KB-LDA: Jointly Learning a Knowledge Base of Hierarchy, Relations, and Facts

Dana Movshovitz-Attias
Computer Science Department
Carnegie Mellon University
dma@cs.cmu.edu

William W. Cohen
Machine Learning Department
Carnegie Mellon University
wcohen@cs.cmu.edu

Abstract

Many existing knowledge bases (KBs), including Freebase, Yago, and NELL, rely on a fixed ontology, given as an input to the system, which defines the data to be cataloged in the KB, i.e., a hierarchy of categories and relations between them. The system then extracts facts that match the predefined ontology. We propose an unsupervised model that jointly learns a latent ontological structure of an input corpus, and identifies facts from the corpus that match the learned structure. Our approach combines mixed membership stochastic block models and topic models to infer a structure by jointly modeling text, a latent concept hierarchy, and latent semantic relationships among the entities mentioned in the text. As a case study, we apply the model to a corpus of Web documents from the software domain, and evaluate the accuracy of the various components of the learned ontology.

1 Introduction

Knowledge base (KB) construction methods can be broadly categorized along several dimensions. One dimension is ontology-guided construction, where the list of categories and relations that define the schema of the KB are explicit, versus open IE methods, where they are not. Another dimension is the type of relations and types included in the KB: some KBs, like WordNet, are hierarchical, in that they contain mainly concept types, supertypes and instances, while other KBs contain many types of relationships between concepts. Hierarchical knowledge can be learned by methods including distributional clustering (Pereira et al., 1993), as well as Hearst patterns (Hearst, 1992) and similar techniques (Snow et al., 2006). Reverb (Fader et al., 2011) and TextRunner (Yates et al., 2007) are open methods for learning multi-relation KBs. Finally, NELL (Carlson et al., 2010; Mitchell et al., 2015), FreeBase (Google, 2011) and Yago (Suchanek et al., 2007; Hoffart et al., 2013) are ontology-guided methods for extracting KBs containing both hierarchies and relations.

One advantage of ontology-guided methods is that the extracted knowledge is easier to reason with. An advantage of open IE methods is that ontologies may be incomplete, and are expensive to construct for a new domain. Ontology design involves assembling a set of categories, organized in a meaningful hierarchical structure, often providing seeds, i.e., representative examples for each category, and finally, defining inter-category relations. This process is often done manually (Carlson et al., 2010) leading to a rigid set of categories. Redesigning a new ontology for a specialized domain represents an additional challenge as it requires extensive knowledge of the domain.

In this paper, we propose an unsupervised model that learns a latent hierarchical structure of categories from an input corpus, learns latent semantic relations between categories, and also identifies facts from the corpus that match the learned structure. In other words, the model learns both the schema for a KB, and a set of facts that are related to that schema, thus combining the processes of KB population and ontology construction. The intent is to build systems that extract facts which can be interpreted relative to a meaningful ontology without requiring the effort of manual ontology construction.

The input to the learning method is a corpus of documents, plus two sets of resources extracted from the same corpus: a set of hypernym-hyponym pairs (e.g., "animal", "horse") extracted using Hearst patterns, and a set of subject-verb-object triples (e.g., "horse", "eats", "hay") extracted from parsed sentences. These resources are analogous to the output of open IE systems for

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 1449–1459,
Beijing, China, July 26-31, 2015. ©2015 Association for Computational Linguistics

hierarchies and relations, and as we demonstrate, our method can be used to highlight domain-specific data from open IE repositories.

Our approach combines mixed membership stochastic block models and topic models to infer a structure by jointly modeling text documents, and links that indicate hierarchy and relation among the entities mentioned in the text. Joint modeling allows information on topics of nouns (referred to as *instances*) and verbs (referred to as *relations*) to be shared between text documents and an ontological structure, resulting in a set of compelling topics. This model offers a complete solution for KB construction based on an input corpus, and we therefore name it *KB-LDA*.

We additionally propose a method for recovering meaningful names for concepts in the learned hierarchy. These are equivalent to category names in other KBs, however, following our method we extract from the data a set of potential alternative concepts describing each category, including probabilities for their strength of association.

To show the effectiveness of our method, we apply the model to a dataset of Web based documents from the software domain, and learn a software KB. This is an example of a specialized domain in which, to our knowledge, no broad-coverage ontology exists. We evaluate the model on the induced categories, relations, and facts, and we compare the proposed categories with an independent set of human-provided labels for documents. Finally, we use KB-LDA to retrieve domain-specific relations from an open IE resource. We provide the learned software KB as supplemental material.

2 KB-LDA

Modeling latent sets of entities from observed interactions among them is a well researched task, often encountered in social network analysis for the purpose of identifying specialized communities in the network. Mixed Membership Stochastic Blockmodels (Airoldi et al., 2009; Parkkinen et al., 2009) model entities as graph nodes with pairwise relations drawn from latent blocks with mixed membership. A related approach is taken by topic models such as LDA (Latent Dirichlet Allocation; (Blei et al., 2003)), which model documents as generated by a mixture of latent topics, and words in the documents as generated by topic-specific word distributions. The KB-LDA model combines the two approaches. It models links between tuples of two or three entities using stochastic block models, and these are additionally influenced by latent topic assignments of the entities in a document corpus.

