A Latent Dirichlet Allocation method for Selectional Preferences

Alan Ritter, Mausam and Oren Etzioni
Department of Computer Science and Engineering
Box 352350, University of Washington, Seattle, WA 98195, USA
{*aritter,mausam,etzioni*}*@cs.washington.edu*

Abstract

The computation of *selectional preferences*, the admissible argument values for a relation, is a well-known NLP task with broad applicability. We present LDA-SP, which utilizes LinkLDA (Erosheva et al., 2004) to model selectional preferences. By simultaneously inferring latent topics and topic distributions over relations, LDA-SP combines the benefits of previous approaches: like traditional class-based approaches, it produces human-interpretable classes describing each relation's preferences, but it is competitive with non-class-based methods in predictive power.

We compare LDA-SP to several state-of-the-art methods achieving an 85% increase in recall at 0.9 precision over mutual information (Erk, 2007). We also evaluate LDA-SP's effectiveness at filtering improper applications of inference rules, where we show substantial improvement over Pantel *et al.*'s system (Pantel et al., 2007).

1 Introduction

Selectional Preferences encode the set of admissible argument values for a relation. For example, locations are likely to appear in the second argument of the relation *X is headquartered in Y* and companies or organizations in the first. A large, high-quality database of preferences has the potential to improve the performance of a wide range of NLP tasks including semantic role labeling (Gildea and Jurafsky, 2002), pronoun resolution (Bergsma et al., 2008), textual inference (Pantel et al., 2007), word-sense disambiguation (Resnik, 1997), and many more. Therefore, much attention has been focused on automatically computing them based on a corpus of relation instances.

Resnik (1996) presented the earliest work in this area, describing an information-theoretic approach that inferred selectional preferences based on the WordNet hypernym hierarchy. Recent work (Erk, 2007; Bergsma et al., 2008) has moved away from generalization to known classes, instead utilizing distributional similarity between nouns to generalize beyond observed relation-argument pairs. This avoids problems like WordNet's poor coverage of proper nouns and is shown to improve performance. These methods, however, no longer produce the generalized class for an argument.

In this paper we describe a novel approach to computing selectional preferences by making use of unsupervised topic models. Our approach is able to combine benefits of both kinds of methods: it retains the generalization and human-interpretability of class-based approaches and is also competitive with the direct methods on predictive tasks.

Unsupervised topic models, such as latent Dirichlet allocation (LDA) (Blei et al., 2003) and its variants are characterized by a set of hidden topics, which represent the underlying semantic structure of a document collection. For our problem these topics offer an intuitive interpretation – they represent the (latent) set of classes that store the preferences for the different relations. Thus, topic models are a natural fit for modeling our relation data.

In particular, our system, called LDA-SP, uses LinkLDA (Erosheva et al., 2004), an extension of LDA that simultaneously models *two* sets of distributions for each topic. These two sets represent the two arguments for the relations. Thus, LDA-SP is able to capture information about the pairs of topics that commonly co-occur. This information is very helpful in guiding inference.

We run LDA-SP to compute preferences on a massive dataset of binary relations $r(a_1, a_2)$ ex-

tracted from the Web by TEXTRUNNER (Banko and Etzioni, 2008). Our experiments demonstrate that LDA-SP significantly outperforms state of the art approaches obtaining an 85% increase in recall at precision 0.9 on the standard pseudo-disambiguation task.

Additionally, because LDA-SP is based on a formal probabilistic model, it has the advantage that it can naturally be applied in many scenarios. For example, we can obtain a better understanding of similar relations (Table 1), filter out incorrect inferences based on querying our model (Section 4.3), as well as produce a repository of class-based preferences with a little manual effort as demonstrated in Section 4.4. In all these cases we obtain high quality results, for example, massively outperforming Pantel et al.'s approach in the textual inference task.[1]

2 Previous Work

Previous work on selectional preferences can be broken into four categories: class-based approaches (Resnik, 1996; Li and Abe, 1998; Clark and Weir, 2002; Pantel et al., 2007), similarity based approaches (Dagan et al., 1999; Erk, 2007), discriminative (Bergsma et al., 2008), and generative probabilistic models (Rooth et al., 1999).

