksonomies, and other types of controlled vocabulary. The SKOS is an ideal model for representing relations between concepts or topics (often in a form of noun-phrases) in scientific research, for example, advance in research accelerates co-operations between different disciplines: to solve a problem in one research area (e.g., ontology learning in Semantic Web) one often utilises techniques from others areas, such as natural language processing and machine learning. Another advantage of knowledge representation using SKOS ontology compared to traditional concept hierarchy is that it is able to represent emerging concepts and relations. This chapter shows how to learn SKOS relations based on probabilistic topic models which are originally developed for document modelling.

Ontology Learning Tasks

Cimiano (Cimiano, 2006) identifies six subtasks for learning ontologies. The subtasks are organised in a “layered cake” which is composed of learning terms, synonyms, concepts, concept hierarchies, relations, and rules, from bottom to top. Learn-

ing of terminological ontology can be divided to learning two sub-tasks, concepts or terms⁶.

Ontology Learning from Text

Existing methods for learning ontologies of different types can be classified into four main categories as discussed in (Biemann, 2005): Lexico-syntactic based approach (Hearst, 1992, Cimiano & Staab, 2004, Cimiano et al., 2005), Information Extraction (Cimiano & Völker, 2005, Kiryakov et al., 2004), Clustering and Classification (Maedche et al., 2002), and Data Co-occurrence Analysis (Sanderson & Croft, 1999, Diederich & Balke, 2007).

Lexico-Syntactic Based Approach

Lexico-syntactic method uses natural language processing techniques to extract regular expressions that occur repeatedly in human languages, for example, the so-called Hearst-patterns (Hearst, 1992) are originally used to acquire hyponyms of some words from large text corpus. One of the limitations of this method is that the Hearst-patterns may not occur frequently in the text corpus which results in very low recall. Cimiano *et al* (Cimiano & Staab, 2004, Cimiano et al., 2005) attempt to explore the potential of large Web search engines to extract more occurrence of the language patterns, calling the method “Learning by Googling”. In ontology learning, the technique is often used to learn instances of pre-defined concepts.

Information Extraction

Information Extraction (IE) (Cunningham, 2005) techniques, in particular, Named Entity Recognition (NER) is used to automatically populate knowledge base (Cimiano & Völker, 2005, Kiryakov et al., 2004). The most significant limitation is that the current NER technology is only

effective for extracting instances of some general concepts such as “People”, “Location”, “Organisation” and their corresponding sub-concepts. This method, together with the Lexico-syntactic based approach have been primarily used for learning terms and concepts.

Clustering and Classification

Clustering is used to populate prototype-based ontology or to learn ontologies from scratch (corresponds to the learning concept hierarchies sub-task), whereas classification is used to augment a thesaurus (or extend concept hierarchies) with new lexical terms (Maedche et al., 2002). In methods based on these statistical learning techniques, terms are represented by vectors of terms in the vicinity. The representation is often based on an assumption called Harris’ distributional hypothesis, which states that similar words tend to occur in similar contexts (Harris, 1968). Similarity or dissimilarity functions are then used to determine to which clusters or classes a new term or object is assigned.

Data Co-Occurrence Analysis

Data Co-occurrence Analysis is a simple while effective approach for learning concept hierarchies by exploiting first-order or high-order co-occurrence⁷ of data. Sanderson *et al* (Sanderson & Croft, 1999) assume that “a term A subsumes B if the documents in which B occurs are (or nearly) a subset of the documents in which A occurs”. Another method uses a variation of the PageRank algorithm (Brin & Page, 1998) called GrowBag (Diederich & Balke, 2007) to analyse high-order data co-occurrence for constructing concept hierarchies. The learned concept hierarchy has been deployed in the FacetedDBLP⁸ browser to help users browsing and retrieving scientific publications.

INTRODUCTION TO LATENT DIRICHLET ALLOCATION

Latent Dirichlet Allocation (LDA) (Blei et al., 2003, Steyvers & Griffiths, 2005) is an instance of statistical topic models in which semantic properties of words and documents are represented with probabilistic (latent) topics. LDA, together with the probabilistic Latent Semantic Analysis (pLSA) (Hofmann, 1999a, 1999b), are originally developed as probabilistic extension of the Latent Semantic Analysis (LSA) (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990) for document modelling and topic extraction. Similar as its predecessors LSA and pLSA, LDA can also be used as an efficient dimension reduction technique. For example, in document clustering and classification, a document can be represented with a vector of topics in a low dimension space instead of vector of words in a high dimension to compute similarity with other documents. Before we elaborate the procedure of learning terminological ontologies using LDA, we provide some background knowledge of LDA in this section.

LATENT SEMANTIC ANALYSIS

The basic idea of LSA (Deerwester et al., 1990) is to reform document representation in high dimension word space to low dimension latent semantic space in order to capture implicit associations between words and documents. The low dimension latent space is obtained by decomposing a large term-document matrix using Singular Value Decomposition, a technique in linear algebra. An attractive characteristic of LSA is its capability to alleviate two classical Information Retrieval problems of synonymy (i.e., different words refer to the same meaning) and polysemy (i.e., one word has different meanings). One of the limitations of the technique is that although words and documents can be represented as points in the Euclidean space, the semantics of the model is

difficult to be explicitly interpreted (Hofmann, 1999a, 1999b).

PROBABILISTIC LATENT SEMANTIC ANALYSIS

Probabilistic Latent Semantic Analysis (pLSA) (Hofmann, 1999a, 1999b) is a probabilistic extension of the LSA model. It is a statistical generative model for analysing general co-occurrence data using probabilistic theories. pLSA specifies a simple generative process for a word in a document:

1. choose a document d with a prior probability $P(d)$,
2. choose a latent class z from the document with probability $P(z | d)$,
3. choose a word w from the latent class distribution with probability $P(w | z)$.

Parameters (i.e., $P(w | z)$, $P(d | z)$ and $P(d)$) in pLSA are estimated using the Expectation Maximisation (EM) algorithm (Hofmann, 1999a, 1999b).

LATENT DIRICHLET ALLOCATION

While pLSA is a significant step toward probabilistic modelling of text, it is incomplete in that it provides no probabilistic model at the level of documents, in other words, pLSA does not make any assumptions about how the mixture weights $p(z | d)$ are generated (Blei et al., 2003). Consequently, the number of parameters need to be estimated grows linearly with the size of the corpus, which leads to a serious problem of overfitting (Blei et al., 2003); and it is also difficult to test generalisability of the model to new documents (Steyvers & Griffiths, 2005).

In the original LDA model (Blei et al., 2003) a Dirichlet prior is introduced on document-topic

distribution. As a conjugate prior for multinomial distributions (D'Agostini, 2003), Dirichlet prior simplifies the problem of statistical inference. Steyvers and Griffiths (Steyvers & Griffiths, 2005, T. L. Griffiths & Steyvers, 2004, T. Griffiths & Steyvers, 2002) modify the original LDA model by introducing another symmetric Dirichlet prior on topic-word distribution. They also show how to perform parameter estimation using Gibbs sampling, one form of the Markov Chain Monte Carlo (Andrieu, Freitas, Doucet, & Jordan, 2003).

