Probabilistic Topic Models for Learning Terminological Ontologies

Wang Wei, Payam Barnaghi, *Member, IEEE,* and Andrzej Bargiela, *Member, IEEE*

Abstract—Probabilistic topic models were originally developed and utilised for document modeling and topic extraction in Information Retrieval. In this paper we describe a new approach for automatic learning of terminological ontologies from text corpus based on such models. In our approach, topic models are used as efficient dimension reduction techniques, which are able to capture semantic relationships between word-topic and topic-document interpreted in terms of probability distributions. We propose two algorithms for learning terminological ontologies using the principle of topic relationship and exploiting information theory with the probabilistic topic models learned. Experiments with different model parameters were conducted and learned ontology statements were evaluated by the domain experts. We have also compared the results of our method with two existing concept hierarchy learning methods on the same dataset. The study shows that our method outperforms other methods in terms of recall and precision measures. The precision level of the learned ontology is sufficient for it to be deployed for the purpose of browsing, navigation, and information search and retrieval in digital libraries.

Index Terms—Knowledge acquisition, Ontology learning, Ontology, Probabilistic topic models.

1 Introduction

Knowledge acquisition and management are key to developing successful applications on the semantic Web [1] aiming at enabling people and computers to interact more naturally. While some structured and semi-structured data and information (i.e., data in relational database) can be transformed into structured knowledge (i.e., RDF[1] datasets) for semantic Web applications, the transformation of other data might be difficult. This is due to the shortage of general methods for generating structured or semi-structured data. Also, many data sources choose to protect key data from being accessed by the public, so as to maintain their competitive advantages.

In this context automated ontology learning is regarded as a promising approach to acquiring knowledge from unstructured text. In recent years, researchers in the semantic Web community have applied techniques from Natural Language Processing [2], [3], [4], Information Extraction [5], [6], [7], and Machine Learning [8], [9], [10] to learn different types of ontologies from text corpus. A plausible assumption is that given a sufficiently large amount of text in a domain of interest, one can distill adequate knowledge in that domain [2]. More importantly compared to manual approaches, the automated building of ontology has the advantages of fast development, cost effectiveness, and being resistant to obsolescence.

In this paper we limit the scope of our discussion to learning terminology-like ontologies with respect to the SKOS model[2] using probabilistic topic models. Although the term "terminological ontology" and "concept hierarchy" are used interchangeably in some texts, we differentiate between them and use the term terminological ontology in the sense of "concept hierarchy" enriched by "related" relationships. Another point worth noting is that we differentiate between the "concept hierarchy" and "topic hierarchy". In research communities such as knowledge management, the term "concept hierarchy" is exclusively referred to as concepts organised using "is-a" or "a-kind-of" relationships. However, in the semantic Web modelling, i.e., the SKOS model, "Concept" is defined as the top-level abstract class and a super-class of "Topic". Bearing in mind the above, the hierarchy we try to learn is in fact a "topic hierarchy" since the relationship between topics is "more specific".

The probabilistic topic models have been primarily used in document modeling, topic extraction, and classification purposes in Information Retrieval [11], [12], [13], [14]. In this paper, we explore a possibility of applying topic models in a different area, i.e., ontology learning. More specifically, given a set of concepts or topics, the objective is to learn relationships between them. With the topic models, we propose a new method for learning terminological ontologies based on the "Information Theory Principle for Concept Relationship" and topic hierarchy learning algorithms. The empirical evaluation of our method on a representative data set shows that the accuracy of our learned ontologies is up to 85% in

• *Wang Wei and Andrzej Bargiela are with the School of Computer Science, University of Nottingham (Malaysia Campus). Payam Barnaghi is with the Centre for Communication Systems Research (CCSR) at the University of Surrey.*
E-mail: eyx6ww@nottingham.edu.my, p.barnaghi@surrey.ac.uk, abb@cs.nott.ac.uk

Manuscript received Sep 4, 2008; revised April 26, 2009.

1. http://www.w3.org/RDF/

2. http://www.w3.org/TR/swbp-skos-core-guide/

terms of "broader" relationships, and 90% in terms of "related" relationships in the SKOS model. This is a notable improvement in terms of recall and precision measures, compared to the results generated using the existing concept hierarchy learning algorithms applied to the same dataset.

The rest of the paper is organised as follows. Section 2 provides an overview on existing ontology learning methods. Section 3 provides background knowledge on probabilistic topic models, in particular, probabilistic Latent Semantic Analysis and Latent Dirichlet Allocation. In Section 4 we describe in detail our ontology learning approach. We define the concept relationship principle inspired by the information theory and present two algorithms for constructing SKOS ontologies with "broader" and "related" relationships. Section 5 describes the experiment conducted including dataset preparation, model training, and ontology learning. Section 6 focuses on the evaluation of results as well as a comparison with other ontology learning methods. Additionally, we discuss in this section the effect of different parameters on the performance of our method. Section 7 concludes the paper and discusses future work.

2 Ontology Learning Methods

In the research area of the semantic Web, and more generally computer science, ontology represents a formal conceptualisation of particular domains of knowledge. According to Sowa[3], ontologies can be categorised into three types: formal, prototype-based, and terminological ontologies. Our objective is to automatically learn terminological ontologies from unstructured text corpus. The task of learning ontologies can be further divided into six sub-tasks: learning terms, synonyms, concepts, concept hierarchies, relations, and rules [2]. Existing methods for these sub-tasks can be categorised into four groups: Lexico-syntactic based approach [2], [3], [4], [15], Information Extraction [6], [7], Machine Learning [8], [9], [10], and Data Co-occurrence Analysis [16], [17] (A survey of ontology learning methods is also provided in [18]).

2.1 Lexico-syntactic based Approach

Lexico-syntactic methods apply Natural Language Processing (NLP) techniques to learn instances and relations in general domains. The so-called Hearst-patterns [19] have been used for the purposes of concept hierarchy induction and learning taxonomic relations such as "is-a" or "part-of" (see Chapter 6 and 7 in [2] for details). A significant limitation of this method is that the patterns may not appear frequently in the underlying texts which results in very low recall. To solve the problem, some methods which attempt to take advantages of existing lexicon resources to learn taxonomic (i.e., *is-a* relation) and non-taxonomic relations have been proposed. For example, the work in [3] aims to build a domain-independent and large-coverage taxonomy by using lexico-syntactic matching and the category system in the Wikipedia[4]. The method in building the Yago ontology [4] takes this approach one step further by unifying the WordNet[5] and Wikipedia to learn both taxonomic and non-taxonomic (i.e., *hasWonPrize*) relations. However, the Yago only deals with very shallow semantics as it always uses the most frequent sense in the WordNet (i.e., no deep analysis is performed). In the OntoLearn system [15], NLP techniques are used to extract and interpret terminological terms. The terms are then linked to each other using taxonomic relations to create a domain ontology, possibly attached to existing lexical resources (e.g., WordNet).

2.2 Information Extraction

Information Extraction (IE) [5] techniques, in particular, Named Entity Recognition (NER) have been utilised to automatically populate general-purpose knowledge bases [6], [7]. The limitation is that the current NER technique is only effective for recognising instances of general concepts such as "People" and "Organisation", limiting its applicability to more specific and abstract domains.

2.3 Machine Learning based Approach

Statistical learning based on classification techniques have been used to infer ontologies from unstructured text. More specifically, they were utilised to augment a thesaurus with new lexical terms, or populate an ontology with new instances [18]. Ontological terms are normally represented as vectors using words in the vicinity of the represented terms based on the Harris' distributional hypothesis, which states that "similar words tend to occur in similar contexts" [20]. The work in [8] populates a person ontology by classifying person instances into eight pre-defined subcategories. Both the local context surrounding the instances as well as global semantic information derived from the WordNet are used as features for classification. Suchanek *et al* [9] describe an approach that combines deep linguistic structures and machine learning techniques to extract given relations from web documents. Pasca *et al* [10] augment the WordNet with named entities presented in text documents by classifying them to the leaf concepts in WordNet. One of the common limitations of the machine learning-based approaches is that the learned ontologies demonstrate rather poor performance compared to alternatives [18], [3], [4].