Class-based approaches, first proposed by Resnik (1996), are the most studied of the four. They make use of a pre-defined set of classes, either manually produced (e.g. WordNet), or automatically generated (Pantel, 2003). For each relation, some measure of the overlap between the classes and observed arguments is used to identify those that best describe the arguments. These techniques produce a human-interpretable output, but often suffer in quality due to an incoherent taxonomy, inability to map arguments to a class (poor lexical coverage), and word sense ambiguity.

Because of these limitations researchers have investigated non-class based approaches, which attempt to directly classify a given noun-phrase as plausible/implausible for a relation. Of these, the *similarity based approaches* make use of a distributional similarity measure between arguments and evaluate a heuristic scoring function:

$$S_{\text{rel}}(\text{arg}) = \sum_{\text{arg}' \in \text{Seen(rel)}} \text{sim}(\text{arg}, \text{arg}') \cdot \text{wt}_{\text{rel}}(\text{arg})$$

[1]Our repository of selectional preferences is available at `http://www.cs.washington.edu/research/ldasp`.

Erk (2007) showed the advantages of this approach over Resnik's information-theoretic class-based method on a pseudo-disambiguation evaluation. These methods obtain better lexical coverage, but are unable to obtain any abstract representation of selectional preferences.

Our solution fits into the general category of *generative probabilistic models*, which model each relation/argument combination as being generated by a latent class variable. These classes are automatically learned from the data. This retains the class-based flavor of the problem, without the knowledge limitations of the explicit class-based approaches. Probably the closest to our work is a model proposed by Rooth et al. (1999), in which each class corresponds to a multinomial over relations and arguments and EM is used to learn the parameters of the model. In contrast, we use a LinkLDA framework in which each relation is associated with a corresponding multinomial distribution over classes, and each argument is drawn from a class-specific distribution over words; LinkLDA captures co-occurrence of classes in the two arguments. Additionally we perform full Bayesian inference using collapsed Gibbs sampling, in which parameters are integrated out (Griffiths and Steyvers, 2004).

Recently, Bergsma *et. al.* (2008) proposed the first *discriminative approach* to selectional preferences. Their insight that pseudo-negative examples could be used as training data allows the application of an SVM classifier, which makes use of many features in addition to the relation-argument co-occurrence frequencies used by other methods. They automatically generated positive and negative examples by selecting arguments having high and low mutual information with the relation. Since it is a discriminative approach it is amenable to feature engineering, but needs to be retrained and tuned for each task. On the other hand, generative models produce complete probability distributions of the data, and hence can be integrated with other systems and tasks in a more principled manner (see Sections 4.2.2 and 4.3.1). Additionally, unlike LDA-SP Bergsma et al.'s system doesn't produce human-interpretable topics. Finally, we note that LDA-SP and Bergsma's system are potentially complimentary – the output of LDA-SP could be used to generate higher-quality training data for Bergsma, potentially improving their results.

Topic models such as LDA (Blei et al., 2003) and its variants have recently begun to see use in many NLP applications such as summarization (Daumé III and Marcu, 2006), document alignment and segmentation (Chen et al., 2009), and inferring class-attribute hierarchies (Reisinger and Pasca, 2009). Our particular model, LinkLDA, has been applied to a few NLP tasks such as simultaneously modeling the words appearing in blog posts and users who will likely respond to them (Yano et al., 2009), modeling topic-aligned articles in different languages (Mimno et al., 2009), and word sense induction (Brody and Lapata, 2009).

Finally, we highlight two systems, developed independently of our own, which apply LDA-style models to similar tasks. Ó Séaghdha (2010) proposes a series of LDA-style models for the task of computing selectional preferences. This work learns selectional preferences between the following grammatical relations: verb-object, noun-noun, and adjective-noun. It also focuses on jointly modeling the generation of both predicate and argument, and evaluation is performed on a set of human-plausibility judgments obtaining impressive results against Keller and Lapata's (2003) Web hit-count based system. Van Durme and Gildea (2009) proposed applying LDA to general knowledge templates extracted using the KNEXT system (Schubert and Tong, 2003). In contrast, our work uses LinkLDA and focuses on modeling multiple arguments of a relation (*e.g.*, the subject and direct object of a verb).