Generative Process

In LDA, each word w in a document d is generated by sampling a topic z from topic distribution, and then sampling a word from topic-word distribution. The probability of any particular word token is defined by Equation (1):

$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

where $P(z_i = j)$ is the probability that j th topic is sampled for the i th word token, $P(w_i | z_i = j)$ is the probability of sampling w_i under topic j , and T is the number of hidden topics. Let $\phi^{(j)} = P(w | z = j)$ refer to multinomial distribution over words for the topic j , and $\theta^{(d)} = P(z)$ refer to multinomial distribution over topics in the document d . The complete generative process can be expressed as:

$$w_i | z_i, \phi^{(z_i)} \sim \text{Discrete}(\phi^{(z_i)})$$

$$\phi \sim \text{Dirichlet}(\beta)$$

$$z_i | \theta(d_i) \sim \text{Discrete}(\theta(d_i))$$

$$\theta \sim \text{Dirichlet}(\alpha)$$

Parameter Estimation

The original LDA paper (Blei et al., 2003) uses variational inference with EM algorithm to estimate parameters. In this work, we adopt the parameter estimation method using Gibbs sampling because of its simplicity as discussed in (Steyvers & Griffiths, 2005, T. L. Griffiths & Steyvers, 2004). In parameter estimation using Gibbs sampling, instead of directly estimating the topic-word $p(w | z)$ and document-topic $p(z | d)$ distributions, one estimates the posterior probability distribution over latent variable z given the observed data conditioned on topic assignment for all the other word tokens using Equation (2) (See (T. Griffiths & Steyvers, 2002, T. L. Griffiths & Steyvers, 2004, Steyvers & Griffiths, 2005) for details).

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(0)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,j}^{(d_i)} + T\alpha}$$

The first term in the right side of Equation (2) indicates the probability of a word w under topic j , whereas the second term indicates the probability of topic j in the document d . The intuition of this equation is that once many tokens of a word have been assigned to a topic j (across documents), it will increase the probability of assigning any particular token of that word to topic j ; once a topic j has appeared many times in a document, it will increase the probability of any word tokens from that document will be assigned to topic j . Therefore, the assignment of a word to a topic depends on not only how likely the word is for a topic, but also how dominant a topic is in a document (Steyvers & Griffiths, 2005).

The Gibbs sampling algorithm starts with random assignment of word tokens to topics. Each Gibbs sample consists of topic assignments to all of the word tokens in the corpus. Samples before the “burn-in”¹⁰ period are discarded due to poor estimates of the posterior probability. After the “burn-in” period, successive Gibbs samples start to approximate the posterior distributions over topic assignments. A number of Gibbs samples are preserved at regular intervals to prevent correlations between samples (T. Griffiths, 2002, T. L. Griffiths & Steyvers, 2004, Steyvers & Griffiths, 2005).

The word-topic ϕ and topic-document θ distribution can be obtained using Equation (3) and (4):

$$\hat{\phi}_j^{(w)} = \frac{n_j^{(w)} + \beta}{n_j^{(0)} + W\beta}$$

$$\hat{\theta}_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_j^{(d)} + T\alpha}$$

The values correspond to the predictive distributions of sampling a new word token w from a topic j , and sampling a new word token w from a topic j in a document d (For detailed derivation of the Equation (2), (3), and (4) refer to (T. Griffiths, 2002)).

LEARNING RELATIONS IN TERMINOLOGICAL ONTOLOGIES

The task of learning terminological ontology can be divided to two sub-tasks: learning concepts and relations. The problem of learning concepts in various domains has been well studied in existing work as discussed in Section “Related Work”, whereas relation learning is less discussed. The main method for relation learning in existing work is the Data Co-occurrence Analysis: in (Sanderson

& Croft, 1999) the method establishes subsumption relation based on a simple assumption to exploit first-order co-occurrence; in (Diederich & Balke, 2007) the method constructs concept hierarchies based on identifying high-order data co-occurrence. In a sense, both of the methods attempt to build concept hierarchies without content analysis and rely solely on the co-occurrence of data or concepts, making them ineffective in situations where desired concepts do not co-occur with each other frequently.

In contrast, our method first represents concepts (e.g., concepts in computer science literature) identified in the underlying corpus as documents¹¹ and then performs content analysis using Latent Dirichlet Allocation which has been proved as a superior model for document modelling. By using it as an effective dimension reduction technique, concept representation is reduced to a vector of topics extracted by LDA. Finally, the concepts are organised into terminological ontologies using an iterative algorithm based on the “broader” relation definition. The definition is named as “Information Theory Principle for Concept Relationship”. The intuition is that by exploiting superior performance of topic models, semantic relations between words and documents can be captured in terms of probabilistic topics. Since concepts in our methods are represented as documents, semantic relations between concepts can then be captured based on modelling relations between topic distributions of the documents.

Throughout the paper, we limit the scope of our discussion to learning terminological ontologies in the domain of scientific research, and in particular, computer science, though it can be naturally applied to other research areas such as medical science. Differing from the work developed in (Diederich & Balke, 2007, Zavitsanos, Paliouras, Vouros, & Petridis, 2007) which exclusively learns concept hierarchies, our objective is to learn ontologies with respect to two SKOS relations (i.e., “broader” and “related”).

CONCEPT REPRESENTATION

Concepts in various scientific research areas often have a form of noun-phrase in which several words together form a specific meaning that is often well understood by human. For example, we understand that “Support Vector Machines” and “Latent Dirichlet Allocation” are two specific techniques from Artificial Intelligence, where the former is widely adopted for classification and the latter for document modelling. However, the task of understanding meanings of these concepts and distinguishing between similar concepts is difficult for machines because they do not possess human cognition. Though the probabilistic topic models pLSA and LDA, as well as LSI, provide means for computing similarities between individual word tokens, they are not suitable to be applied to concepts in form of noun-phrases. Decomposing a noun-phrase representation of a concept into individual word tokens destroys the intentional meaning of that concept. The concept “Artificial Neural Network” denotes a specific technique in Artificial Intelligence, developed by computer scientists inspired by network of neurons which work together to perform some functions in human body. The word “Neural” means “relating to the nervous system”, and “Network” is an ambiguous term (polysemy) and has different interpretations in different contexts such as Social Science, Computer Science, or Medical Science. The meaning of a concept is often completely different from the combined meanings of its constituents (i.e., individual tokens).

We represent concepts using the documents which are annotated by the concept mentions in the text corpus. Collectively, those word mentions (across documents) form a stable context which is able to explain the meaning of a concept. The idea is in line with the so-called “distributional hypothesis” (Harris, 1968) which states that similar words tend to appear in similar contexts. This is also similar to the idea of “Word Window” where

words within a fixed size window centered on a particular word are used as “context” of that word.