$\boldsymbol{\pi_O}$ – multinomial over ontology topic pairs, with Dirichlet prior α_O
$\boldsymbol{\pi_R}$ – multinomial over relation topic tuples, with Dirichlet prior α_R
$\boldsymbol{\theta_d}$ – topic multinomial for document d, with Dirichlet prior α_D
$\boldsymbol{\sigma_k}$ – multinomial over instances for topic k, with Dirichlet prior γ_I
$\boldsymbol{\delta_{k'}}$ – multinomial over relations for topic k', with Dirichlet prior γ_R
$\text{CI}_i = \langle C_i, I_i \rangle$ – i-th ontological assignment pair
$\text{SVO}_j = \langle S_j, O_j, V_j \rangle$ – j-th relation assignment tuple
$\boldsymbol{z_i^{CI}} = \langle z_{C_i}, z_{I_i} \rangle$ – topic pair chosen for example $\langle C_i, I_i \rangle$
$\boldsymbol{z_j^{SVO}} = \langle z_{S_j}, z_{O_j}, z_{V_j} \rangle$ – topic tuple chosen for example $\langle S_j, O_j, V_j \rangle$
$\boldsymbol{z^D_{E_1}}, \boldsymbol{z^D_{E_2}}$ – topic chosen for instance entity E_1, or relation entity E_2, respectively, in a document
$\boldsymbol{n^I_{z,i}}$ – number of times instance i is observed under topic z (in either z^D, z^{CI} or z^{SVO})
$\boldsymbol{n^R_{z,r}}$ – number of times relation r is observed under topic z (in either z^D or z^{SVO})
$\boldsymbol{n^O_{\langle z_c, z_i \rangle}}$ – count of ontological pairs assigned the topic pair $\langle z_c, z_i \rangle$ (in z^{CI})
$\boldsymbol{n^R_{\langle z_s, z_o, z_v \rangle}}$ – count of relation tuples assigned the topic tuple $\langle z_s, z_o, z_v \rangle$ (in z^{SVO})

Table 1: KB-LDA notation.

In the KB-LDA model, shown as a plate diagram in Figure 1 with notation in Table 1, information is shared between three components, through common latent topics over noun and verb entities. The *Ontology* component (upper right) models hierarchical links between Concept-Instance (CI) entity pairs. The *Relations* component (left) models links between Subject-Verb-Object (SVO) entity triples, where the subject and object are nouns and the verb represents a relation between them. Finally, the *Documents* component (lower left) is a link-LDA model (Erosheva et al., 2004) of text documents containing a combination of noun and verb entity types. In this formulation, distributions over noun and verb entities that are related according to hierarchical or relational constraints, are linked with a text model via shared parameters.

In more detail, the *Documents* component provides the context in which noun and verb entities are being used in text. It is modeled as an extension of LDA, viewing documents as sets of "bags of words", where in this case, each bag contains either noun or verb entities. Each entity type has a topic-wise multinomial distribution over the set of entities in the vocabulary of that type.

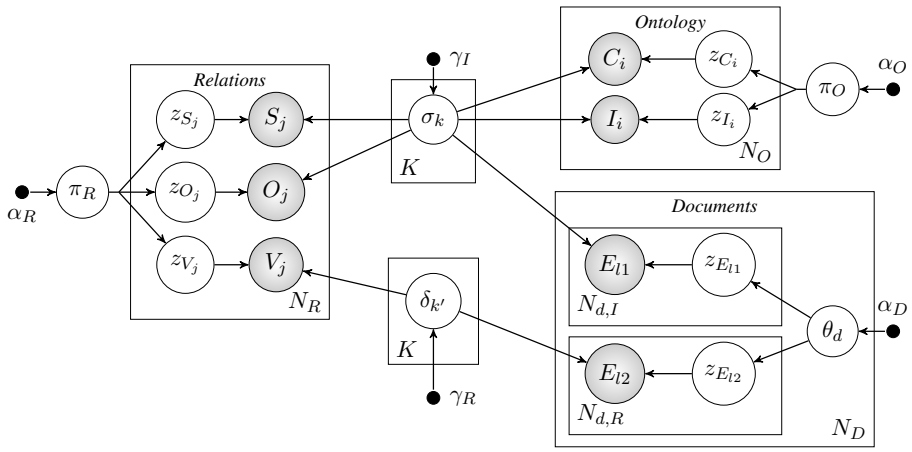

Figure 1: Plate Diagram of KB-LDA.

The *Ontology* component is a generative model representing hierarchal links between pairs of nouns. The examples for this component are extracted using a small collection of Hearst patterns indicating concept-instance or concept-concept links, including, 'X such as Y', and 'X including Y'. For example, the sentence "websites such as StackOverflow" indicates that Stackoverflow is a type of website, leading to the extracted noun pair ⟨websites, StackOverflow⟩. We refer to the examples extracted using these hierarchical patterns as *concept-instance pairs*, and to the individual entities as *instances*.

The pairs have an underlying block structure derived from a sparse block model (Parkkinen et al., 2009). They are generated by topic specific instance distributions conditioned on topic pair edges, which are defined by the multinomial π_O over the Cartesian product of the noun topic set with itself. The individual instances, therefore, have a mixed membership in topics. Note that we allow for a concept and instance to be drawn from different noun topics, defined by σ. For example, we may learn a topic highlighting concept tokens like 'websites', 'platforms', 'applications'. Another topic can highlight instances shared by these concepts, such as, 'stackoverflow', 'google', and 'facebook'. Finally, the observation that the former topic frequently contains concepts of instances from the latter topic, is encoded in the multinomial distribution π_O. From this we infer that the former topic should be placed higher in the induced hierarchy.