3 Topic Models for Selectional Prefs.

We present a series of topic models for the task of computing selectional preferences. These models vary in the amount of independence they assume between a_1 and a_2. At one extreme is IndependentLDA, a model which assumes that both a_1 and a_2 are generated completely independently. On the other hand, JointLDA, the model at the other extreme (Figure 1) assumes both arguments of a specific extraction are generated based on a single hidden variable z. LinkLDA (Figure 2) lies between these two extremes, and as demonstrated in Section 4, it is the best model for our relation data.

We are given a set R of binary relations and a corpus $\mathcal{D} = \{r(a_1, a_2)\}$ of extracted instances for these relations. [2] Our task is to compute, for each argument a_i of each relation r, a set of usual argument values (noun phrases) that it takes. For example, for the relation *is headquartered in* the first argument set will include companies like *Microsoft, Intel, General Motors* and second argument will favor locations like *New York, California, Seattle*.

3.1 IndependentLDA

We first describe the straightforward application of LDA to modeling our corpus of extracted relations. In this case two separate LDA models are used to model a_1 and a_2 independently.

In the generative model for our data, each relation r has a corresponding multinomial over topics θ_r, drawn from a Dirichlet. For each extraction, a hidden topic z is first picked according to θ_r, and then the observed argument a is chosen according to the multinomial β_z.

Readers familiar with topic modeling terminology can understand our approach as follows: we treat each relation as a document whose contents consist of a bags of words corresponding to all the noun phrases observed as arguments of the relation in our corpus. Formally, LDA generates each argument in the corpus of relations as follows:

for each topic $t = 1 \ldots T$ **do**
  Generate β_t according to symmetric Dirichlet distribution Dir(η).
end for
for each relation $r = 1 \ldots |R|$ **do**
  Generate θ_r according to Dirichlet distribution Dir(α).
  for each tuple $i = 1 \ldots N_r$ **do**
    Generate $z_{r,i}$ from Multinomial(θ_r).
    Generate the argument $a_{r,i}$ from multinomial $\beta_{z_{r,i}}$.
  end for
end for

One weakness of IndependentLDA is that it doesn't jointly model a_1 and a_2 together. Clearly this is undesirable, as information about which topics one of the arguments favors can help inform the topics chosen for the other. For example, class pairs such as (*team*, *game*), (*politician*, *political issue*) form much more plausible selectional preferences than, say, (*team*, *political issue*), (*politician*, *game*).

[2] We focus on binary relations, though the techniques presented in the paper are easily extensible to n-ary relations.

3.2 JointLDA

As a more tightly coupled alternative, we first propose JointLDA, whose graphical model is depicted in Figure 1. The key difference in JointLDA (versus LDA) is that instead of one, it maintains *two* sets of topics (latent distributions over words) denoted by β and γ, one for classes of each argument. A topic id k represents a pair of topics, β_k and γ_k, that co-occur in the arguments of extracted relations. Common examples include *(Person, Location), (Politician, Political issue), etc.* The hidden variable $z = k$ indicates that the noun phrase for the first argument was drawn from the multinomial β_k, and that the second argument was drawn from γ_k. The per-relation distribution θ_r is a multinomial over the topic ids and represents the selectional preferences, both for arg1s and arg2s of a relation r.

Although JointLDA has many desirable properties, it has some drawbacks as well. Most notably, in JointLDA topics correspond to pairs of multinomials (β_k, γ_k); this leads to a situation in which multiple redundant distributions are needed to represent the same underlying semantic class. For example consider the case where we we need to represent the following selectional preferences for our corpus of relations: (*person*, *location*), (*person*, *organization*), and (*person*, *crime*). Because JointLDA requires a separate pair of multinomials for each topic, it is forced to use 3 separate multinomials to represent the class *person*, rather than learning a single distribution representing *person* and choosing 3 different topics for a_2. This results in poor generalization because the data for a single class is divided into multiple topics.