In some applications using LDA, the learned topic-word distributions are directly interpreted as topics (T. L. Griffiths & Steyvers, 2004, Zazvitsanos et al., 2007). For the purpose of learning ontology or concept hierarchy, the problem with direct interpretation is that one has to manually associate labels to those learned topics in order to provide meaningful interpretation. Some of the learned topics are expressed as distribution of intuitive word tokens, making the topic-label matching straightforward. While for other topics the label matching is difficult because the topics are expressed in distribution of general, non-sense, or even ambiguous word tokens. Intuitively, less latent topics generate more general concepts (and more topics generate more specific concepts), however, relations between topics learned using different number of latent classes can not be established in a precise way. Furthermore, the relations are often difficult to establish using automated approaches. On the contrary, representing concepts as documents does not suffer from these problems. Once the concepts are represented using documents, relations between them can be approximated by the relations between the representing documents in the low-dimension latent semantic space.

INFORMATION THEORY PRINCIPLE FOR CONCEPT RELATIONSHIP

As mentioned earlier, the SKOS model provides a convenient framework for expressing scientific terms in an ontological form, differing from traditional concept hierarchies in which concepts are organised in a tree-like structure. Our objective is to learn two relations: “broader” (“narrower” is the inverse property of “broader”), and “related” defined in SKOS. The “broader” relation can be interpreted as “more general than” (“nar-

rower” as “more specific than”) and is learned using a top-down approach. The mechanism for establishing “broader” relation between two concepts, called “Information Theory Principle for Concept Relationship” is defined based on Information Theory (MacKay, 2003). Before we give the definition of the principle, we provide some prerequisite definitions (for simplicity, these definitions are presented in mathematical form): Cosine similarity, Kullback-Leibler divergence (KL divergence, $D_{KL}(P \parallel Q)$) and Jason-Shannon Divergence (JS divergence, $D_{JS}(P \parallel Q)$) (MacKay, 2003). In the following definitions, P and Q are two probability distributions. The subscript i is used to denote the i th element in a distribution.

Definition: The Cosine similarity between two vectors with same dimension (could be probability distributions) is defined as:

$$COS(P, Q) = \frac{\sum_i P(i) \cdot Q(i)}{\sqrt{\sum_i P(i)^2} \cdot \sqrt{\sum_i Q(i)^2}}$$

Definition: The Kullback-Leibler divergence or relative entropy between two probability distributions $P(x)$ and $Q(x)$ is defined as:

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

The KL divergence is an asymmetric function, which means that $D_{KL}(P \parallel Q)$ is not equal to $D_{KL}(Q \parallel P)$. As such, the function is not a real metric and cannot be used as dissimilarity measures. Following the Gibbs inequity (MacKay, 2003), all values of KL divergence $D_{KL} > 0.0$

Definition: The Jason-Shannon divergence between two probability distributions $P(x)$ and $Q(x)$ is defined as:

$$D_{JS}(P \parallel Q) = \frac{1}{2} D_{KL}(P \parallel Q) + \frac{1}{2} D_{KL}(Q \parallel P)$$

The JS divergence is a symmetric function defined over the KL divergence and is a popular metric for calculating dissimilarity measures.

We define the “Information Theory Principle for Concept Relationship” as the following (Wang, Barnaghi, & Bargiela, 2009):

definition: A concept C_p is broader than a concept C_q if the following two conditions hold:

1. (Similarity condition) the similarity measure between them is greater than certain threshold TH_s (or divergence measure is less than certain threshold TH_d), and
2. (Divergence difference condition) the difference between Kullback-Leibler divergence measures $D_{KL}(P \parallel Q) - D_{KL}(Q \parallel P) < 0$.

In the context of this paper, P and Q stand for probabilistic distributions of latent topics for concepts C_p and C_q respectively. The similarity or divergence measures can be calculated using Cosine similarity or Jensen-Shannon divergence measures.

CONNECTION TO OTHER THEORIES

In most of the work reported in literature, the symmetric metric JS divergence is used for comparing dissimilarity between two vectors (or probability distributions) (Maedche et al., 2002, Biemann, 2005). Due to its asymmetry, KL divergence is less utilised for quantitative measures. One of explanations for the KL divergence $D_{KL}(P \parallel Q)$ is that P normally represents the

“true” distribution of data, while Q represents a practical approximation of P . In other words, it is the average “surprise” of an incoming message drawn from distribution Q when it is expected that it comes from the true distribution P . The second condition or the divergence difference condition states that given the similarity condition holds, if the KL divergence $D_{KL}(P \parallel Q) - D_{KL}(Q \parallel P) < 0$, then the concept C_p is said to be a broader concept of C_q . An example would provide an intuitive explanation: a source transmits a message to a receiver who has been informed that the topic of the message is about “Ontology”. The receiver is then expecting that the message is about the broad research topic “Ontology”. However, after inspecting its content the receiver realises that exact topic is “Ontology Learning” (for example, ontology learning using Natural Language Processing techniques which is more related to Natural Language Processing rather than ontology). The receiver has an amount of “surprise”, however, the quantity of the “surprise” is less than the amount after s/he observes that the paper is on “Ontology” while he has been told that the paper is about “Ontology Learning”.

We exploit the asymmetry of KL divergence measure between two probability distributions to determine whether the “broader” relation holds between two concepts. Compared to the assumption made by Sanderson *et al* (Sanderson & Croft, 1999) in which “a term A subsumes B if the documents in which B occurs are (or nearly) a subset of the documents in which A occurs”, our definition can be explained intuitively by the Information Theory (MacKay, 2003) and provides a probabilistic foundation for quantitative measure. Recently, the theory of “surprise” proposed in (Itti & Baldi, 2006) provides another method for measuring “surprise” of information. The quantity is defined as the KL divergence of prior and posterior distribution of a random variable. Our

definition is similar to the theory, while it measures the difference between two divergence values of “true” and “observed” distributions and does not involve calculation of posterior probability.

In practice, the divergence difference condition cannot be satisfied exactly and the rigorous condition results in low recall. In our experiment, we assume there exists certain amount of noise in the input data. Thus, another empirical parameter TH_N (called the noise factor) is introduced into the second inequity to balance the effect of the noise, resulting in the equation $D_{KL}(P \parallel Q) - D_{KL}(Q \parallel P) < TH_N$. The introduction of the constant TH_N increases the recall, while the precision is only reduced slightly (The recall, precision, and F1 measures of the experimental results are illustrated in Section “Evaluation”).

ONTOLOGY LEARNING ALGORITHM

The “related” function defined in the SKOS model is rather ambiguous and in this paper it is interpreted as follows: if similarity between two concepts satisfies the similarity condition while not the divergence difference condition, the two concepts are said to be related. The intuition is that even though a “broader” relation cannot be established between them, they are very similar to each other. Together with the principle for establishing “broader” relationship defined in the previous section, the next step is to automatically organising concepts into hierarchies and then terminological ontologies. An additional constraint on the concept hierarchy to be learned is that a node can only have one parent.