Similarly, the *Relations* component represents relational links between a noun subject, a verb and a noun object. The examples for this component are extracted from SVO patterns found in the document corpus, following Talukdar et al. (2012). An extracted example looks like: ⟨websites, execute, javascript⟩. Subject and object topics are drawn from the noun topics (σ), while the verb topics is drawn from the verb topics, defined by δ. The multinomial π_R encodes the interaction of noun and verb topics based on the extracted relational links, and it is defined over the Cartesian product of the noun topic set with itself and with

Let K be the number of target latent topics.

1. **Generate topics:** For topic $k \in 1, \ldots, K$, sample:
   - $\sigma_k \sim$ Dirichlet(γ_I), the per-topic instance distribution
   - $\delta_k \sim$ Dirichlet(γ_R), the per-topic relation distribution
2. **Generate ontology:** Sample $\pi_O \sim$ Dirichlet(α_O), the instance topic pair distribution.
   - For each concept-instance pair CI$_i$, $i \in 1, \ldots, N_O$:
     - Sample topic pair $z_i^{CI} \sim$ Multinomial(π_O)
     - Sample instances $C_i \sim$ Multinomial($\sigma_{z_{C_i}}$), $I_i \sim$ Multinomial($\sigma_{z_{I_i}}$), then CI$_i = \langle C_i, I_i \rangle$
3. **Generate relations:** Sample $\pi_R \sim$ Dirichlet(α_R), the relation topic tuple distribution.
   - For each tuple SVO$_j$, $j \in 1, \ldots, N_R$:
     - Sample topic tuple $z_j^{SVO} \sim$ Multinomial(π_R)
     - Sample instances, $S_j \sim$ Multinomial($\sigma_{z_{S_j}}$), $O_j \sim$ Multinomial($\sigma_{z_{O_j}}$), and sample a relation $V_j \sim$ Multinomial($\delta_{z_{V_j}}$)
4. **Generate documents:** For document $d \in 1, \ldots, D$:
   - Sample $\theta_d \sim$ Dirichlet(α_D), the topic mixing distribution for document d.
   - For every noun entity (E_{l1}) and verb entity (E_{l2}), $l1 \in 1, \ldots, N_{d,I}$, $l2 \in 1, \ldots, N_{d,R}$:
     - Sample topics $z_{E_{l1}}, z_{E_{l2}} \sim$ Multinomial(θ_d)
     - Sample entities $E_{l1} \sim$ Multinomial($\sigma_{z_{E_{l1}}}$) and $E_{l2} \sim$ Multinomial($\delta_{z_{E_{l2}}}$)

Table 2: KB-LDA generative process.

the verb topic set.

The generative process of KB-LDA is described in Table 2. Given the hyperparameters $(\alpha_O, \alpha_R, \alpha_D, \gamma_I, \gamma_R)$, the joint distribution over CI pairs, SVO tuples, documents, topics and topic assignments is given by

$$p(\pi_O, \pi_R, \sigma, \delta, \mathbf{CI}, \boldsymbol{z^{CI}}, \mathbf{SVO}, \boldsymbol{z^{SVO}}, \boldsymbol{\theta}, \mathbf{E}, \boldsymbol{z^D} | \alpha_O, \alpha_R, \alpha_D, \gamma_I, \gamma_R) =$$

$$\prod_{k=1}^{K} \text{Dir}(\sigma_k|\gamma_I) \times \prod_{k'=1}^{K} \text{Dir}(\delta_{k'}|\gamma_R) \times \quad (1)$$

$$\text{Dir}(\pi_O|\alpha_O) \prod_{i=1}^{N_O} \pi_O^{\langle z_{C_i}, z_{I_i} \rangle} \sigma_{z_{C_i}}^{C_i} \sigma_{z_{I_i}}^{I_i} \times$$

$$\text{Dir}(\pi_R|\alpha_R) \prod_{j=1}^{N_R} \pi_R^{\langle z_{S_j}, z_{O_j}, z_{V_j} \rangle} \sigma_{z_{S_j}}^{S_j} \sigma_{z_{O_j}}^{O_j} \delta_{z_{V_j}}^{V_j} \times$$

$$\prod_{d=1}^{N_D} \text{Dir}(\theta_d|\alpha_D) \prod_{l1=1}^{N_{d,I}} \theta_d^{z^D_{E_{l1}}} \sigma_{z^D_{E_{l1}}}^{E_{l1}} \prod_{l2=1}^{N_{d,R}} \theta_d^{z^D_{E_{l2}}} \delta_{z^D_{E_{l2}}}^{E_{l2}}$$

2.1 Inference in KB-LDA

Exact inference is intractable in the KB-LDA model. We use a collapsed Gibbs sampler (Griffiths and Steyvers, 2004) to perform approximate inference in order to query the topic distributions and assignments. It samples a latent topic pair for a CI pair in the corpus conditioned on the assignments to all other CI pairs, SVO tuples, and document entities, using the following expression, after collapsing π_O:

$$\hat{p}(z_i^{CI}|\text{CI}_i, z_{\neg i}^{CI}, z^{SVO}, z^D, \text{CI}_{\neg i}, \alpha_O, \gamma_I) \quad (2)$$

$$\propto \left(n_{z_i^{CI}}^{O\neg i} + \alpha_O \right) \times \frac{(n_{z_{C_i}, C_i}^{I\neg i} + \gamma_I)(n_{z_{I_i}, I_i}^{I\neg i} + \gamma_I)}{(\sum_C n_{z_{C_i}, C}^{I\neg i} + T_I\gamma_I)(\sum_I n_{z_{I_i}, I}^{I\neg i} + T_I\gamma_I)}$$

where counts of observations from the training set are noted by n (see Table 1), and T_I is the number of instance entities (size of noun vocabulary).