2.4 Data Co-occurrence Analysis

Data Co-occurrence Analysis is a simple but effective approach that exploits first-order or high-order

3. http://www.jfsowa.com/ontology/gloss.htm

4. http://wikipedia.org/

5. http://wordnet.princeton.edu/

co-occurrence of data for learning concept hierarchies. Sanderson *et al* [16] propose a method based on an assumption 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". Despite its simplicity, experiments conducted on a large text corpus show that data co-occurrence gives better results then lexico-syntactic methods. Another method exploiting high-order co-occurrence (the algorithm is called GrowBag), is presented in [17]. The learned ontology has been deployed in the FacetedDBLP[6] search engine to help users exploring scientific publications.

3 Probabilistic Topic Models

Latent Semantic Analysis (LSA) [21] was introduced to overcome the shortcomings of conventional Information Retrieval (IR) techniques with regard to computation of similarity between the documents. Variants of the LSA such as probabilistic Latent Semantic Analysis (pLSA) [11] and Latent Dirichlet Allocation (LDA) [12], [13], [14] have been developed to improve the interpretation of the results generated by the LSA. These techniques are proved to be effective for document modeling and topic extraction. In the statistical models of topics, semantic properties of words and documents are represented using probabilistic (latent) topics and internal structure of the text is interpreted using word-topic and topic-document distributions. These models are also used as efficient dimension reduction techniques [11], [12], [14] in which a document is often represented using a vector of topics instead of vector of words.

In what follows we provide a brief overview of the latent semantic analysis techniques (i.e., Latent Semantic Analysis, probabilistic Latent Semantic Analysis, and Latent Dirichlet Allocation). The intuition behind our approach is that by using probabilistic topic models and representing topics as documents, semantic relationships between those topics can be captured. Details of our approach are elaborated in Section 4.

3.1 Latent Semantic Analysis

The Latent Semantic Analysis (LSA) [21] is a technique originally introduced to solve 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). The idea behind the LSA is to transform document representation in high-dimension word space to low-dimension latent semantic space to capture implicit structure in the association of terms with documents. The low-dimension semantic latent space is obtained by decomposing a large term-document matrix using Singular Value Decomposition, a technique from linear algebra. The LSA model was shown to be effective for modeling implicit semantic structures between documents and words [22]. However, the LSA has an inherent weakness in that although the words and documents can be represented as points in Euclidean space, the semantics of the model is difficult to interpret, for example, the LSA approximation of the term-document matrix may contain negative values [11].

6. http://dblp.l3s.de/

3.2 Probabilistic Latent Semantic Analysis (pLSA)

The Probabilistic Latent Semantic Analysis (also called the aspect model) [11] is a generative statistical model for analysing co-occurrence of data (originally for document and word analysis) which associates a latent variable z with an observation (i.e., each occurrence of the word w in the document d). The generative process of the pLSA is defined as:

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 $P(w|z)$.

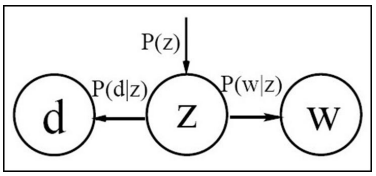


Fig. 1. Graphical representation of the pLSA model in asymmetric parameterization

The generative process of the pLSA model is illustrated in Figure 1. This shows that the document and the word are independently conditioned on the state of the latent variable.

Parameter estimation in the pLSA is based on the likelihood principle and is approximated using Expectation-Maximisation (EM) algorithm [23]. In the pLSA, the expectation step computes the posterior probabilities of the latent variables given the current estimate of the parameters (the initial values are chosen randomly). The equation can be obtained using the Bayes's rule [11]:

$$P(z|d,w) = \frac{P(z)P(w|z)P(d|z)}{\sum_{z'} P(z')P(w|z')P(d|z')} \tag{1}$$

In the maximisation step, the objective function (i.e., the expectation of the complete data log-likelihood) is maximised. Hofmann [11] shows that the Maximisation step equations can be obtained as follows:

$$P(w|z) = \frac{\sum_d n(d,w)P(z|d,w)}{\sum_{d,w'} n(d,w')P(z|d,w')} \tag{2}$$

$$P(d|z) = \frac{\sum_w n(d,w)P(z|d,w)}{\sum_{d',w} n(d',w)P(z|d',w)} \tag{3}$$

$$P(z) = \frac{\sum_{d,w} n(d,w)P(z|d,w)}{\sum_{d,w} n(d,w)} \tag{4}$$

where n is the count matrix, an element $n(d,w)$ holds the number of times that a word w appears in a document d.

3.3 Latent Dirichlet Allocation (LDA)

While the pLSA represents a significant step towards probabilistic modelling of text, it is incomplete in that it provides no probabilistic model at the level of documents [12]. In other words, the model does not have any control over how the mixture weights $p(z|d)$ are generated. The limitation leads to two problems: the number of parameters that need to be estimated grows linearly with the size of the corpus, which leads to overfitting [12]; and it is difficult to test generalisability of the model to new documents [14].

The Latent Dirichlet Allocation (LDA) [12] is an attempt to improve the pLSA by introducing a Dirichlet prior on document-topic distribution. As a conjugate prior for multinomial distributions [24], Dirichlet prior simplifies the problem of statistical inference. Steyvers and Griffiths explore a variant of the LDA by placing a symmetric Dirichlet prior on topic-word distribution [14], [13], and demonstrate how to perform parameter estimation using Gibbs sampling, a form of the Markov Chain Monte Carlo [25].

The generative process in the LDA is similar to that in the pLSA: 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 generative process can be represented using the Equation (5):

$$P(w_i) = \sum_{j=1}^{T} P(w_i|z_i = j)P(z_i = j) \tag{5}$$

where $P(z_i = j)$ is the probability that jth topic is sampled for the ith word token[7], and $P(w_i|z_i = j)$ is the probability of sampling w_i under topic j. 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 generative process of the LDA can be illustrated using a plate representation [14] as shown in Figure 2. The shaded variable w is the observed data and unshaded variables ϕ, θ, and z are latent variables which need to be estimated. The α and β are constant hyperparameters in the model and are Dirichlet priors on ϕ and θ, respectively.

There are various algorithms available for estimating parameters in the LDA, for example, the method explained in [12] introduces an approach called variational inference with EM algorithm. We adopt an approach using Gibbs sampling which is proposed in [14], [13]. The rationale behind Gibbs sampling for parameter estimation is that instead of directly estimating the topic-word $p(w|z)$ and document-topic $p(z|d)$ distributions, we estimate the posterior probability distribution over latent variable z given the observed data conditioned on topic

7. A word token is an occurrence of an indexed word in the training corpus.

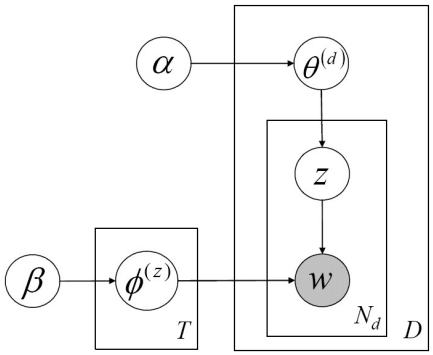


Fig. 2. Plate representation of the LDA model

assignment for all the other word tokens using Equation (6) (see [13], [14] for more details).

$$P(z_i = j|\boldsymbol{z}_{-i}, \boldsymbol{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha} \tag{6}$$

where n is a count matrix, and $n_{-i,j}^{(w_i)}$ is the count of word token w being assigned to the topic j, not including the current word token w_i. The first term in the Equation (6) represents the probability of the word w under topic j, whereas the right term in the Equation (6) represents the probability of topic j in the document d. Intuitively, 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 not only on how likely the word is associated with a topic, but also on how dominant the topic is in a document [14].

The Gibbs sampling algorithm then 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"[8] 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[9] between samples [26], [13], [14].

The Gibbs sampling algorithm approximates posterior distribution of a topic for all words. The word-topic ϕ and topic-document θ distribution can be obtained as follows:

8. In implementation of MCMC samplers, it is difficult to determining the length of the Markov chain, i.e., when the chain approaches stationarity. In practice, one often discards an initial set of samples (burn-in) to avoid starting biases [25].