We have developed a recursive algorithm which is named as Global Similarity Hierarchy Learning (GSHL). In another work, we also developed an algorithm called Local Similarity Hierarchy Learning (LSHL) which is only able

to learn concept hierarchies. Comparison of the two algorithms can be found in (Wang et al., 2009). To learn “broader” relations, the algorithm recursively searches for the most similar concepts of the current “root” concept (the root node is specified by users) and eliminates those do not fulfill the divergence difference condition. The criteria for termination is either pre-defined number of iterations is reached or all concepts have been organised into an ontology. There are some parameters defined in the algorithms whose values can be modified to generate different ontology structures. Effect of these parameter values on precision of learned ontologies are discussed in Section “Evaluation”. The parameters defined in the algorithm are listed as follows:

1. N_c — The number of topics or classes used to learn parameters in LDA model.
2. M_c — The maximum number of designated sub-nodes for a particular node.
3. TH_s and TH_D — The thresholds for similarity and divergence measures.
4. TH_N — The noise factor, which is the threshold for the difference between two KL divergence measures $D_{KL}(P || Q)$ and $D_{KL}(Q || P)$.
5. I — Maximum number of iterations.

The constant parameters TH_s , TH_D , TH_N can be modified to generate different results and they have direct effect on recall, precision and F1 measures. The parameter N_c defines the dimension of the semantic space and is used to identify appropriate number of latent classes in LDA learning. The parameter M_c is defined to assess the optimum number of sub-nodes of a particular node. In our experiment the number is assumed to be between 5 to 10. The result shows that the constant has no effect on those concepts which have less than 5 sub concepts (concepts which have more than 10 sub concepts are rare in our

sample dataset). In addition, three pair-wise similarity and divergence matrices (denoted using M) are pre-calculated to improve performance of the algorithms.

Global Similarity Hierarchy Learning

The ontology learning algorithm first specifies the root concept and adds it into the Processing Vector, V . The vector V_{temp} stores most similar concepts of the current “root” node using either Cosine similarity measure or JS divergence measure. Then the algorithm eliminates those concepts following a two-step procedure: first, those concepts whose KL divergence values with the current “root” do not satisfy the divergence difference condition are removed from V_{temp} . Then, for each concept left in the V_{temp} , only if the similarity value between the concept and the current “root” is greater than similarity value between the concept and any of the siblings of the current “root”, a “broader” relation statement is asserted. Otherwise, a “related” relation statement is asserted. The pseudo-code is given in Figure 1. Algorithm 1: where the function $Sim(a, b)$ returns similarity (i.e., Cosine similarity) or divergence (JS divergence) values between concepts a and b . The function $Sibling(root)$ returns the siblings of the current “node”. To establish “broader” relation between two concepts, the algorithm searches for the most similar concepts “globally” because of which we name it as Global Similarity Hierarchy Learning algorithm. After testing the divergence difference condition for a topic and the current root topic, the algorithm will conduct another search for the most similar topics among siblings of the current root. The time complexity of GSHL algorithm is $O(m^2 \cdot n^2)$, where $m = M_c$, $n = N$, and $n \gg M_c$. In (Wang et al., 2009), we also analysed the time complexity of another ontology learning approach ($O(n^3)$) (Zavitsanos et al., 2007) and reported that our algorithm is more

Figure 1.

Algorithm 1 *GSHL*(*root*)

Require: Initialise V , M_s , I , TH_s , TH_d , TH_n , and M_c .
Ensure: A terminological ontology with “broader” and “related” relations.

- 1: Initialise V , M_s , I , TH_s , TH_d , TH_n , and M_c ;
- 2: **while** ($i < I$ and V is not empty) **do**
- 3: Add current root into V ;
- 4: Select most similar M_c nodes of root from M_s ;
- 5: Add similar nodes into V_{temp} ;
- 6: Remove nodes in V_{temp} against Definition 2;
- 7: **for** (all nodes n_i in V_{temp}) **do**
- 8: **if** ($\text{Sim}(n_i, \text{root}) > \text{Sim}(n_i, \text{Sibling}(\text{root}))$) **then**
- 9: Assert broader relations between root and topic n_i ;
- 10: **else**
- 11: Assert related relation between root and topic n_i ;
- 12: **end if**
- 13: Move topic n_i from V_{temp} to V ;
- 14: Increment i by 1;
- 15: **end for**
- 16: Remove current root from V ;
- 17: **end while**

efficient. In our experiment, the best precision is achieved using GSHL with 40 classes for LDA training (see Section “Evaluation” for evaluation details). The best precision for statements of “broader” relation is about 86% and for “related” relation is up to 90%.

EXPERIMENT

The experiment of using the GSHL algorithm to learn terminological ontologies with LDA models consists of the following four steps:

1. Dataset preparation and concepts extraction,
2. Learning LDA models with different number of classes,
3. Folding-in documents of concepts to learned LDA models,
4. Running GSHL algorithms to learn terminological ontologies.

Each of the steps is elaborated in the subsequent sections.

DATASET PREPARATION AND CONCEPT EXTRACTION

Around 5,000 Web pages containing abstracts of published work (conference paper, journal articles, and book chapters) in the Semantic Web research area are collected from the ACM digital library¹². A Web page scraper is developed using the NekoHTML¹³ HTML parser to extract relevant information, such as abstract, category annotation using terms from the ACM classification tree, user specified keywords, and related bibliographic information. Annotations of an article in a Web page normally consists of three types of terms: terms from the ACM classification tree, general terms (such as “Algorithm” and “Experiment”), and user supplied keywords. The ACM classification tree is maintained manually and seems a bit outdated for some research areas, for example, the term

“Semantic Web” does not exist in the tree. By using the user specified keywords the classification tree can be extended to deeper levels and expanded to broader range. The concept extraction process is done using a simple statistics on these keywords, that is, keywords appeared more than ten times in the whole corpus are regarded as stable concepts to be organised in the ontology. The approach for extracting concepts is similar to the one used in (Diederich & Balke, 2007). To increase recall of concepts extraction, a simple syntactical method is used to identify synonyms and plural form of some keywords, for example, ontology and ontologies. In total 80 concepts are extracted.

The corpus is further processed using some shallow natural language processing techniques. The standard English stopwords as well as a list of customised stopwords (which consists of words such as “1977” and “1st”) are removed. The Stanford Log-linear Part-Of-Speech Tagger¹⁴ is used to POS-tag all documents in the corpus and only nouns, verbs, and adjectives are preserved for training LDA models. All words are also stemmed using the Porter’s stemmer¹⁵ packaged in the Apache’s Lucene Java¹⁶, resulting in approximately 7000 unique words in total.

The extracted concepts are represented using documents which are annotated by the concepts. There are two main reasons for doing this. First, LDA is an efficient dimension reduction technique for documents, and it is able to capture the semantic relations between words, topics, and documents. A new document can be conveniently folded-in to the LDA model and represented using topic distributions. Second, as discussed earlier, a document represents a concrete context in which a concept is mentioned and explained.