We similarly sample topics for each SVO tuple conditioned on the assignments to all other tuples, CI pairs and document entities, using the following expression, after collapsing π_R:

$$\hat{p}(z_j^{SVO}|\text{SVO}_j, z_{\neg j}^{SVO}, z^{CI}, z^D, \text{SVO}_{\neg j}, \alpha_R, \gamma_I, \gamma_R) \quad (3)$$

$$\propto \left(n_{z_j^{SVO}}^{R\neg j} + \alpha_R \right) \times \frac{(n_{z_{S_j}, S_j}^{I\neg j} + \gamma_I)(n_{z_{O_j}, O_j}^{I\neg j} + \gamma_I)(n_{z_{V_j}, V_j}^{R\neg j} + \gamma_R)}{(\sum_I n_{z_{S_i}, I}^{I\neg j} + T_I\gamma_I)(\sum_I n_{z_{O_i}, I}^{I\neg j} + T_I\gamma_I)(\sum_V n_{z_{V_j}, V}^{R\neg j} + T_R\gamma_R)}$$

We sample a latent topic for an entity mention in a document from the text corpus conditioned on the assignments to all other entity mentions after collapsing θ_d. The following expression shows topic sampling for a noun entity in a document:

$$\hat{p}(z_{E_{l1}}|E, \text{CI}, \text{SVO}, z^D, z^{CI}, z^{SVO}, \alpha_D, \gamma_I) \quad (4)$$

$$\propto (n_{d,z}^{\neg l1} + \alpha_D) \frac{n_{z_{E_{l1}}, E_{l1}}^{I\neg l1} + \gamma_I}{\sum_{E'_{l1}} n_{z_{E_{l1}}, E'_{l1}}^{I\neg l1} + T_I\gamma_I}$$

The per-topic multinomial parameters and topic distributions of CI pairs, SVO tuples and documents can be recovered with MLE estimates using their observation counts:

$$\hat{\sigma}_k^I = \frac{n_{k,I}^I + \gamma_I}{\sum_{I'} n_{k,I'}^I + T_I\gamma_I}, \hat{\delta}_k^R = \frac{n_{k,R}^R + \gamma_R}{\sum_{R'} n_{k,R'}^R + T_R\gamma_R}$$

$$\hat{\theta}_d^z = \frac{n_{z,d} + \alpha_D}{\sum_{z'} n_{z',d} + K\alpha_D}$$

$$\hat{\pi}_O^{\langle z_C, z_I \rangle} = \frac{n_{\langle z_C, z_I \rangle}^O + \alpha_O}{\sum_{z'_C, z'_I} n_{\langle z'_C, z'_I \rangle}^O + K^2 \cdot \alpha_O}$$

$$\hat{\pi}_R^{\langle z_S, z_O, z_V \rangle} = \frac{n_{\langle z_S, z_O, z_V \rangle}^R + \alpha_R}{\sum_{z'_S, z'_O, z'_V} n_{\langle z'_S, z'_O, z'_V \rangle}^R + K^3 \cdot \alpha_R}$$

Using the KB-LDA model we can describe the latent topic hierarchy underlying the input corpus. We consider the multinomial of the *Ontology* component, π_O, as an adjacency matrix describing a network where the nodes are instance topics and edges indicate a hypernym-to-hyponym relation. By extracting the maximum spanning tree over this adjacency matrix, we recover a hierarchy over the input data. We recover relations among instance topics by extracting from the Relations multinomial, π_R, the set of most probable tuples of a ⟨subject topic, verb topic, object topic⟩.

Our model is implemented using a fast, parallel approximation of collapsed Gibbs sampling, following Newman et al. (2009). In each sampling iteration, topics are sampled locally on a subset of the training examples. At the end of each iteration, data from worker threads is joined and model parameters are updated with complete information. In the next iteration, thread-local sampling starts with complete topic assignment information from the previous iteration. In each thread, the process can be viewed as a reordering of the input examples, where the examples sampled in that thread

are viewed first. It has been shown that parallel approaches considerably speed up iterative inference methods such as collapsed Gibbs sampling, resulting in test data log probabilities indistinguishable from those obtained using serial methods (Porteous et al., 2008; Newman et al., 2009). A parallel approach is especially important when training the KB-LDA model due to the large dimensions of the multinomials of the *Ontology* and *Relations* components (K^2 and K^3, respectively for a model with K topics). We train KB-LDA over 2000 iterations, more than what has traditionally been used for collapsed Gibbs samplers.

2.2 Data-driven discovery of topic concepts

The KB-LDA model described above clusters noun entities into sets of instance topics, and recovers a latent hierarchical structure among these topics. Each instance topic can be described by a multinomial distribution of the underlying nouns. It is often more intuitive, however, to refer to a topic containing a set of high probability nouns by a name, or category, just as traditional ontologies describe hierarchies over categories.

Our model is trained over nouns that originate from concept-instance example pairs (used to train the *Ontology* component). We describe a method for selecting a category name for a topic, based on concepts that best represent high probability nouns of the topic in the concept-instance examples.