9. High correlation indicates that the Markov chain is a poor mixing, and hence poor convergence.

$$\hat{\phi}_j^{(w)} = \frac{n_j^{(w)} + \beta}{n_j^{(.)} + W\beta} \quad (7)$$

$$\hat{\theta}_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_\cdot^{(d)} + T\alpha} \quad (8)$$

where n is a count matrix, $n_j^{(w)}$ is the number of word token w assigned to topic j, $n_j^{(d)}$ is the total number of word tokens assigned to j in the corpus, $n_j^{(d)}$ is the number of token assigned to j in document d, and $n_\cdot^{(d)}$ is the number of tokens in d. For detailed derivation of the Equations (6), (7), and (8) refer to [26].

3.4 Applications of Probabilistic Topic Models

The work reported in [12], [13] shows that the LDA could be successfully applied in various applications for the purposes such as identifying topics in scientific publications, classification, and collaborative filtering. Blei *et al* [12] compared performance of the LDA with three other models, namely, unigram, mixture unigram and the pLSA in terms of perplexity, a popular measure in language modelling. They reported that the LDA model showed superior performance and did not suffer from the serious overfitting problem associated with the pLSA.

4 A New Approach for Topic Hierarchy Construction

The two probabilistic topic models discussed in the previous section were developed to solve problems caused by the gap between human cognition and ambiguity of human natural language. Compared to the original LSA model, they provided a means of interpreting explicitly the relationships between document-topic and topic-word in terms of probability weights. Another significance of the probabilistic topic models is the relaxation of one-to-one relationship between the document and the topic, which is associated with the classical document modelling paradigms [13].

The specific problem we discuss in this section is that of learning terminological ontology using probabilistic topic models. We construct terminological ontology by learning a topic hierarchy and subsequently enriching it with additional relationships. In our discussion we focus on the domain of scientific literature in Computer Science research, but clearly the method is applicable to other research areas such as Medicine with the corpus of PubMed[10] documents. In general, the problem of learning topic hierarchies can be simplified as learning topics (or concepts) and relations. Differing from other work which exclusively learns hierarchies, our objective is to learn terminological ontologies with more relations (i.e., "broader" and "related" relations).

10. http://www.ncbi.nlm.nih.gov/

4.1 Problem Formalisation

In Computer Science, a topic or a concept is a complex entity which is normally well-understood by human and is often described in a form of a noun-phrase, for example, "Artificial Neural Network" and "Latent Dirichlet Allocation". However, noun phrases are too "complicated" for computers to interpret. Breaking the phrase into individual words destroys the intentional structure of the entity and the derived meanings might be totally different from the intended ones. Complex noun-phrases can be interpreted using documents, which explain those topics in a detailed manner, for example, representing a topic using words in its vicinity, or documents that are annotated with the topic. The idea is in line with the so-called "distributional hypothesis" [20] which states that similar words tend to appear in similar contexts. The context within which a noun-phrase appears contributes to the meaning of this noun-phrase. By applying topic modelling, a document vector can be transformed into a vector of latent topics in a lower dimensional space, which hopefully provides adequate representation of the document.

Having established an idea of representing topics using documents we now concentrate on answering the following questions:

What is the advantage of representing topics as documents over explicitly interpreting the learned topics? Both the pLSA and LDA learn latent topics represented as probability distributions over words. Intuitively, fewer latent topics denote more general concepts, and a greater number of latent topics denote more specific concepts. However, the precise relationship between the number of latent topics and the generality of concepts cannot be established in a precise way. That is why one has to resort to the manual assignment of labels to topics in order to provide a meaningful interpretation for ontology learning purpose, which in itself is a tedious process. In contrast, representing topics or concepts as documents (e.g., as a collection of words in the vicinity of those concepts or topics) does not suffer from this problem. Once the noun-phrases are represented using documents, similarity between them can be approximated by calculating similarity between those documents. The pLSA and LDA are efficient dimension reduction techniques for computing semantic similarity between documents and, in doing so, solving one of the most challenging problems that of "Data Sparseness". However, the problem remains how to model the "broader" and "related" relationships between topics.

How can we organise topics into a meaningful hierarchy? Probabilistic topic models learn a set of latent topics, which can be used as features in the low dimensional semantic space. Our objective is to infer relationships between topics, i.e., are the topics siblings on the same level? Or are there any parent-child relations between them? If the "broader" relationship cannot be established, can we say that they are related? To answer

these questions, we have to define a mechanism to designate the relationships. In the following section we will give formal definitions of two types of relationships: "broader" and "related", based on the Information Theory.

4.2 Information Theory Principle for Concept Relationship

We concentrate on learning topic hierarchies based on the SKOS vocabulary, in particular two relationships: "broader", and "related". The SKOS model provides a convenient framework for expressing scientific terms in an ontological form. This differs from traditional concept hierarchies in which concepts are organised in a tree-like structure. Our approach to establishing "broader" or "narrower" relations between topics follows the principle from Information Theory [27]. The definition for Kullback-Leibler divergence (KL divergence or $D_{KL}(P||Q)$) [27] (also known as Relative Entropy) is given first to underpin the principle for establishing relationships between topics.

Definition 1: The relative entropy or Kullback-Leibler divergence between two probability distributions P and Q over the same discrete space A is defined as:
$D_{\mathrm{KL}}(P\|Q) = \sum_{i \in A} P(i) \log \frac{P(i)}{Q(i)}$
Following the Gibbs inequality [27], KL divergence value is: $D_{KL} > 0$.

Definition 2: A topic C_p is broader than another topic C_q if the following two conditions hold:

- (similarity condition) Similarity measure between C_p and C_q is greater than certain threshold TH_s (or divergence measure is less than certain threshold TH_d), and
- (divergence difference condition) Difference between Kullback-Leibler divergence measures: $D_{KL}(P||Q) - D_{KL}(Q||P) < 0$.

In the above definitions, P and Q are the probabilistic distributions of latent topics for C_p and C_q respectively. The similarity or divergence measures can be calculated using Cosine similarity or Jensen-Shannon (JS) divergence measures. The similarity and divergence thresholds TH_s and TH_d are user specified parameters, the values of which can be tuned to achieve satisfactory results.

In the KL divergence measures, P normally represents the "true" distribution of data or a precise theoretical distribution, while the measure Q represents an approximation of P. The KL divergence therefore represents the expected number of bits that are wasted by encoding events from the true distribution P with a code based on the not-quite-right distribution Q. In other words, it is the average "surprise" due to incoming message being drawn from distribution Q when it is expected to arrive form the true distribution P. Intuitively, given that the first condition holds, if the KL divergence $D_{KL}(P||Q)$ is less than $D_{KL}(Q||P)$, then the topic C_p is said to be more general than the topic C_q. For example, a source sends a document to a receiver who has been informed that the document is about "Ontology", and is expecting that paper is about the broad research topic "Ontology". However, after reading the document the receiver observes that the exact topic of the document is about "Ontology Learning" (for example, ontology learning using Natural Language Processing techniques). The receiver will experience some "surprise", but the amount of "surprise" will be less than the surprise in the alternative scenario of expecting document on "Ontology Learning" but receiving a document on the broad topic of "Ontology". By exploiting the difference between the two values of "surprise" we are able to achieve a notable improvement to recall and precision in the proposed model.

4.3 Comparison to Other Theories

Most of the published works on comparing dissimilarity of linguistic objects report the use of the symmetric divergence metric JS interpreted as a linguistic distance. Our approach is different in that we use the Kullback-Leibler divergence measure, which is asymmetrical and, as such, does not measure the distance between topics. The asymmetry of the KL divergence measure is utilised instead to determine whether there is a "broader" relationship between the two topics. Compared to the assumption made by Sanderson *et al* [16] 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 approach is supported and explained by the Information Theory [27] and is rooted in a formal mathematical definition.

The work reported in [28] applies "conditional independence testing" on number of topic sets obtained through repeatedly training of the LDA models to learn concept hierarchies. One of the major limitations of this approach is that the learned topics have to be manually labeled thus placing a practical constraint on the number of concepts. Another limitation is that the method is not computationally efficient because it involves training of several topic sets. A topic set that consists of fewer probabilistic topics is assumed to be more generic than the one with more topics. However, in our experiment and evaluation, even domain experts are perplexed about deciding the relationship between two topics from two different topic sets. Furthermore, the method is also unable to infer subsumption relationship in situations where a topic only subsumes one other topic.