LEARNING LDA MODELS

We use the Gibbs sampling algorithm provided by LingPipe¹⁷ to learn LDA models (LingPipe is a Java library for linguistic analysis). In the

experiment, seven LDA models are trained using different number of classes (i.e., from 30 to 90) in order to find out reasonable number of classes that can achieve better precision measures. The values of parameters for training LDA models are set as follows:

$$\alpha = 0.1,$$

$$\beta = 0.01,$$

$$numberofGibbssamples = 1000,$$

$$burn - inepochs = 1000,$$

$$SampleLag = 1.$$

The values of α and β are selected according to the experiment presented in (Steyvers & Griffiths, 2005) which shows satisfying results. The Gibbs sampling algorithm starts with random assignment of a topic to every word token in the corpus, every subsequent assignment of topic is determined by keeping previous assignments fixed (a complete Gibbs sample consists of topic assignment to every word token in the corpus). The algorithm is then run for specified number of times (2000 in our experiment). The first 1000 Gibbs samples are discarded due to poor estimates of the target distribution. After the “burn-in” period (assume after the first 1000 Gibbs samples), the subsequent Gibbs samples start to approach the target distribution. Another 1000 Gibbs samples are saved at regular intervals to prevent correlations between samples (T. Griffiths, 2002, T. L. Griffiths & Steyvers, 2004, Steyvers & Griffiths, 2005). The probability values of $p(w | z)$, and $p(z | d)$ are saved for purpose of folding in new documents, i.e., the documents which represent concepts.

Figure 2 shows the LDA model learned using 30 topics. Only the top-10 words with highest

topic-word probability values are shown. The first column is the index of word in the corpus, the second column is the stemmed word, and the third column is the topic-word probability. Four topics (topic 2, 5, 10, and 25) are excerpted from all the topic-word distributions and shown here. Topic 2 can be interpreted as “Agent” and Topic 5 is about “Ontology”. However, it is difficult to associate concepts or literal labels to the Topic 10 and 25. Figure 3 shows the LDA model learned using 90 topics. Topic 0 is related to “Information Retrieval”, or more precisely “Ranking”. Topic 2 is about “Web Service”. As the topics learned using 30 classes, it is confusing to associate meaningful labels or extracted concepts to the Topics 12 and 87.

Similar to the examples, among all topic models learned, there are always some topics that can be interpreted using the extracted concepts or labels easily, while others are difficult to be interpreted in the Semantic Web research area. By representing concepts as documents, our method does not suffer from such a problem. Through folding-in these concept documents and using the “Information Theory Principle for Concept Relationship”, concepts can be organised into hierarchies or ontologies in an unsupervised manner.

FOLDING-IN DOCUMENTS OF CONCEPTS

The concept of document “folding-in” is first introduced in the original LSA papers (Deerwester et al., 1990, Berry, Dumais, & O’Brien, 1995) in order to represent previously unseen documents (also referred to as “pseudo-documents” or query) in the reduced latent semantic space. In conventional information retrieval models, a document is represented using a vector of all indexed words (i.e., the vector space model) (Baeza-Yates & Ribeiro-Neto, 1999), while in LSA, a document is approximated by a vector which can be viewed as coordinates in the reduced latent dimension

space (Deerwester et al., 1990, Berry et al., 1995). The similarity between documents can then be calculated using their low dimension vector representations. While the vectors of words are often sparse, the vectors in the low dimension space are dense (i.e., without zero values). Thus, to a great extent the problem of data sparseness or “zero-frequency” can be resolved. As such, LSA is often regarded as an effective technique to deal with the synonym and polysemy problems in information retrieval (Manning, Raghavan, & Schödtze, 2008).

In probabilistic topic models (i.e., LDA) (Steyvers & Griffiths, 2005), the document “folding-in” process follows a similar idea, which is, to calculate a lower dimension representation of previously unseen documents. In pLSA, a new document is folded in by a similar procedure as in the model learning phase using the EM algorithm. The difference is that in the folding-in phase, the EM algorithm is conditioned on the parameter $p(w | z)$ (i.e., keeping $p(w | z)$ fixed).

Similarly, the “fold-in” process in LDA is performed using the Gibbs sampling algorithm as in the learning phase¹⁸. A new document is folded into a learned LDA model by running the Gibbs sampling in which the topic-word distribution are kept constant and assigning topics to every word token in the new document.

The documents which represent the extracted concepts are treated as new documents¹⁹. We wrote a program using the Gibbs sampling to calculate document-topic distributions conditioned on the learned topic-word distributions for all new documents. Parameters settings are same as those used during LDA training. Document-topic distributions for all new documents are then used to compute pair-wise similarity (Cosine similarity) and divergence values (JS and KL divergence) before running the ontology learning algorithms.

Figure 2. Topic-word distributions learned using 30 classes

TOPIC 10				TOPIC 25		
SYMBOL	WORD	PROB		SYMBOL	WORD	PROB
6886	document	0.102		987	research	0.046
3098	structur	0.041		6078	comput	0.027
4207	multimedia	0.041		5454	commun	0.023
4572	content	0.031		4613	develop	0.020
2093	link	0.031		6441	technolog	0.019
601	adapt	0.026		1020	issu	0.017
410	hypermedia	0.025		2933	challeng	0.015
948	hypertext	0.025		5844	scienc	0.014
3452	media	0.023		4223	discuss	0.013
2787	author	0.020		5643	futur	0.011
(a) Topic 2				(c) Topic 10		

TOPIC 2				TOPIC 5		
SYMBOL	WORD	PROB		SYMBOL	WORD	PROB
737	agent	0.221		3893	ontolog	0.308
5055	system	0.081		5079	semant	0.035
5862	multi	0.043		1683	domain	0.031
1953	base	0.033		4581	map	0.028
2643	intellig	0.033		6805	concept	0.028
5454	commun	0.021		1953	base	0.025
3100	softwar	0.020		2254	approach	0.016
618	interact	0.013		68	web	0.014
1569	distribut	0.012		1644	construct	0.014
3723	architectur	0.012		2387	relat	0.013
(b) Topic 5				(d) Topic 25		

Figure 3. Topic-word distributions learned using 90 classes

TOPIC 0				TOPIC 2		
SYMBOL	WORD	PROB		SYMBOL	WORD	PROB
1873	search	0.269		6397	servic	0.280
2831	rank	0.046		3923	composit	0.160
1953	base	0.043		68	web	0.126
534	engin	0.037		4471	compos	0.034
1103	keyword	0.036		5393	autom	0.026
1767	result	0.034		5523	automat	0.023
1669	index	0.031		2254	approach	0.016
3093	user	0.031		1180	describ	0.013
5292	queri	0.031		5146	techniqu	0.012
2168	relev	0.025		5361	net	0.012
(a) Topic 0				(c) Topic 12		

TOPIC 12				TOPIC 87		
SYMBOL	WORD	PROB		SYMBOL	WORD	PROB
2971	medic	0.077		5454	commun	0.230
3551	biomed	0.039		6170	wiki	0.051
1410	clinic	0.031		1613	share	0.031
624	health	0.027		4697	paper	0.027
6230	patient	0.027		6400	director	0.026
5500	guidelin	0.026		3901	practic	0.024
987	research	0.024		4223	discuss	0.024
5639	terminolog	0.024		1180	describ	0.023
5315	care	0.023		5814	extens	0.022
6642	us	0.022		2836	collabor	0.021
(b) Topic 2				(d) Topic 87		

LEARNING TERMINOLOGICAL ONTOLOGIES

Ontology learning using GSHL is straightforward once the LDA models are learned and concept documents are folded-in. The algorithm is executed using different parameter values (for example, the number of classes used for training LDA, similarity and divergence threshold values and maximum number of sub-nodes) to find optimum parameter combinations. According to our experiment, satisfying results are achieved by setting the range of similarity threshold $TH_s \in [0.5, 0.75]$, the divergence threshold $TH_d \in [0.25, 0.45]$, and the noise factor $TH_n \in [0.3, 0.5]$. The best result in terms of precision is achieved when Cosine similarity measure and 40 classes (for training LDA) are used. In (Wang et al., 2009), we reported that in terms of “broader” relation learning, the GSHL algorithm performs slightly better than LSHL in almost all cases. Another finding is that in most cases, the GSHL algorithm performs better when Cosine similarity measure is used, while the LSHL algorithm performs better with JS divergence measure. In a different experiment, we used the GSHL algorithm to learn a number of ontologies based on another topic model, pLSA, using same number of classes. The results show that overall precision of the ontological statements learned using LDA model is higher than those learned using pLSA.