We calculate the probability that a concept noun c describes the set of instances I that have been assigned the topic z using

$$\begin{aligned} p(c, z|I) &\propto p(I|c, z) * p(c, z) \\ &= p(I|c, z) * p(z|c) * p(c) \end{aligned} \quad (5)$$

Let $rep(c, z) = \sum_{i:C_i=c} n^I_{z,I_i}$ describe how well concept c represents topic z according to the assignments of instances with concept c to the topic. Then,

$$p(z|c) = \frac{rep(c, z)}{\sum_{z'} rep(c, z')} \quad (6)$$

The concept prior, $p(c)$, is based on the relative weight of instances with concept c in the concept-instance example set, and is an indicator of the generality of a concept:

$$p(c) = \frac{\sum_{i:C_i=c} w_{c,I_i}}{\sum_{c'} \sum_{i:C_i=c'} w_{c',I_i}} \quad (7)$$

where $w_{C,I}$ is the number of occurrences of concept-instance pair $\langle C, I \rangle$ in the corpus.

Finally, $p(I|c, z)$ measures how specific are the topic instances to the concept c,

$$p(I|c, z) = \frac{\sum_{i:I_i \in I, C_i=c} w_{c,I_i}}{\sum_{i:C_i=c} w_{c,I_i}} \Big/ Z \quad (8)$$

where I is the set of training instances assigned with topic z, and Z is a normalizer over all concepts and topics.

Following this method we extract concepts that have a high probability $p(c, z|I)$ with respect to a topic z. These can be thought of as equivalent to the single, fixed, category name provided by traditional KB ontologies; however, here we extract *from the data* a set of potential alternative noun phrases describing each topic, including a probability for the strength of this association.

3 Experimental Evaluation

We evaluate the KB-LDA model on a corpus of 5.5M documents from the software domain; thereby we are using the model to construct a software domain knowledge base. Our evaluation explores the following questions:

- Can KB-LDA learn categories, relations, a hierarchy and topic concepts with high precision?
- How well do KB-LDA topics correspond with human-provided document labels?
- Is KB-LDA useful in extracting facts from existing open IE resources?

3.1 Data

We use data from the Q&A website StackOverflow[1] where users ask and answer technical questions about software development, tools, algorithms, etc'. We extracted 562K concept-instance example pairs from the data, and kept the 17K examples appearing at least twice. Noun phrases in these examples make up our *Instance Dictionary*. Out of 6.8M SVO examples found in the data we keep 37K in which the subject and object are in the Instance Dictionary, and the example appears at least twice in the corpus. The verbs in these SVOs make up our *Relation Dictionary*. Finally, we consider as documents the 5.5M questions from StackOverflow with all their answers.

3.2 Evaluating the learned KB precision

In this section we evaluate the direct output of a model trained with 50 topics: the extracted in-

[1]Data source: https://archive.org/details/stackexchange

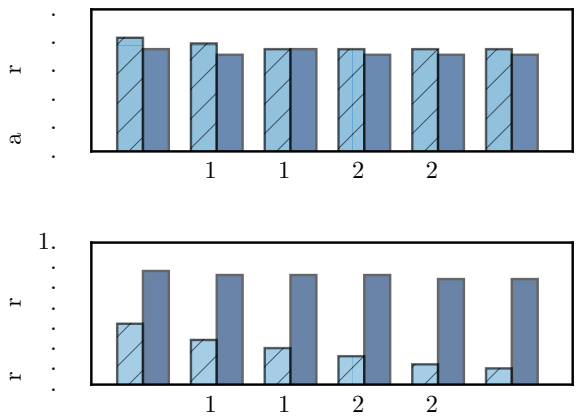

Figure 2: Average *Match* (top) and *Group* (bottom) precision of top tokens of 50 topics learned with KB-LDA, according to expert (dark blue) and non-expert (light blue, stripes) labeling.

stance topics, topic hierarchy, relations among topics and extracted topic concepts. In each of the experiments below, we extract facts based on one of the learned components and evaluate each fact based on annotations from human judges: two experts and three non-expert users, collected using Mechanical Turk, that were pre-tested on a basic familiarity with concepts from the software domain, such as *programming languages, version control systems*, and *databases*.

3.2.1 Precision of Instance Topics

We measure the coherence of instance topics using an approach called *word intrusion* (Chang et al., 2009). We extract the top 30 instance tokens of a topic ranked by the instance topic multinomial σ. We present to workers tokens 1-5,6-10,...,26-30, where each 5 tokens are randomly ordered and augmented with an extra token that is ranked low for the topic, (the intruder). We ask workers to select all tokens that do not belong in the group (and at least one). We define the topic *Match Precision* as the fraction of questions for which the reviewer identified the correct intruder (out of 6 questions per topic), and the topic *Group Precision* as the fraction of correct tokens (those not selected as not belonging in the group). Thus *Match Precision* measures how well labelers understand the topic, and *Group Precision* measures what fraction of words appeared relevant to the topic.

Figure 2 shows the average Match and Group precision over the top tokens of all 50 topics learned with the model, as evaluated by expert and non-expert workers. Both groups find the intruder token in over 75% of questions. In the more subtle task of validating each topic token (Group precision) we see a greater variance among the two labeler groups. This highlights the difficulty of evaluating domain specific facts with non-expert users. Table 3 displays the top 20 instance topics learned with KB-LDA, ranked by expert Group precision.