The emergence of the theory of "surprise" provides another formal mathematical definition on measuring "surprise" of information [29], which is defined as KL divergence of prior distribution and posterior distribution. In a sense, our definition is similar to the theory of "surprise", while the difference is that our definition

provides quantitative measure for difference between two divergences and does not involve the posterior probability distribution calculation.

In practice, the second condition specified in Definition 2 cannot be fulfilled exactly due to the noisy nature of data. In our experiment, we add into the second inequality an empirical parameter TH_n, called the noise factor: $D_{KL}(P||Q) - D_{KL}(Q||P) < TH_n$. This noise factor increases the recall measure in practical applications as explained in Section 6.1 (Recall, precision, and F1 measures are discussed in [30]).

4.4 Topic Hierarchy Construction Algorithms

Having defined the principle for establishing a relationship between topics, the next step is to organise topics into hierarchies. We have developed two recursive algorithms: Local Similarity Hierarchy Learning and Global Similarity Hierarchy Learning. The basic idea behind both algorithms is the recursive search for the most similar topics of the current "root" topic and removal of those that do not satisfy the condition on the difference of KL divergence. Both algorithms start with an initial topic as the root node and look for the top n most similar topics according to (dis)similarity measures. The KL divergence measures are then used to verify whether the second condition in Definition 2 holds. The algorithms will stop either when pre-defined number of iterations is reached or when all topics have been selected and organised. The parameters used in the algorithms are as follows:

- N — The total number of topics.
- M_c — The maximum number of sub-nodes for a particular node.
- TH_s and TH_d — The thresholds for similarity and divergence measures.
- TH_n — The noise factor, defined with the difference between two KL divergence measure $D_{KL}(P||Q)$ and $D_{KL}(Q||P)$.
- I — Maximum number of iterations.

The parameters TH_s, TH_d, TH_n are user specified constants, which are tuned to obtain desirable recall, precision and F1 values. Specifically, in our experiment we have found that setting TH_s, TH_d, and TH_n within some narrow ranges results in only slight variation of recall and precision values. The pair-wise measures of Cosine similarity, JS divergence and KL divergence are collectively denoted as the M_s matrix.

4.4.1 Local Similarity Hierarchy Learning

The Local Similarity Hierarchy Learning (LSHL) algorithm specifies the current root topic and adds it into a vector called Processing Vector, V, which stores nodes being processed by the algorithm. The vector V_{temp} is a temporary vector for storing the most similar topics of the current "root" node. The most similar M_c topics are selected from matrices M_s. Then KL divergence values between the current root topics and the topics in the V_{temp} vector are retrieved from matrices M_s and compared against the divergence difference condition defined in Definition 2. If the condition holds, a "broader" relationship is asserted between the current node and the topic in V_{temp}. Conversely, if the condition does not hold, the node will be eliminated from V_{temp}. After all the narrower topics of the current root have been found, the current root node is removed from the vector V, and all nodes in V_{temp} are moved into V. The algorithm will move to the next level of the hierarchy by making recursive calls to the same function. The algorithm will stop when either maximum number of iterations is reached or all topics have been organised into the topic hierarchy. The pseudo-code for LSHL is shown in Algorithm 1.

Algorithm 1 $LSHL(root)$

Require: V, M_s, I, TH_s, TH_d, TH_n, and M_c.
Ensure: A topic hierarchy with "broader" relation.

```
Initialise V, M_s, I, TH_s, TH_d, TH_n, and M_c;
Add current root into V;
Select most similar M_c nodes of root from M_s;
Add all similar nodes into V_temp;
Remove nodes in V_temp against Definition 2;
Assert "broader" relation between root and nodes in V_temp;
Move nodes in V_temp to V;
Remove current root node from V;
while (i < I and V is not empty) do
    LSHL(1st node in V);
    Increment i by 1;
end while
```

The output of LSHL is a knowledge base consisting of statements of "broader" relations between topics. The LHSL is a greedy algorithm, which finds the most similar topics locally and asserts a statement immediately during each iteration. It is worth noting that the LSHL algorithm is only able to learn topic hierarchies. The statements of "related" relation are not generated by the LHSL. Since for every topic the algorithm selects M_c most similar topics of the current root topic from matrices M_s, the time complexity of LSHL is $O(m \cdot n^2)$, where $m = M_c$ and $n = N$. Since $N \gg M_c$, the LSHL algorithm is more efficient than the algorithm presented in [28] which has time complexity of $O(n^3)$.

4.4.2 Global Similarity Hierarchy Learning

The Global Similarity Hierarchy Learning (GSHL) algorithm establishes both the "broader" and "related" relationships between topics using global similarity measures. Similar to LSHL, the algorithm starts with specifying the root topic and adding it into the Processing Vector, V. The vector V_{temp} stores the most similar topics of the current "root" node. The selected most similar topics in V_{temp} are filtered using a two-step procedure:

- Topics whose KL divergence values with the current "root" do not satisfy the divergence difference condition are removed from V_{temp},
- For each topic left in the V_{temp}, a "broader" relation is asserted if the similarity value between the topic and the current "root" is greater than similarity value between the topic and any of the siblings of the current "root"; otherwise, a "related" relation is asserted.

The algorithm will terminate according to the conditions specified in the while loop. The pseudo-code for GSHL is given in Algorithm 2.

Algorithm 2 $GSHL(root)$

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

```
Initialise V, M_s, I, TH_s, TH_d, TH_n, and M_c;
while (i < I and V is not empty) do
    Add current root into V;
    Select most similar M_c nodes of root from M_s;
    Add similar nodes into V_temp;
    Remove nodes in V_temp against Definition 2;
    for (all nodes n_i in V_temp) do
        if (Sim(n_i, root) > Sim(n_i, Sibling(root))) then
            Assert broader relations between root and topic n_i;
        else
            Assert related relation between root and topic n_i;
        end if
        Move topic n_i from V_temp to V;
        Increment i by 1;
    end for
    Remove current root from V;
end while
```

In the algorithm, the function $Sim(a, b)$ returns either similarity (i.e., cosine similarity) or divergence (JS divergence) values between nodes a and b. The function $Sibling(root)$ returns a list of siblings of the current "node". To establish a "broader" relationship between two topics, the algorithm searches for the most similar topics in a broader range, i.e., 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. This step ensures that a topic will be listed under the most similar parent, thus, the time complexity of GSHL algorithm is $O(m^2 \cdot n^2)$, where $m = M_c$, $n = N$, and $n \gg M_c$. Since the value of m is much smaller than n, the algorithm is still more efficient compared to the one in [28]. In our experiment, the results generated using GSHL are slightly better than LSHL. The best precision is achieved using GSHL (see Section 6 for evaluation details). Accuracy of the statements for "related" relation is close to 90% in the evaluation domain. Precision of the statements generated using the LDA is notably higher than those generated using GSHL+pLSA and other algorithms.

5 Experiment Setting

For the experiment, we have prepared a dataset consists of about 4,600 abstracts of published papers in the area of the semantic Web. A total number of 77 topics of research were collected and then organised using GSHL and LSHL algorithms. The steps taken in the experiment are depicted in Figure 3 and are explained in detail in the following sub-sections.