In order to address this problem while maintaining the sharing of influence between a_1 and a_2, we next present LinkLDA, which represents a compromise between IndependentLDA and JointLDA. LinkLDA is more flexible than JointLDA, allowing different topics to be chosen for a_1, and a_2, however still models the generation of topics from the same distribution for a given relation.

3.3 LinkLDA

Figure 2 illustrates the LinkLDA model in the plate notation, which is analogous to the model in (Erosheva et al., 2004). In particular note that each a_i is drawn from a different hidden topic z_i, however the z_i's are drawn from the same distribution θ_r for a given relation r. To facilitate learning related topic pairs between arguments we employ a sparse prior over the per-relation topic distributions. Because a few topics are likely to be assigned most of the probability mass for a given relation it is more likely (although not necessary) that the same topic number k will be drawn for both arguments.

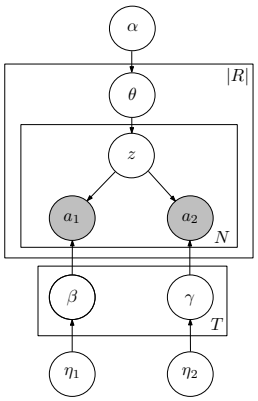

Figure 1: JointLDA

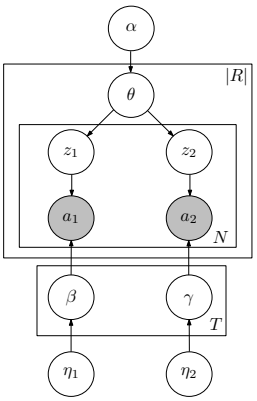

Figure 2: LinkLDA

When comparing LinkLDA with JointLDA the better model may not seem immediately clear. On the one hand, JointLDA jointly models the generation of both arguments in an extracted tuple. This allows one argument to help disambiguate the other in the case of ambiguous relation strings. LinkLDA, however, is more flexible; rather than requiring both arguments to be generated from one of $|Z|$ possible pairs of multinomials (β_z, γ_z), LinkLDA allows the arguments of a given extraction to be generated from $|Z|^2$ possible pairs. Thus, instead of imposing a hard constraint that $z_1 = z_2$ (as in JointLDA), LinkLDA simply assigns a higher probability to states in which $z_1 = z_2$, because both hidden variables are drawn from the same (sparse) distribution θ_r. LinkLDA can thus re-use argument classes, choosing different combinations of topics for the arguments if it fits the data better. In Section 4 we show experimentally that LinkLDA outperforms JointLDA (and IndependentLDA) by wide margins. We use LDA-SP to refer to LinkLDA in all the experiments below.

3.4 Inference

For all the models we use collapsed Gibbs sampling for inference in which each of the hidden variables (*e.g.*, $z_{r,i,1}$ and $z_{r,i,2}$ in LinkLDA) are sampled sequentially conditioned on a full-assignment to all others, integrating out the parameters (Griffiths and Steyvers, 2004). This produces robust parameter estimates, as it allows computation of expectations over the posterior distribution

as opposed to estimating maximum likelihood parameters. In addition, the integration allows the use of sparse priors, which are typically more appropriate for natural language data. In all experiments we use hyperparameters $\alpha = \eta_1 = \eta_2 = 0.1$. We generated initial code for our samplers using the Hierarchical Bayes Compiler (Daume III, 2007).

3.5 Advantages of Topic Models

There are several advantages to using topic models for our task. First, they naturally model the class-based nature of selectional preferences, but don't take a pre-defined set of classes as input. Instead, they compute the classes automatically. This leads to better lexical coverage since the issue of matching a new argument to a known class is side-stepped. Second, the models naturally handle ambiguous arguments, as they are able to assign different topics to the same phrase in different contexts. Inference in these models is also scalable – linear in both the size of the corpus as well as the number of topics. In addition, there are several scalability enhancements such as SparseLDA (Yao et al., 2009), and an approximation of the Gibbs Sampling procedure can be efficiently parallelized (Newman et al., 2009). Finally we note that, once a topic distribution has been learned over a set of training relations, one can efficiently apply inference to unseen relations (Yao et al., 2009).