3.2.2 Precision of Topic Concepts

We assess the precision of the top 5 concept names proposed for instance topics, following the method presented in Section 2.2. Top concepts for a subset of topics are shown in Table 3. For each topic, we present to the user a hypernym-hyponym pattern of the topic based on the top concepts and top instances of the topic. As an example, if the top 5 instances of a topic are *ie, firefox, chrome, buttons, safari* and the top 5 concepts for this topic are *web browsers, web browser, browser, ie, chrome*, the pattern presented to workers is

- [ie, firefox, chrome, buttons, safari] **is a** [web browsers, web browser, browser, ie, chrome]

Workers were asked to match at least 3 instances to a proposed concept name. In addition, the same assessment was applied for each topic using randomly sampled concepts. We present in Table 4 the number and precision of patterns based on extracted concepts (Concepts) and random concepts (Random), that were labeled by 1, 2 or 3 workers, as well as the average results among experts. We achieve nearly 90% precision according to expert labeling, however we do not observe large agreement among non-expert labelers.

3.2.3 Precision of Relations

To assess the precision of the relations learned in the KB-LDA model, we extract the top 100 relations learned according to their probability in the relation multinomial π_R. Relation patterns were presented to workers as sets of the top subject-verb-object tokens of the respective topics in the relation. An example relation is

- Subject words: [user, users, people, customer, client]
- Verb words: [clicks, selects, submits, click, hits]
- Object words: [function, method, class, object, query]

and workers are asked to state whether the pattern indicates a valid relation or not, by checking whether a reasonable number of combinations of subject-verb-object triples extracted from each of the relation groups can produce valid relations.

Top 2 Topic Concepts	Top 10 Topic Tokens
table, key	table, query, database, sql, column, data, tables, mysql, index, columns
properties, css	image, code, images, problem, point, color, data, size, screen, points
credentials, user information	name, images, id, number, text, password, address, strings, files, string
page, content	page, html, code, file, image, javascript, browser, http, jquery, js
orm tools, orm tool	tomcat, hibernate, server, boost, apache, spring, mongodb, framework, nhibernate, png
clients, apps	app, application, http, android, device, phone, code, api, iphone, google
applications, systems	devices, systems, applications, services, platforms, tools, sites, apps, system, service
systems, platforms	google, windows, linux, facebook, git, ant, database, gmail, android, so
limits, limit	memory, time, thread, code, threads, process, file, program, data, object
data, table	query, table, data, list, example, number, results, search, database, rows
type, value	code, function, value, type, pointer, array, memory, compiler, example, string
table, request	data, information, types, properties, details, fields, values, content, resources, attributes
dependencies, jar file	libraries, library, framework, frameworks, formats, format, database, databases, tools, server
type, object	value, focus, place, property, method, reference, interface, effect, pointer, data
kinds, code	languages, language, features, objects, functions, methods, code, operations, structures, types
element, elements	button, form, link, item, *file*, mouse, image, value, option, row
javascript libraries, javascript framework	jquery, mysql, http, json, xml, library, html, sqlite, *asp*, php
process, operating system	server, client, connection, *data*, http, socket, message, request, port, service
folder, files	file, files, directory, folder, path, *code*, name, resources, project, folders
value, array	array, list, value, values, number, string, code, elements, *loop*, object

Table 3: Top 20 instance topics learned with KB-LDA. For each topic we show the top 2 concepts recovered for the topic, and top 10 tokens. In *italics* are words marked as out-of-topic by expert labelers.

Workers	Concepts		Relations		Subsumptions	
	KB-LDA (p)	Random (p)	KB-LDA (p)	Random (p)	KB-LDA (p)	Random (p)
1	48 (0.96)	6 (0.12)	90 (0.9)	69 (0.69)	31 (0.63)	28 (0.57)
2	42 (0.84)	0 (0.0)	63 (0.63)	22 (0.22)	16 (0.33)	9 (0.18)
3	26 (0.52)	0 (0.0)	15 (0.15)	4 (0.05)	3 (0.06)	4 (0.08)
Experts	44 (0.88)	0 (0.0)	70 (0.7)	13 (0.13)	25 (0.51)	4 (0.08)

Table 4: Precision of topic concepts, relations, and subsumptions. For items extracted from the model (KB-LDA), and randomly (Random), we show the number of items marked as correct, and precision in parentheses (p), as labeled by 1, 2, or 3 non-expert workers, and the average precision by experts.

We present in Table 4 the number and precision of patterns based on the top 100 relations (Relations) and 100 random relations (Random), that were labeled by 1, 2 or 3 workers, and the average results among experts. We achieve 80% precision according to experts, and only 18% on random relations. We observe similar agreement among expert and non-expert workers as in the concept evaluation experiment, however we note that random relations prove more confusing for non-experts and more of them are (falsely) labeled as correct.

3.2.4 Precision of Hierarchy

We assess the precision of subsumption relations making up the ontology hierarchy. These are extracted using the maximum spanning tree over the graph represented by the *Ontology* component, π_O (see Section 2.1 for details), resulting in 49 subsumption relations. We compare their quality to that of 49 randomly sampled subsumption relations. Subsumptions are presented to the worker using *is a* patterns, similar to the ones described above for concept evaluation, however in this case, the concept tokens are the top tokens of the hypernym topic. An example subsumption relation is

- [java, python, javascript, lists, ruby] **is a** [languages, language, features, objects, functions]

The results shown in Table 4 indicate a low precision among the extracted subsumption relations. This might be explained by the fact that at the final training iteration (2K) of the model, the perplexity of the *Ontology* component was still improving, while the perplexity of the other model components seemed closer to convergence. It is possible that the low precision observed here indicates that more training iterations are needed to achieve an accurate ontology using KB-LDA.