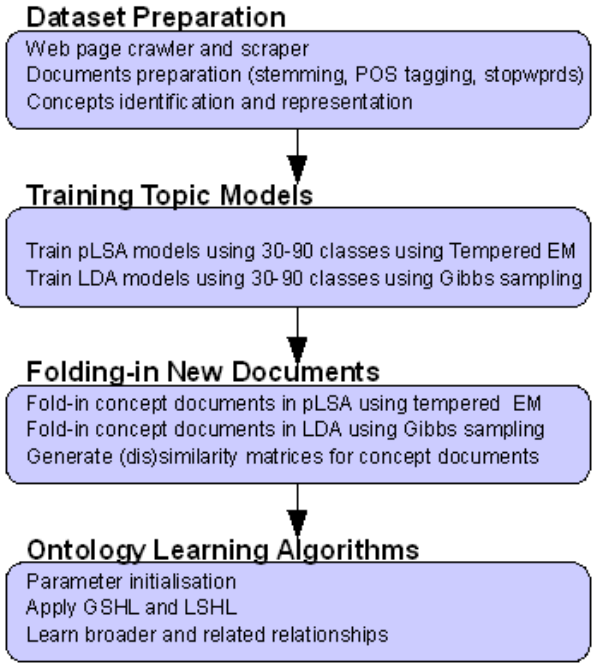


Fig. 3. The four steps taken for learning SKOS ontologies based on the pLSA and LDA models

5.1 Dataset Preparation

We have used a Web crawler to collect around 4,600 web pages containing publication abstracts related to the semantic Web research from the ACM digital library[11]. The Web pages are then processed with a scraper implemented based on the NekoHTML[12] parser. The abstracts, user specified keywords, and other information (such as bibliographic information) were extracted and stored. The collection of abstracts was used as our training corpus for probabilistic topic models. The corpus was processed by removing stopwords and applying Part-of-Speech (POS) tagging. The Stanford Log-linear POS Tagger[13] was used to POS-tag the corpus and only nouns, verbs, and adjectives were kept. The words were

11. http://portal.acm.org/
12. http://sourceforge.net/projects/nekohtml/
13. http://nlp.stanford.edu/software/tagger.shtml

also stemmed using the Porter's stemmer[14] packaged in the Apache's Lucene Java[15], resulting in approximately 6900 unique words in total.

We used a similar technique for extracting topics as in [17], i.e., by extracting keyword annotations of documents. Frequently appearing keywords were used as topics whose relations were to be learned. In our experiment, those keywords which occurred more than 10 times in the corpus are kept resulting in 77 topics. In making the selection we have used a synonym table to identify abbreviations and their original forms, for example, "SNA" and "social network analysis". A topic was then represented using documents annotated by that topic (i.e., documents were merged, words that occur only once were removed). The resulting document can be viewed as one that described that particular topic. After probabilistic topic models have been trained, documents representing topics were used as "new documents" and "folded-in" to the trained topic models to obtain their low dimensional representations in terms of probabilistic topic distributions.

5.2 Training Probabilistic Topic Models

The PennAspect Java API[16] was used to train and validate the pLSA model. The dataset was divided into two equal segments which are then transformed into the specific format required by the PennAspect. The dataset was then sent to the EM algorithm for training. The learned model parameters were $p(w|z)$, $p(d|z)$, and $p(z)$ which could be used to calculate similarity values between documents and words.

For training of the LDA, we used the Gibbs sampling algorithm provided by the LingPipe[17] which is a suite of Java libraries for linguistic analysis of human languages. The first 2000 Gibbs samples were dropped to make sure that the "burn-in" period was passed. The subsequent Gibbs samples that provided good estimates of the target distributions were saved for computation. The output of the training was a set of parameters $p(w|z)$ and $p(z|d)$. The two topics models were both trained using different number of classes (e.g., 30 to 90) for the purpose of comparison of the results.

5.3 Folding-in New Documents

The idea of "fold-in" new documents (also referred to as "pseudo-documents") into topic models was first discussed in the LSA papers [21], [22]. Since the new documents are represented using vectors in the lower dimensional space, the similarity measures between these new documents and the documents in the corpus can be calculated using their low dimensional representation vectors. The intuition is that even if the vocabularies used in the documents are different (word appearing in one document does not appear in another), the similarity between them could be high because they are both represented using vectors in the semantic space.

In the pLSA, a new document is folded in by a similar procedure as in the training phase using the EM algorithm. The difference is that in the folding-in phase the EM algorithm is conditioned on the learned parameter $p(w|z)$, (i.e., keeping the $p(w|z)$ fixed).

The "fold-in" process for the LDA can also be realised as in the training phase. For variational inference algorithms, like those described in the original LDA paper [12], one can condition on the distributions over words for the topics and run the variational algorithm to infer the distribution over topics for a new document. If one uses Gibbs sampling algorithm to learn the LDA model (as proposed here), the new document is folded-in by running the Gibbs sampling (with fixed topic-word probabilities), and sampling the assignments of words to topics in the new document.

We wrote two programs to fold documents into the learned topic models: the first uses the EM algorithm to fold documents into the learned pLSA models and the second one uses Gibbs sampling to fold-in documents into the learned LDA models. The output of the "folding-in" process is a set of document-topic distributions in the lower semantic space. These topics are subsequently organised into terminological ontologies. To improve the efficiency, three matrices containing pair-wise similarity and divergence values (Cosine, JS and KL divergence) between the folded documents are calculated before running the topic hierarchy learning algorithms (i.e., LSHL and GSHL).

5.4 Constructing Topic Hierarchies

The topic hierarchy construction procedure is a straightforward process once the topic models are learned and new documents are folded-in. In our experiment, we have used different combinations of parameters, for example, the number of classes (varied from 30 to 90) and the maximum number of subnodes as well as the use of similarity (Cosine) and divergence (JS divergence) measures were modified in different experiments in order to find their optimum settings. We have achieved a good balance between the recall and precision 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 $TH_n \in [0.3, 0.5]$. Overall precision of the topic hierarchies learned using the LDA model was higher than the one learned using the pLSA. With the LDA model the best result was achieved using 40 topics for training and for the pLSA model the best result was achieved using 60 topics. The GSHL algorithm performed better than LSHL in most cases. The best performance of the GSHL algorithm was achieved when using the Cosine similarity measure and the best performance of the LSHL algorithm was found when using the JS divergence measure.

14. http://tartarus.org/ martin/PorterStemmer/
15. http://lucene.apache.org/
16. http://www.cis.upenn.edu/datamining/software_dist/PennAspect/
17. http://alias-i.com/lingpipe/

Because of the large number of topics and topic relations, it is not possible to provide a visualisation of the whole ontology learned[18]. Figure 4 shows a snippet of the ontology centering on the topic "Ontology" with "broader" relationships. The ontology was learned with GSHL and Cosine similarity measure, and the LDA model with 40 classes.

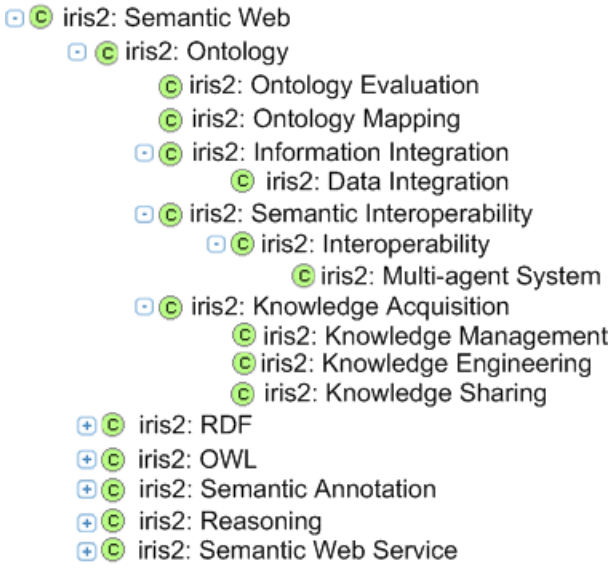


Fig. 4. A snippet of the ontology centering on the topic "Ontology" with "broader" relationships

6 Evaluation

The results generated by the experiment are evaluated by domain experts. Only if correctness of a statement of "broader" or "related" relationship is agreed by all domain experts, the statement is marked as correct. In the following sections we report the evaluation results, and discuss the effect of different parameter settings on the results (i.e., different number of classes for training topic models, and the maximum number of sub-topics).

6.1 Evaluation Methods

We use recall, precision and F1 measures [30], denoted as R, P, and F1 respectively, to evaluate the performance of our method and the quality of learned ontologies. These metrics are common in assessing performance of text retrieval. Ad-hoc measures such as precision at n and mean average precision which are widely used in the TREC[19] text retrieval conference are not used as we were not concerned with ranking of the results. The recall is defined as:

$$R = \frac{n_{tc}}{N_{cs}} \tag{9}$$

18. The learned ontologies and diagrams can be accessed at http://baggins.nottingham.edu.my/~eyx6ww/ol.php by clicking on "Experiment and Evaluation" and "Examples" tabs.