4 Experiments

We perform three main experiments to assess the quality of the preferences obtained using topic models. The first is a task-independent evaluation using a pseudo-disambiguation experiment (Section 4.2), which is a standard way to evaluate the quality of selectional preferences (Rooth et al., 1999; Erk, 2007; Bergsma et al., 2008). We use this experiment to compare the various topic models as well as the best model with the known state of the art approaches to selectional preferences. Secondly, we show significant improvements to performance at an end-task of textual inference in Section 4.3. Finally, we report on the quality of a large database of Wordnet-based preferences obtained after manually associating our topics with Wordnet classes (Section 4.4).

4.1 Generalization Corpus

For all experiments we make use of a corpus of $r(a_1, a_2)$ tuples, which was automatically extracted by TEXTRUNNER (Banko and Etzioni, 2008) from 500 million Web pages.

To create a *generalization corpus* from this large dataset. We first selected 3,000 relations from the middle of the tail (we used the 2,000-5,000 most frequent ones)[3] and collected all instances. To reduce sparsity, we discarded all tuples containing an NP that occurred fewer than 50 times in the data. This resulted in a vocabulary of about 32,000 noun phrases, and a set of about 2.4 million tuples in our generalization corpus.

We inferred topic-argument and relation-topic multinomials (β, γ, and θ) on the generalization corpus by taking 5 samples at a lag of 50 after a burn in of 750 iterations. Using multiple samples introduces the risk of *topic drift* due to lack of identifiability, however we found this to not be a problem in practice. During development we found that the topics tend to remain stable across multiple samples after sufficient burn in, and multiple samples improved performance. Table 1 lists sample topics and high ranked words for each (for both arguments) as well as relations favoring those topics.

4.2 Task Independent Evaluation

We first compare the three LDA-based approaches to each other and two state of the art similarity based systems (Erk, 2007) (using mutual information and Jaccard similarity respectively). These similarity measures were shown to outperform the generative model of Rooth et al. (1999), as well as class-based methods such as Resnik's. In this pseudo-disambiguation experiment an observed tuple is paired with a pseudo-negative, which has both arguments randomly generated from the whole vocabulary (according to the corpus-wide distribution over arguments). The task is, for each relation-argument pair, to determine whether it is observed, or a random distractor.

4.2.1 Test Set

For this experiment we gathered a primary corpus by first randomly selecting 100 high-frequency relations *not* in the generalization corpus. For each relation we collected all tuples containing arguments in the vocabulary. We held out 500 randomly selected tuples as the test set. For each tu-

[3] Many of the most frequent relations have very weak selectional preferences, and thus provide little signal for inferring meaningful topics. For example, the relations *has* and *is* can take just about any arguments.