Topic	string, character, characters, text, line
Tags	regex, string, python, php, ruby
Topic	element, div, css, elements, http
Tags	css, html, jquery, html5, javascript
Topic	table, query, database, sql, column
Tags	sql, mysql, database, performance, php
Topic	jquery, mysql, http, json, xml
Tags	jquery, json, javascript, ruby, string

Table 5: Top tags associated with sample topics.

3.3 Overlap of KB-LDA topics with human-provided labels

We evaluated how well topics from KB-LDA correspond to document labels provided by humans, over a randomly sampled set of 40K documents from our corpus. In StackOverflow, questions (which we consider as documents) can be labeled with predefined tags. Here, we estimate the overlap with the most frequently used tags. First, for topic k, we aggregate tags from documents where $k = \text{argmax}_{k'} \theta_d^{k'}$, where θ_d is the document topic distribution. Table 5 shows examples of the top tags associated with sample topics, indicating a good correlation between top topic words and the underlying concepts.

Next, for each tested document $d \in D$, let W_d be the top 30 words of the most probable topic in θ_d, and T_d the set of human provided document tags. We consider the following metrics:

$$\text{Docs-Overlap} = \frac{\sum_d^D \mathbb{1}_{\{\exists t \in T_d : t \in W_d\}}}{|D|}$$

measures the ratio of documents for which at least one tag overlaps with a top topic word. The average ratio of overlapping tags per document is

$$\text{Tag-Overlap} = \frac{1}{|D|} \sum_d^D \frac{|t : t \in T_d \wedge t \in W_d|}{|T_d|}$$

As a baseline, we measure similar overlap metrics using the 30 most frequent instance tokens in the document corpus. The results in Table 6 indicate an overlap of nearly half of the 20, 50, 100, and 500 most frequent tags with top topic tokens – significantly higher than the overlap with frequent token. Our evaluation is based on the subset of tags found in the instance dictionary of KB-LDA.

Top Tags	Found in Dictionary	KB-LDA Docs	KB-LDA Tag	Frequent Tokens Docs	Frequent Tokens Tag
20	14	0.45	0.42	0.21	0.16
50	36	0.48	0.42	0.20	0.14
100	72	0.45	0.38	0.20	0.13
500	322	0.44	0.33	0.18	0.10

Table 6: Docs and Tag overlap of human-provided tags with KB-LDA topics, and frequent tokens.

Top 10 ranked triples: ⟨server, not found, error⟩, ⟨user, can access, file⟩, ⟨method, not found, error⟩, ⟨user, can change, password⟩, ⟨page, not found, error⟩, ⟨user, can upload, videos⟩, ⟨compiler, will generate, error⟩, ⟨users, can upload, files⟩, ⟨users, can upload, files⟩, ⟨object, not found, error⟩
Bottom 10 ranked triples: ⟨france, will visit, germany⟩, ⟨utilities, may include, heat⟩, ⟨iran, has had, russia⟩, ⟨russia, can stop, germany⟩, ⟨macs, do not support, windows media player⟩, ⟨cell phones, do not make, phone calls⟩, ⟨houses, have made, equipment⟩, ⟨guests, will find, restaurants⟩, ⟨guests, can request, bbq⟩, ⟨inspectors, do not make, appointments⟩

Table 7: Top and bottom ReVerb software triples ranked with KB-LDA.

3.4 Extracting facts from an open IE resource

We use KB-LDA to extract domain specific triples from an existing open IE KB, the 15M relations extracted using ReVerb (Fader et al., 2011) from ClueWeb09. By extracting the relations in which the subject, verb and object noun phrases are included in the KB-LDA dictionary, we are left with under 5K triples, indicating the low coverage of software related triples using open domain extraction, in comparison with the 37K triples extracted from StackOverflow and given as an input to KB-LDA.

Due to word polysemy, many of the 5K extracted triples are themselves not specific to the domain. This suggests a hybrid approach in which KB-LDA is used to rank open IE triples for relevance to a domain. We ranked the 5K open triples by the probability of the triple given a trained KB-LDA model: $p(s,v,o) = \sum_{k_s}^K \sum_{k_v}^K \sum_{k_o}^K \pi_R^{\langle k_s,k_v,k_o\rangle} \sigma_{k_s}^s \sigma_{k_o}^o \delta_{k_v}^v$. Table 7 shows the top and bottom 10 triples according to this ranking, which suggests that the triples ranked higher by KB-LDA are more relevant to the software domain.

We compare the ranking based on KB-LDA to

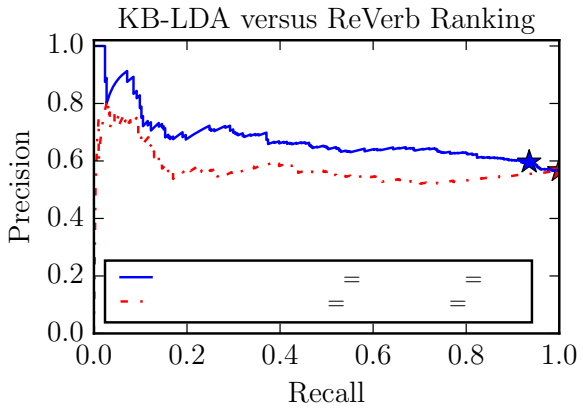

Figure 3: Precision-recall curves of rankers of open IE triples by software relevance, based on KB-LDA probabilities (blue), and ReVerb confidence (red). A star is pointing the highest F1.

a ranking using a confidence score for the triple as assigned by ReVerb. We manually labeled 500 of the triples according to their relevance to the software domain, and measured the precision and recall of the two rankings at any cutoff threshold. Figure 3 shows precision-recall curves for the two rankings, demonstrating that the ranking using probabilities based on KB-LDA leads to a more accurate detection of domain-relevant triples (with AUC of 0.67 for KB-LDA versus 0.57 for ReVerb).