19. http://trec.nist.gov/

TABLE 1
Recall measures of ontology statements based on the pLSA and LDA

Settings VS NoOfClasses		30	40	50	60	70	80	90
pLSA	LSHL+COS	0.463	0.476	0.531	0.467	0.331	0.439	0.498
	LSHL+JS	**0.641**	0.491	0.568	0.59	0.34	0.428	0.502
	GSHL+COS	0.52	0.413	0.522	0.494	0.338	0.41	0.47
	GSHL+JS	0.59	0.417	0.524	0.592	0.425	0.371	0.481
LDA	LSHL+COS	0.713	0.706	0.691	**0.779**	0.652	0.713	0.671
	LSHL+JS	0.708	0.651	0.596	0.698	0.612	0.61	0.456
	GSHL+COS	0.712	0.746	0.73	0.766	0.744	0.7	0.61
	GSHL+JS	**0.776**	0.704	0.61	0.713	0.656	0.632	0.603

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

The precision is defined as:

$$P = \frac{n_{tc}}{N_{ls}} \tag{10}$$

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

Precision measure alone is not sufficient for assessing the performance of ontology learning algorithms. Low recall signifies that a large portion of topics are not selected and organised into ontologies by the algorithms, which is undesirable for ontology learning tasks. The F1 measure is defined as the harmonic mean of recall and precision:

$$F1 = \frac{2 * R * P}{R + P} \tag{11}$$

6.2 Comparison of Two Probabilistic Topic Models

The pLSA and LDA models are trained using different number of latent classes (i.e., 30, 40, 50, 60, 70, 80, and 90). For each trained model, topic hierarchies are constructed with different maximum number of sub-nodes (i.e., from 5 to 10) using the LSHL and GSHL algorithms. Table 1, 2 and 3 show the results of recall, precision and F1 under different algorithm settings. The first and second columns of three tables represent parameter settings, for example, "LDA and LSHL+COS" means that the result is generated using LSHL algorithm using the LDA model and Cosine as the similarity measure. Note that the numbers in the three tables are averaged over maximum number of sub-nodes.

Averaged recall measure (recall averaged over different classes and algorithm settings) of the experiment based on the pLSA is 47.6%, and the highest recall 64.%, is achieved using LSHL+JS with 30 classes for training the pLSA model. The averaged recall of the experiment based on the LDA is 67.7%, and the highest recall is 77.9% when LSHL+COS is used with 60 classes for the LDA training. In fact, as shown in Table 1, in most cases,

TABLE 2
Precision measures of ontology statements based on the pLSA and LDA

Settings VS NoOfClasses		30	40	50	60	70	80	90
pLSA	LSHL+COS	0.645	0.524	0.576	0.682	0.354	0.443	0.541
	LSHL+JS	0.676	0.593	0.626	**0.701**	0.386	0.521	0.555
	GSHL+COS	0.629	0.524	0.6	**0.719**	0.393	0.422	0.527
	GSHL+JS	0.65	0.514	0.586	**0.707**	0.499	0.47	0.545
LDA	LSHL+COS	0.713	0.745	0.691	0.779	0.678	0.732	0.777
	LSHL+JS	**0.841**	0.681	0.626	0.797	0.632	0.678	0.592
	GSHL+COS	0.712	**0.846**	0.732	0.771	0.785	0.737	0.722
	GSHL+JS	0.8	0.76	0.709	0.837	0.743	0.736	0.768

TABLE 3
F1 measures of ontology statements based on the pLSA and LDA

Settings VS NoOfClasses		30	40	50	60	70	80	90
pLSA	LSHL+COS	0.539	0.499	0.553	0.554	0.343	0.441	0.519
	LSHL+JS	**0.658**	0.537	0.595	0.64	0.361	0.47	0.527
	GSHL+COS	0.569	0.461	0.558	0.585	0.363	0.416	0.497
	GSHL+JS	0.619	0.46	0.553	0.644	0.459	0.415	0.511
LDA	LSHL+COS	0.713	0.726	0.691	0.779	0.665	0.722	0.72
	LSHL+JS	0.769	0.666	0.611	0.743	0.622	0.642	0.515
	GSHL+COS	0.712	**0.792**	0.731	0.768	0.764	0.718	0.661
	GSHL+JS	**0.798**	0.731	0.656	0.77	0.697	0.68	0.675

recall values of the experiment using the LDA model are above 70%.

The best precision of the experiment with the pLSA is 71.9% with settings GSHL+COS, while the best precision of the LDA is 84.6% with settings GSHL+COS. Averaged precision (precision averaged with different number of classes and algorithm settings) of statements generated using the LDA model is about 17.9% higher than those generated using the pLSA model (73.6% to 55.7%).

The precision values in Table 2 are averaged over different maximum number of sub-nodes (i.e., 5 to 10). Among all scenarios, the best precision measure of the pLSA model of 81.1% was achieved for the following parameter settings: 60 classes for training the pLSA model, the maximum number of sub-nodes was set to 9, and we used GSHL+COS. The lowest precision for this model, within the range of variability of parameters, was 31.5%. The highest precision with the LDA model was 87.5% when the number of classes for training was set to 30, the maximum number of sub-nodes was set to 8, and the LSHL+JS were used. The lowest precision was about 55%. Overall the performance of the LDA model was better than that of the pLSA model and was shown to be less sensitive to the changes in the number of training topics.

As shown in Table 3, the best F1 value of the generated statements using the pLSA is 65.8% and the bset of the LDA is 79.8%. This confirms the above conclusion about the overall performance of the LDA and pLSA models.

We attribute the superior performance of the LDA model to its capability of generalising to new documents. Although the pLSA model represents a significant step toward probabilistic modelling of text, it is incomplete in that it provides no probabilistic model at the level of documents [12]. The limitation leads to two problems: the number of parameters that need to be estimated grows linearly with the size of the corpus, which leads to overfitting; it is difficult to test generalisability of the model to new documents [14]. Our results are consistent with those reported by Blei *et al* [12].

TABLE 4
Correlation analysis between number of classes and recall and precision

Settings			LSHL+COS	LSHL+JS	GSHL+COS	GSHL+JS
NoOfClasses	pLSA	recall	-0.20681	-0.58069	-0.38474	-0.46031
		precision	-0.46031	-0.54183	-0.48101	-0.44058
	LDA	recall	-0.28897	**-0.75429**	-0.57577	**-0.75317**
		precision	0.29903	-0.62208	-0.2223	-0.19908

6.3 Effect of Number of Classes for Training Topic Models

Based on the results of our experiments we have performed a correlation analysis between the number of classes used for training topic models and the recall and precision so as to assess the "relatedness" of these two concerns. The correlation coefficient values between them are shown in Table 4. Except for two values -0.75429 and -0.75317 in the table (i.e., the row of the LDA recall), the correlation values between number of classes and recall and precision can be interpreted as weakly and moderately correlated. This coincides with the presumption made in [11], [12], [14] which states that the number of classes used for training probabilistic topic models is essentially an empirical parameter, and appropriate values can be found through repetitive experiments.

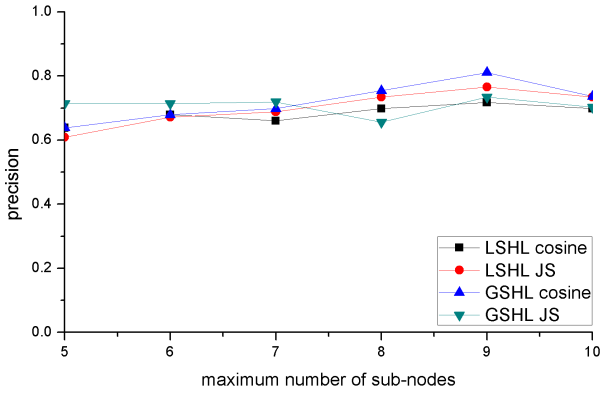
6.4 Effect of Maximum Number of Sub-Nodes

The maximum number of sub-nodes parameter was used to set the approximate number of child nodes of a parent node. Values from 5 to 10 were used in our experiments and the results are reported in Figure 5.