We have re-implemented the concept hierarchy construction algorithms proposed in (Sanderson & Croft, 1999, Zavitsanos et al., 2007). The algorithms were then applied to the same dataset with similar parameter settings. The results produced using the algorithms (COO is the algorithm used in (Sanderson & Croft, 1999) and CIT is the algorithm used in (Zavitsanos et al., 2007)) are compared to ours, i.e., GSHL combined with the LDA. The precision curves are shown in Figure 3. As can be seen, our method performs notably

better than the other two in terms of recall, precision, and F1 measures.

An example of a learned ontology is provided in the following with parameter settings: GSHL algorithm, Cosine similarity measure, LDA model with 40 classes, and the maximum number of sub-nodes equal to 8. The top level concept “Semantic Web” is predicted by the algorithm being broader than the following seven concepts: “Ontology”, “RDF”, “OWL”, “Semantic Annotation”, “Semantic Web Services”, “Web”, “Reasoning”. All of the sub-concepts are correctly predicted except “Web”, the reason is that there are few documents whose topics are directly related to “Web” in the corpus. Therefore, a problem of the method is that it is unable to exclude a concept which is not subsumed by the top concept. Although it can be eliminated during the concept extraction phase, for example, manually exclusion of such concepts, we aim to solve the problem using an automated approach in the future work.

Figure 5 shows a snippet of the automatically generated ontology²⁰ centering on the concept “Semantic Web” with “broader” relations only. The concept “Ontology” is one of the sub-concepts of “Semantic Web”. In this particular case, the GSHL algorithm predicts that “Ontology” is a broader concept of “Ontology Evolution”, “Information Integration”, “Ontology Mapping”, “Semantic Interoperability”, and “Knowledge Acquisition”. The diagrams (Figure 5 and 6) are generated using Protege²¹ with “OntoViz” plugin.

Figure 6 shows a snippet of the ontology centering on the concept “Reasoning” with both “broader” and “related” relations. The GSHL algorithms predicts that “Reasoning” is related to “Ontology Mapping”, “Knowledge Acquisition”, “Ontology Evolution”, and “Knowledge Representation”.

Figure 4. Precision measures between COO, CIT, GSHL+LDA, and GSHL+pLSA

Figure 5. A snippet of the learned ontology centering on the concept “Semantic Web” with “broader than” relations

Figure 6. A snippet of the ontology centering on the concept “Reasoning” which is a sub-concept of “Semantic Web” with “broader” and “related” relations

EVALUATION

The learned ontologies are transformed into the RDF format with respect to the SKOS model and evaluated by domain experts. The following sections report the evaluation results and also discuss the effect of different parameter values on the results.

EVALUATION METHODS

We use recall, precision and F1 measures (Manning et al., 2008) to assess the correctness of learned ontology statements. These metrics are popular measures of search performance for text retrieval. For our evaluation purpose, these metrics are defined as follows:

$$R = \frac{n_{tc}}{N_{cs}}$$

where n_{tc} is the number of correct learned statements given by the algorithms, N_{cs} is the total number of correct statements. We assume that a concept can only have one broader concept, thus the value of N_{cs} is equal to $N_c - 1$, where N_c is the total number of topics.

Precision is defined as:

$$P = \frac{n_{tc}}{N_{ls}}$$

where n_{tc} is the number of correct learned statements, N_{ls} is the total number of learned state-

Table 1. Precision of ontology statements generated using different parameter values

Algorithm Settings\LDA classes			30	40	50	60	70	80	90	Average
5	LSHL	COS	0.684	0.694	0.645	0.737	0.685	0.743	0.815	0.715
		JS	0.813	0.653	0.657	0.788	0.667	0.721	0.661	0.709
	GSHL	COS	0.697	0.791	0.693	0.73	0.806	0.767	0.734	0.745
		JS	0.787	0.746	0.746	0.828	0.769	0.738	0.729	0.763
6	LSHL	COS	0.697	0.764	0.724	0.724	0.671	0.716	0.815	0.73
		JS	0.828	0.625	0.589	0.833	0.689	0.662	0.621	0.692
	GSHL	COS	0.684	0.866	0.763	0.743	0.778	0.681	0.781	0.757
		JS	0.787	0.803	0.697	0.81	0.761	0.738	0.767	0.766
7	LSHL	COS	0.724	0.75	0.724	0.803	0.685	0.743	0.723	0.736
		JS	0.844	0.712	0.644	0.773	0.608	0.647	0.593	0.689
	GSHL	COS	0.684	0.851	0.684	0.733	0.792	0.694	0.703	0.734
		JS	0.803	0.789	0.703	0.857	0.746	0.742	0.783	0.775
8	LSHL	COS	0.724	0.778	0.684	0.816	0.685	0.73	0.769	0.741
		JS	0.875	0.685	0.644	0.776	0.649	0.706	0.576	0.702
	GSHL	COS	0.71	0.866	0.737	0.787	0.792	0.681	0.683	0.751
		JS	0.816	0.764	0.697	0.833	0.731	0.723	0.767	0.762
9	LSHL	COS	0.724	0.736	0.671	0.789	0.685	0.73	0.776	0.73
		JS	0.859	0.712	0.603	0.806	0.608	0.691	0.55	0.69
	GSHL	COS	0.71	0.866	0.763	0.829	0.792	0.792	0.646	0.771
		JS	0.803	0.736	0.697	0.848	0.716	0.723	0.783	0.758
10	LSHL	COS	0.724	0.75	0.697	0.803	0.658	0.73	0.761	0.732
		JS	0.828	0.698	0.616	0.806	0.568	0.643	0.55	0.673
	GSHL	COS	0.789	0.836	0.75	0.803	0.75	0.806	0.785	0.788
		JS	0.803	0.722	0.716	0.848	0.735	0.754	0.78	0.765
Average			0.767	0.758	0.689	0.796	0.709	0.721	0.715	-

ments by the algorithms. The results are marked as true, only if all domain experts give positive confirmation of the result, otherwise the statement is regarded as false. The difference with the precision measure used in Information Retrieval is that ranking of results does not need to be taken into account. For this reason, ad-hoc measures such as precision at n and average precision which are widely used to measure the ranking of relevant results are not used.