4 Related Work

KB-LDA is an extension to LDA and link-LDA (Blei et al., 2003; Erosheva et al., 2004), modeling documents as a mixed membership over entity types with additional annotated metadata, such as links (Nallapati et al., 2008; Chang and Blei, 2009). It is a generalization of Block-LDA (Balasubramanyan and Cohen, 2011), however, KB-LDA models two link components, and the input links have a meaningful semantic correspondence to a KB structure (hierarchical and relational). In a related approach, Dalvi et al. (2012) cluster web table concepts to non-probabilistically create hierarchies with assigned concept names.

Our work is related to latent tensor representation of KBs, aimed at enhancing the ontological structure of existing KBs with relational data in the form of tensor structures. Nickel et al. (2012) factorized the ontology of Yago 2 for relational learning. A related approach was using Neural Tensor Networks to extract new facts from an existing KB (Chen et al., 2013; Socher et al., 2013). In contrast, in KB-LDA, relational data is learned jointly with the model through the *Relations* component.

Statistical language models have recently been adapted for modeling software code and text documents. Most tasks focused on enhancing the software development workflow with code and comment completion (Hindle et al., 2012; Movshovitz-Attias and Cohen, 2013), learning coding conventions (Allamanis et al., 2014), and extracting actionable tasks from software documentation (Treude et al., 2014). In related work, specific semantic relations, coordinate relations, have been extracted for a restricted class of software entities, ones that refer to Java classes (Movshovitz-Attias and Cohen, 2015). KB-LDA extends previous work by reasoning over a large variety of semantic relations among general software entities, as found in a document corpus.

5 Conclusions

We presented a model that jointly learns a latent ontological structure of a corpus augmented by relations, and identifies facts matching the learned structure. The quality of the produced structure was demonstrated through a series of real-world evaluations employing human judges, which measured the semantic coherence of instance topics, relations, topic concepts, and hierarchy. We further validated the semantic meaning of topic concepts, by their correspondence to an independent source of human-provided document tags. The experimental evaluation validates the usefulness of the proposed model for corpus exploration.

The results highlight the benefits of generalizing pattern-based facts (hypernym-hyponym pairs and subject-verb-object tuples), using text documents in a topic model framework. This modular approach offers opportunities to further improve an induced KB structure by posing additional constraints on corpus entities in the form of additional components to the model.

Acknowledgments

The authors wish to thank Premkumar Devanbu and Kathryn Rivard Mazaitis for helpful discussions, and the anonymous reviewers for their insightful comments. This work was funded by NSF under grant CCF-1414030.

References

Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. 2009. Mixed membership stochastic blockmodels. In *Advances in Neural Information Processing Systems*, pages 33–40.

Miltiadis Allamanis, Earl T Barr, and Charles Sutton. 2014. Learning natural coding conventions. *arXiv preprint arXiv:1402.4182*.

Ramnath Balasubramanyan and William W. Cohen. 2011. Block-lda: Jointly modeling entity-annotated text and entity-entity links. In *Proceedings of the 7th SIAM International Conference on Data Mining*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr, and T.M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*.

Jonathan Chang and David M Blei. 2009. Relational topic models for document networks. In *International Conference on Artificial Intelligence and Statistics*, pages 81–88.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-graber, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296.

Danqi Chen, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2013. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. *arXiv preprint arXiv:1301.3618*.

Bhavana Bharat Dalvi, William W Cohen, and Jamie Callan. 2012. Websets: Extracting sets of entities from the web using unsupervised information extraction. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 243–252. ACM.

Elena Erosheva, Stephen Fienberg, and John Lafferty. 2004. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

Google. 2011. Freebase data dumps. http://download.freebase.com/datadumps/.

Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proc. of the National Academy of Sciences of the United States of America*.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*. ACL.

Abram Hindle, Earl T Barr, Zhendong Su, Mark Gabel, and Premkumar Devanbu. 2012. On the naturalness of software. In *Software Engineering (ICSE), 2012 34th International Conference on*, pages 837–847. IEEE.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61.

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.

Dana Movshovitz-Attias and William W. Cohen. 2013. Natural language models for predicting programming comments. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Dana Movshovitz-Attias and William W. Cohen. 2015. Grounded discovery of coordinate term relationships between software entities. *arXiv preprint arXiv:1505.00277*.

Ramesh M Nallapati, Amr Ahmed, Eric P Xing, and William W Cohen. 2008. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 542–550. ACM.

David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2009. Distributed algorithms for topic models. *The Journal of Machine Learning Research*, 10:1801–1828.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280. ACM.

Juuso Parkkinen, Janne Sinkkonen, Adam Gyenge, and Samuel Kaski. 2009. A block model suitable for sparse graphs. In *Proceedings of the 7th International Workshop on Mining and Learning with Graphs (MLG 2009), Leuven*.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of english words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 183–190. Association for Computational Linguistics.

Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2008. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577. ACM.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 801–808. Association for Computational Linguistics.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. 2012. Acquiring temporal constraints between relations. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 992–1001. ACM.

Christoph Treude, M Robillard, and Barthélémy Dagenais. 2014. Extracting development tasks to navigate software documentation. *IEEE Transactions on Software Engineering*.

Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 25–26. Association for Computational Linguistics.