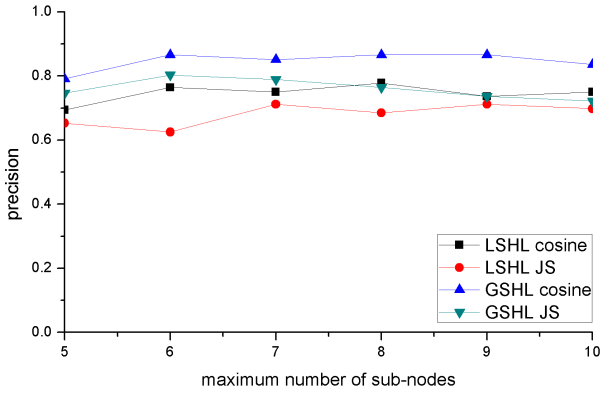
The pLSA model results are for 60 classes and the LDA model results are for 40 classes. It is clear that the precision measure evaluated for the LDA model is not very sensitive to the maximum number of sub-nodes. The precision measure evaluated for the pLSA model is also quite robust except when using the GSHL and Cosine similarity measures. In this case the statements generated using different maximum number of sub-nodes were found to be slightly different.

6.5 Effect of (dis)Similarity Measures

The experiment suggests that with the LDA model, a better overall result is achieved using GSHL+COS or LSHL+COS, although some of the extreme values may be produced using GSHL+JS or LSHL+JS, as reported in Section 6.2. On the other hand, with the pLSA model, better overall results were obtained when using the JS divergence measure.



(a) pLSA with 60 classes



(b) LDA with 40 classes

Fig. 5. The effect of maximum number of sub-nodes on precision measures of the pLSA and LDA

6.6 Comparing with Other Methods

In order to conduct a comparative study we have implemented the concept hierarchy construction algorithms proposed in [16], [28]. The algorithms were then applied to the same dataset. The algorithm in [16] establishes subsumption relation between any two concepts by exploiting their co-occurrence in a document collection. The algorithm is called "COO" in short.

The algorithm in [28] is based on training a sets of the LDA models using different classes, and deploying a principle called "conditional independence testing". The algorithm is referred here as CIT. CIT generates sets of topics by iterative training of the LDA models using different number of classes K. The topics trained using a small value of K are assumed to be more generic than those trained using large values of K. The algorithm first produces a number of topic sets $L_0, L_1, ...L_i, ..., L_n$ with increasing value of K, where L_i is the ith topic set. The task of the "conditional independent testing" is to detect whether a pair of topics A and B in the topic set L_{i+1} are independent given a topic C in the topic set L_i, which means the absolute value of the difference between conditional joint probability $P(A, B|C)$ and product of conditionals $P(A|C)$ and $P(B|C)$ is less than a threshold (see [28]). We trained 6 topic sets using 10, 20, 30, 40, 50, 60 classes respectively and applied "conditional independence testing" to construct topic hierarchy. We adopted two strategies for the tedious topic labeling process, i.e., labeling trained topics with extracted concept labels before, and after "conditional independence testing". The recall and precision measures are higher when labeling is done before the testing (recall: 53.9% vs 48.1%; precision: 67.2% vs 57.3%).

The results produced using the above two algorithms are compared to ours, i.e., GSHL combined with the LDA and pLSA. The recall and precision curves are shown in Figure 6 and 7 respectively. Note that the measures for the COO and CIT algorithms are not affected by the change of the number of classes. In Figure 6, the recall values of GSHL+LDA and GSHL+pLSA correspond to the best recall values generated by the GSHL algorithm using different number of classes for training topic models. In Figure 7, the GSHL+LDA and GSHL+pLSA curves show the best precision values generated by the GSHL algorithm using different number of classes.

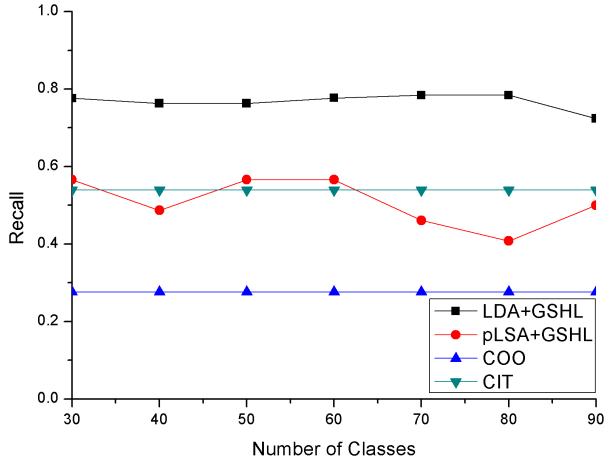


Fig. 6. Recall measures between COO, CIT, GSHL+LDA, and GSHL+pLSA

The weak performance of the COO algorithm: recall (27.6%) and precision (58.3%), is due to the fact that extracted concepts in the dataset do not co-occur frequently. The result reveals correctly the major limitation of the method. Furthermore, the algorithm tends to predict very specific concepts as subsumees of the most general concept, resulting in a shallow concept hierarchy (only 3 levels) and low precision. CIT reports a recall of 53.9% and precision of 67.2%. The GSHL algorithm using the LDA has the highest recall (value averaged over different number of classes is 76.7%) and precision (value averaged over different number of classes is around 81.2%) measures. The performance of the GSHL algorithm using the pLSA is comparable to CIT (50.7% averaged recall and 62.1% averaged precision). As a result of higher recall and precision measures, GSHL+LDA achieves the highest F1 (78.9%) measure among the four algorithms. F1 value for GSHL+pLSA is about 55.8%, slightly lower than CIT (59.8%) with COO having the lowest F1 value of 37.4%.

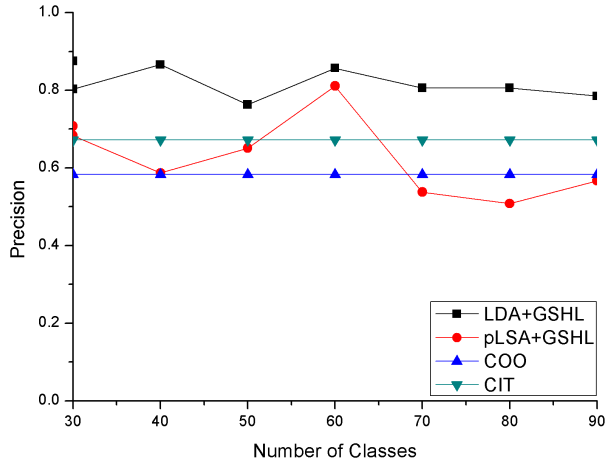


Fig. 7. Precision measures between COO, CIT, GSHL+LDA, and GSHL+pLSA

Experimental results discussed above are fully supportive of our new approach to the construction of topic hierarchy, as presented in Section 4.3.

6.7 Related Relations

The SKOS model formalises a rather ambiguous relationship called "related". Two topics are defined as "related" when the similarity condition in Definition 2 holds and the divergence difference condition does not. The intuition is that even if the "broader" relation can not be established between two topics, they may be similar to a great extent. Figure 8 shows the comparison between precision measures for "related" relations using the pLSA and LDA.

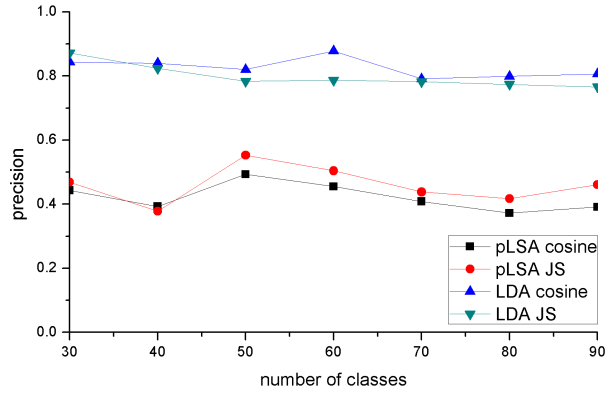


Fig. 8. Comparison of precision measures for "related" relations between the pLSA and LDA

The "related" relationship is only generated using the GSHL algorithm because the LSHL algorithm does not perform a "global" similarity search. It is notable that the precision difference between using the pLSA and LDA models in terms of "related" relationship is higher than the one in terms of "broader" relation. One possible explanation is that the algorithms based on the pLSA tend to select more incorrect similar topics than those based on the LDA.