The F1 measure is defined as a harmonic mean of recall and precision to balance effects of both, same as the one used in information retrieval.

$$F1 = \frac{2 * R * P}{R + P}$$

EVALUATION OF RESULTS FOR “BROADER” RELATIONS

Seven LDA models are trained using different number of latent classes (i.e., 30, 40, 50, 60, 70, 80, and 90). For each model, ontologies are learned using different maximum number of sub-nodes (i.e., from 5 to 10). The two algorithms LSHL

and GSHL are tested with Cosine similarity and JS divergence measures respectively (details of the LSHL algorithm can be found in (Wang et al., 2009)). The precision values for all ontological statements learned are shown in Table 1.

In our previous work, ontologies learned based on the pLSA and LDA models have been compared in terms of recall, precision, and F1 (Wang et al., 2009). The results show that the highest precision measure of pLSA is 81.1% for the “broader” relation, and the lowest precision is about 31.5%. Precision with LDA has the highest value (87.5%) when 30 classes are used in training, the lowest precision is about 55% when 90 classes are used in training. Overall precision measure of ontologies learned using the LDA model is higher than those learned using the pLSA model. Another finding is that structure and precision measure of the ontologies learned using LDA are not affected as significantly as using pLSA when number of training classes are changed.

The performance difference between the two probabilistic topic models probably attributes to the difference of their capability of generalising to new documents. In a sense, the pLSA is incomplete because it does not provide probabilistic modelling at the document level. New documents are assumed to have similar semantic structures as the training documents, which is not always true in many situations, resulting in poor topic modelling for those new documents. In contrast, LDA is a well defined probabilistic model and generalises easily to previously unseen documents.

EVALUATION OF RESULTS FOR “RELATED” RELATIONS

The establishment of “broader” relations among concepts forms a concept hierarchy, which can be seen as the “back bone” of a terminological ontology. The presence of “related” relation interconnect concepts on same or different levels, forming a graph structure. In fact, research (especially

in computer science) in one area (i.e., Semantic Web) often utilises techniques from others, such as artificial neural network borrows ideas from biological science, and computational linguistics emerges from joint research of computer science and natural language processing. Intuitively, concepts are also interconnected with and often dependent on each other. A terminological ontology is able to reflect the interaction and model the hierarchical information in a more complete manner than a typical concept or topic hierarchy.

The “related” relation defined in the SKOS model is an ambiguous term which is not formally defined. In our method, we interpreted the “related” relation as being hold between two concepts if the similarity condition in Definition holds while the divergence difference condition does not. The idea is that related concepts tend to have certain amount of similarity (for example, two concepts are mentioned in a similar context), even if the “subsumption” relation can not be established between them.

Statements of “related” relation between concepts are only generated by the GSHL algorithm because the LSHL algorithm does not perform a “global” similarity search. In the GSHL algorithm, if a concept is more similar to its current parent node’s sibling, the “subsumption” relation between the concept and its parent is substituted with a “related” relation. The evaluation of “related” relation is similar to the evaluation of the “broader” relation. As expected, the GSHL algorithm combined with Cosine measure performs better in most of the cases and the highest precision is about 90%.

EFFECT OF NUMBER OF SUB-NODES

Another important parameter in ontology learning algorithms is the maximum number of sub-nodes, which is used to predict the possible number of concepts that a concept subsumes (in our experiment, we set the parameter values from 5 to 10).

In particular, we would like to assess the effect of the parameter on the precision of statements as well as the number of same statements across ontologies. The evaluation shows that precision measure with the LDA model is not sensitive to variation of this parameter (as shown in the Table 1 the precision difference is less than 5%). Further, the ontological statements generated using different maximum number of sub-nodes vary slightly. In the learned ontologies, there are many cases where one concept is only more general than exactly one other concept. This overcomes the problem in (Zavitsanos et al., 2007) in which one concept has to subsume at least two other concepts, if there are any.

DISCUSSION AND FUTURE WORK

Learning of concept hierarchy or terminological ontology is in fact a process of automated knowledge acquisition in which the main task is to extract concepts and then connect concepts using specified relations. The current work provides an alternative method for learning two ontology relations, “broader” and “related” as specified in the SKOS ontology model. It is based on a well defined probabilistic model for analysing and exploring intrinsic relations between words and documents.

The first step in learning relations is to provide a reasonable measure that specifies conditions based on which a relation can be established. We propose such an assumption inspired by the information theory, calling it “Information Theory Principle for Concept Relationship”. The principle stipulates that two conditions, the similarity condition and divergence difference condition, must be satisfied in order to establish a “broader” relation between any two concepts. The principle borrows ideas from the information theory and specifies a quantitative measure in terms of Kullback-Leibler divergence. Another feature that distinguishes our work from existing work is that our method

is able to learn a terminological ontology instead of traditional concept hierarchies.

The extracted concepts are represented using documents which are then folded into the learned LDA models using the Gibbs sampling algorithm. Based on the concept relationship principle and the trained LDA models, the ontology learning algorithm is then used to connect concepts with “broader” and “related” relations. An extensive evaluation using recall, precision, and F1 measures has been performed by domain experts. The results show that overall accuracy of the learned ontology statements is above 70%. In the best situations, the precision is around 85%. The learned ontology statements in terms of “broader” and “related” relations are stable when parameter values are changed, such as the maximum number of sub-nodes of a concept.

The proposed approach is not dependent on any specific domains since the underlying LDA model is also domain-independent. The evaluation shows that it has achieved encouraging results with a dataset constructed from documents related to the Semantic Web research area. The future work will involve applying the method into other domains such as medical research, where terminological ontologies also have important applications. The current implementation of the method utilises a simple approach for extracting concepts from user annotated keywords in the corpus. Other methods such as information extraction (Cunningham, 2005) or machine learning based methods for identifying key phrases proposed in (Frank, Paynter, Witten, Gutwin, & Nevill-Manning, 1999) can be used to improve concept extraction from full text. This will not only increase the number of concepts that can be extracted, but also will help producing better concept representation.