7 Conclusion and Future Work

Topic and concept hierarchies are widely used in digital libraries for purposes of annotation, navigation, browsing and query suggestion. Research in ontology learning from text corpus has resulted in some promising methods for automatic construction of ontologies. Although ontologies learned using automated approaches can not be used for direct formal reasoning, they can be used in applications where a certain error rate is tolerable, such as information retrieval, browsing and navigation [2]. Furthermore, such ontologies can significantly reduce time and effort in the ontology engineering process, which often involves human experts.

The main contribution of the paper is development of a new method for learning terminological ontologies with respect to the SKOS model from text corpus, based on two probabilistic topic models of the pLSA [11] and LDA [12]. One aspect of our contribution is the use of topic models for the purpose of ontology learning which is a joint research area of the semantic Web and knowledge acquisition. The topic models have their roots in Information Retrieval and have been primarily used in applications such as text classification, collaborative filtering, and document modelling. Our method expands the applicability of topic models to automated ontology learning. Soundness and appropriateness of the method is assessed through experimental evaluation.

Another aspect of our contribution is the proposal of using Kullback-Leibler divergence as a probabilistic proxy for learning ontological relationships. In order to organise topics with two of the relationships defined in the SKOS model, namely, "broader" and "related", we propose "Information Theory Principle for Concept Relationship", which provides formal definitions for establishing "broader" relationship between two topics. The principle is underpinned by a quantitative measure of Kullback-Leibler divergence [27]. Augmentation of "broader" relationships with learning of "related" relationships improves on the existing works that mostly attempt to learn traditional concept hierarchies. Based on the defined principle, two algorithms (Local Similarity Hierarchy Learning and Global Similarity Hierarchy Learning) were developed to organise topics into terminological ontologies. An extensive evaluation using recall, precision, and F1 measures has been performed. The results show that ontologies learned using the LDA are superior to those learned using the pLSA in terms of recall, precision and F1 measures. The effect of different parameters in the algorithms, such as number of classes used to train the pLSA and LDA, maximum number of sub-nodes, and similarity function have also been discussed.

As far as we know, there is only one paper reporting the use of the LDA model for automated learning of concept hierarchies [28]. This work has been accomplished

in the domain of gene study. We have implemented the algorithms proposed in [28] as well as the one proposed in [16], and have performed evaluation of our methods against these algorithms on a common dataset. The results show that our algorithm based on the LDA model performs notably better than the other two in terms of recall, precision, and F1 measures.

The work presented here is a sequel of our previous work reporting the implementation of a semantic search engine for scientific publications. In the early version of the semantic search engine, called IRIS [31], a manually constructed domain ontology (topic hierarchy organised using the SKOS model) was used to help users browse and retrieve publications. The automatically learned ontology presented here will be deployed in the IRIS search engine, automating the process of knowledge acquisition (domain ontology construction).

Although our approach is intuitively domain-independent and has produced encouraging results with a computer science publications dataset, future work will involve testing the performance of the proposed algorithm on other datasets (such as medical research publications). It should also be possible to generalise our approach to learning simultaneously several sub-trees in the topic hierarchy.

ACKNOWLEDGMENTS

The authors would like to thank all the five anonymous reviewers for their valuable comments and suggestions. The authors also would like to thank Tom Griffiths for his help with the Gibbs sampling for document folding-in in LDA model, Andrew I. Schein for assisting with the pLSA training using PennAspect, and Ilias Zavitsanos for elaborating on the conditional independence testing algorithm for concept hierarchy construction.

REFERENCES

[1] T. Burners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, 2001.

[2] P. Cimiano, *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[3] S. Ponzetto and M. Strube, "Deriving a large scale taxonomy from wikipedia," in *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, Vancouver, B.C., July 2007, pp. 1440–1447.

[4] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *WWW '07: Proceedings of the 16th international conference on World Wide Web*. New York, NY, USA: ACM Press, 2007, pp. 697–706.

[5] H. Cunningham, "Information Extraction, Automatic," *Encyclopedia of Language and Linguistics, 2nd Edition*, 2005.

[6] P. Cimiano and J. Völker, "Text2onto," in *NLDB*, 2005, pp. 227–238.

[7] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, "Semantic annotation, indexing, and retrieval," *J. Web Semantics.*, vol. 2, no. 1, pp. 49–79, 2004.

[8] M. Fleischman and E. H. Hovy, "Fine grained classification of named entities." in *COLING*, 2002.

[9] F. M. Suchanek, G. Ifrim, and G. Weikum, "Combining linguistic and statistical analysis to extract relations from web documents," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 712–717.

[10] M. Pasca, "Finding instance names and alternative glosses on the web: Wordnet reloaded." in *CICLing*, ser. Lecture Notes in Computer Science, vol. 3406. Springer, 2005, pp. 280–292.

[11] T. Hofmann, "Probabilistic latent semantic analysis," in *UAI*, 1999, pp. 289–296.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[13] T. L. Griffiths and M. Steyvers, "Finding scientific topics." *Proc Natl Acad Sci USA*, vol. 101 Suppl 1, pp. 5228–5235, April 2004.

[14] M. Steyvers and T. Griffiths, "Probabilistic topic models," in *Latent Semantic Analysis: A Road to Meaning*, T. Landauer, D. Mcnamara, S. Dennis, and W. Kintsch, Eds. Laurence Erlbaum, 2005.

[15] R. Navigli and P. Velardi, "Learning domain ontologies from document warehouses and dedicated web sites." *Computational Linguistics*, vol. 30, no. 2, pp. 151–179, 2004.

[16] M. Sanderson and W. B. Croft, "Deriving concept hierarchies from text," in *SIGIR*, 1999, pp. 206–213.

[17] J. Diederich and W.-T. Balke, "The semantic growbag algorithm: Automatically deriving categorization systems," in *ECDL*, 2007, pp. 1–13.

[18] C. Biemann, "Ontology learning from text: A survey of methods," *LDV Forum*, vol. 20, no. 2, pp. 75–93, 2005.

[19] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *COLING*, 1992, pp. 539–545.

[20] Z. Harris, *Mathematical Structures of Language*. Wiley, 1968.

[21] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JASIS*, vol. 41, no. 6, pp. 391–407, 1990.

[22] M. W. Berry, S. T. Dumais, and G. W. O'Brien, "Using linear algebra for intelligent information retrieval," *SIAM Review*, vol. 37, pp. 573–595, 1995.

[23] J. Bilmes, "A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," ICSI-TR-97-021, University of Berkeley, California, Tech. Rep., 1997.

[24] G. D'Agostini, "Bayesian inference in processing experimental data: principles and basic applications," *Reports on Progress in Physics*, vol. 66, no. 9, pp. 1383–1419, 2003.

[25] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine Learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[26] T. Griffiths, "Gibbs sampling in the generative model of latent dirichlet allocation," Stanford University, Tech. Rep., 2002.

[27] D. J. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[28] E. Zavitsanos, G. Paliouras, G. A. Vouros, and S. Petridis, "Discovering subsumption hierarchies of ontology concepts from text corpora," in *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 402–408.

[29] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," in *Advances in Neural Information Processing Systems, Vol. 19 (NIPS*2005)*. Cambridge, MA: MIT Press, 2006, pp. 547–554.

[30] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[31] W. Wei, P. M. Barnaghi, and A. Bargiela, "The Anatomy and Design of A Semantic Search Engine," UNMC-CS-200712, School of Computer Science, University of Nottingham Malaysia Campus, Tech. Rep., 2007.

Wang Wei is a 3rd-year PhD student at the School of Computer Science, Faculty of Science, the University of Nottingham Malaysia Campus. He obtained his degree at the same university in 2006. His research interests include Information Retrieval, Semantic Web, Ranking, Ontology Learning, and Machine Learning.

Payam Barnaghi is a 3rd-year PhD student at the School of Computer Science, Faculty of Science, the University of Nottingham Malaysia Campus. He obtained his degree at the same university in 2006. His research interests include Information Retrieval, Semantic Web, Ranking, Ontology Learning, and Machine Learning.

Andrzej Bargiela is a 3rd-year PhD student at the School of Computer Science, Faculty of Science, the University of Nottingham Malaysia Campus. He obtained his degree at the same university in 2006. His research interests include Information Retrieval, Semantic Web, Ranking, Ontology Learning, and Machine Learning.