REFERENCES

- Andrieu, C., de Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to mcmc for machine learning. *Machine Learning*, 50(1-2), 5–43. doi:10.1023/A:1020281327116
- Baeza-Yates, R. A., & Ribeiro-Neto, B. A. (1999). *Modern information retrieval*. ACM Press / Addison-Wesley.
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5). doi:10.1038/scientificamerican0501-34
- Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*, 37, 573–595. doi:10.1137/1037127
- Biemann, C. (2005). Ontology learning from text: A survey of methods. *LDV Forum*, 20(2), 75–93.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 993–1022. doi:10.1162/jmlr.2003.3.4-5.993
- Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7), 107–117.
- Cimiano, P. (2006). *Ontology learning and population from text: Algorithms, evaluation and applications*. Secaucus, NJ: Springer-Verlag New York, Inc.
- Cimiano, P., Pivk, A., Schmidt-Thieme, L., & Staab, S. (2005). Learning taxonomic relations from heterogeneous sources of evidence. In *Ontology learning from text: Methods, evaluation and applications* (pp. 59–73). IOS Press.
- Cimiano, P., & Staab, S. (2004). Learning by Googling. *SIGKDD Explorations*, 6(2), 24–33. doi:10.1145/1046456.1046460
- Cimiano, P., & Völker, J. (2005). Text2onto. In *Natural language processing and information systems* (pp. 227–238).
- Cunningham, H. (2005). *Information extraction, automatic* (2nd ed.). Encyclopedia of Language and Linguistics.
- D'Agostini, G. (2003). Bayesian inference in processing experimental data: Principles and basic applications. *Reports on Progress in Physics*, 66(9), 1383–1419. doi:10.1088/0034-4885/66/9/201
- Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JASIS*, 41(6), 391–407. doi:10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASII>3.0.CO;2-9
- Diederich, J., & Balke, W. T. (2007). The semantic growbag algorithm: Automatically deriving categorization systems. In *European Conference on research and advanced technology for digital libraries* (pp. 1–13).
- Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., & Nevill-Manning, C. G. (1999). Domain-specific keyphrase extraction. In *IJCAI* (p. 668–673). Morgan Kaufmann.
- Griffiths, T. (2002). *Gibbs sampling in the generative model of latent dirichlet allocation*. (Tech. Rep.). Stanford University.
- Griffiths, T., & Steyvers, M. (2002). A probabilistic approach to semantic representation. In *Proceedings of the 24th annual Conference of the cognitive science society*.
- Griffiths, T. L., & Steyvers, M. (2004, April). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(1), 5228–5235. doi:10.1073/pnas.0307752101
- Gruber, T. R. (2003). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199–220. doi:10.1006/knac.1993.1008

- Harris, Z. (1968). *Mathematical structures of language*. Wiley.
- Hearst, M. A. (1992). *Automatic acquisition of hyponyms from large text corpora*. In International Conference on computational linguistics (pp. 539-545).
- Hofmann, T. (1999a). Probabilistic latent semantic analysis. In *Proceedings of uncertainty in artificial intelligence* (pp. 289-296).
- Hofmann, T. (1999b). Probabilistic latent semantic indexing. In *Annual ACM SIGIR Conference* (pp. 50-57).
- Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2), 177-196. doi:10.1023/A:1007617005950
- Itti, L., & Baldi, P. (2006). Bayesian surprise attracts human attention. [Cambridge, MA: MIT Press.]. *Advances in Neural Information Processing Systems*, 19, 547-554.
- Kiryakov, A., Popov, B., Terziev, I., Manov, D., & Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2(1), 49-79. doi:10.1016/j.websem.2004.07.005
- Lemaire, B., & Denhière, G. (2006). Effects of high-order co-occurrences on word semantic similarities. *Current Psychology Letters*, 18(1).
- MacKay, D. J. (2003). *Information theory, inference, and learning algorithms*. Cambridge University Press.
- Maedche, A., Pekar, V., & Staab, S. (2002). Ontology learning part one - on discovering taxonomic relations from the Web. *Web Intelligence*, 301-322.
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
- Sanderson, M., & Croft, W. B. (1999). *Deriving concept hierarchies from text* (pp. 206-213). In SIGIR.
- Steyvers, M., & Griffiths, T. (2005). Probabilistic topic models. In T. Landauer, D. Mcnamara, S. Dennis, & W. Kintsch (Eds.), *Latent semantic analysis: A road to meaning*. Laurence Erlbaum.
- Wang, W., Barnaghi, P., & Bargiela, A. (2009). Probabilistic topic models for learning terminological ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 99.
- Zavitsanos, E., Paliouras, G., Vouros, G. A., & Petridis, S. (2007). Discovering subsumption hierarchies of ontology concepts from text corpora. In *Wi '07: Proceedings of the IEEE/WIC/ACM International Conference on Web intelligence* (pp. 402-408). Washington, DC: IEEE Computer Society.

KEY TERMS AND DEFINITIONS

Ontology Learning: is the task to build ontologies from scratch, enriching adapting an existing ontology in (semi)automatic fashion using information from different sources.

Terminological Ontologies: is an ontology whose categories need not be fully specified by axioms and definitions.

SKOS: is an area of work developing specifications and standards to support the use of knowledge organization systems such as thesauri, classification schemes, subject heading lists and taxonomies within the framework of the Semantic Web.

Semantic Web: describes methods and technologies to allow machines to interpret meaning or “semantics” of data on the World Wide Web. According to Tim Berners-Lee (Berners-Lee et al., 2001), the Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, bet-

ter enabling computers and people to work in cooperation.

Probabilistic Topic Model: are models based on Bayesian inference for various applications, in particular, document modelling.

Latent Dirichlet Allocation: is a statistical generative model for analysing latent topics in documents.

Information Theory Principle for Concept Relationship: is a principle for establishing relations, in particular, SKOS relations, for the purpose of learning terminological ontologies.

Gibbs Sampling: is an algorithm to generate a sequence of samples from the joint probability distribution of two or more random variables.

ENDNOTES

- ¹ <http://www.w3.org/TR/skos-primer/>
- ² <http://www.jfsowa.com/ontology/gloss.htm>
- ³ In Semantic Web terminology “broader” and “narrower” relations are said to be inverse properties with each other.
- ⁴ <http://www.acm.org/class/1998/>
- ⁵ <http://www.nlm.nih.gov/mesh/>
- ⁶ In Cimiano’s publication (Cimiano, 2006), the word “concept” is used to denote “class”, where “term” denotes instance. In a traditional concept hierarchy, no explicit distinctiveness has been made between “class” and “instance”.
- ⁷ High-order co-occurrence is often used to analyse strong associations between words even though they do not co-occur in the same documents (Lemaire & Denhière, 2006).
- ⁸ <http://dblp.l3s.de/>
- ⁹ These parameters are derived from the parameters specified in the generative process using the Bayes’ rule. Derivation of equa-

tions on parameter estimation in pLSA can be found in (Hofmann, 2001).

- ¹⁰ In Gibbs samplers, it is difficult to determine when the chain approaches stationarity. In practice, one often discards an initial set of samples (e.g., 1000) to avoid starting biases (Andrieu et al., 2003).
- ¹¹ A document constructed from words in the vicinity of the concept is a better representation for that concept. In a sense, the document is able to explain the concept and can be viewed as a collection of context in which the concept is mentioned. This in fact coincides with Harris’ “distributional hypothesis” assumption.
- ¹² <http://portal.acm.org>
- ¹³ <http://sourceforge.net/projects/nekohtml>
- ¹⁴ <http://nlp.stanford.edu/software/tagger.shtml>
- ¹⁵ [http://tartarus.org:/\\$\sim\\$:>>martin/Porter-Stemmer/](http://tartarus.org:/$\sim$:>>martin/Porter-Stemmer/)
- ¹⁶ <http://lucene.apache.org/>
- ¹⁷ <http://alias-i.com/lingpipe/>
- ¹⁸ If the LDA model learning uses variational inference algorithms as proposed in the original LDA paper (Blei et al., 2003), one can condition on the distributions over words for the topics and run the variational algorithm to infer the distribution over topics for the query.
- ¹⁹ The documents representing the concepts are not the original documents used in the LDA training; thus they are treated as new documents and folded-in to the trained LDA models.
- ²⁰ Due to the large number of concepts and concept relations, it is not possible to provide visualisation of the whole ontology.
- ²¹ <http://protege.stanford.edu/>