Topic t	Arg1	Relations which assign highest probability to t	Arg2
18	The residue - The mixture - The reaction mixture - The solution - the mixture - the reaction mixture - the residue - The reaction - the solution - The filtrate - the reaction - The product - The crude product - The pellet - The organic layer - Thereto - This solution - The resulting solution - Next - The organic phase - The resulting mixture - C.)	was treated with, is treated with, was poured into, was extracted with, was purified by, was diluted with, was filtered through, is disolved in, is washed with	EtOAc - CH2Cl2 - H2O - CH.sub.2Cl.sub.2 - H.sub.2O - water - MeOH - NaHCO3 - Et2O - NHCl - CHCl.sub.3 - NHCl - dropwise - CH2Cl.sub.2 - Celite - Et.sub.2O - Cl.sub.2 - NaOH - AcOEt - CH2C12 - the mixture - saturated NaHCO3 - SiO2 - H2O - N hydrochloric acid - NHCl - preparative HPLC - to0 C
151	the Court - The Court - the Supreme Court - The Supreme Court - this Court - Court - The US Supreme Court - the court - This Court - the US Supreme Court - The court - Supreme Court - Judge - the Court of Appeals - A federal judge	will hear, ruled in, decides, upholds, struck down, overturned, sided with, affirms	the case - the appeal - arguments - a case - evidence - this case - the decision - the law - testimony - the State - an interview - an appeal - cases - the Court - that decision - Congress - a decision - the complaint - oral arguments - a law - the statute
211	President Bush - Bush - The President - Clinton - the President - President Clinton - President George W. Bush - Mr. Bush - The Governor - the Governor - Romney - McCain - The White House - President - Schwarzenegger - Obama	hailed, vetoed, promoted, will deliver, favors, denounced, defended	the bill - a bill - the decision - the war - the idea - the plan - the move - the legislation - legislation - the measure - the proposal - the deal - this bill - a measure - the program - the law - the resolution - efforts - the agreement - gay marriage - the report - abortion
224	Google - Software - the CPU - Clicking - Excel - the user - Firefox - System - The CPU - Internet Explorer - the ability - Program - users - Option - SQL Server - Code - the OS - the BIOS	will display, to store, to load, processes, cannot find, invokes, to search for, to delete	data - files - the data - the file - the URL - information - the files - images - a URL - the information - the IP address - the user - text - the code - a file - the page - IP addresses - PDF files - messages - pages - an IP address

Table 1: Example argument lists from the inferred topics. For each topic number t we list the most probable values according to the multinomial distributions for each argument (β_t and γ_t). The middle column reports a few relations whose inferred topic distributions θ_r assign highest probability to t.

ple $r(a_1, a_2)$ in the held-out set, we removed all tuples in the training set containing either of the *rel-arg* pairs, *i.e.*, any tuple matching $r(a_1, *)$ or $r(*, a_2)$. Next we used collapsed Gibbs sampling to infer a distribution over topics, θ_r, for each of the relations in the primary corpus (based solely on tuples in the training set) using the topics from the generalization corpus.

For each of the 500 observed tuples in the test-set we generated a pseudo-negative tuple by randomly sampling two noun phrases from the distribution of NPs in both corpora.

4.2.2 Prediction

Our prediction system needs to determine whether a specific relation-argument pair is admissible according to the selectional preferences or is a random distractor (D). Following previous work, we perform this experiment independently for the two relation-argument pairs (r, a_1) and (r, a_2).

We first compute the probability of observing a_1 for first argument of relation r given that it is not a distractor, $P(a_1|r, \neg D)$, which we approximate by its probability given an estimate of the parameters inferred by our model, marginalizing over hidden topics t. The analysis for the second argument is similar.

$$\begin{aligned} P(a_1|r, \neg D) \approx P_{LDA}(a_1|r) &= \sum_{t=0}^{T} P(a_1|t)P(t|r) \\ &= \sum_{t=0}^{T} \beta_t(a_1)\theta_r(t) \end{aligned}$$

A simple application of Bayes Rule gives the probability that a particular argument is not a distractor. Here the distractor-related probabilities are independent of r, *i.e.*, $P(D|r) = P(D)$, $P(a_1|D, r) = P(a_1|D)$, *etc.* We estimate $P(a_1|D)$ according to their frequency in the generalization corpus.

$$\begin{aligned} P(\neg D|r, a_1) &= \frac{P(\neg D|r)P(a_1|r, \neg D)}{P(a_1|r)} \\ &\approx \frac{P(\neg D)P_{LDA}(a_1|r)}{P(D)P(a_1|D) + P(\neg D)P_{LDA}(a_1|r)} \end{aligned}$$

4.2.3 Results

Figure 3 plots the precision-recall curve for the pseudo-disambiguation experiment comparing the three different topic models. LDA-SP, which uses LinkLDA, substantially outperforms both IndependentLDA and JointLDA.

Next, in figure 4, we compare LDA-SP with mutual information and Jaccard similarities using both the generalization and primary corpus for

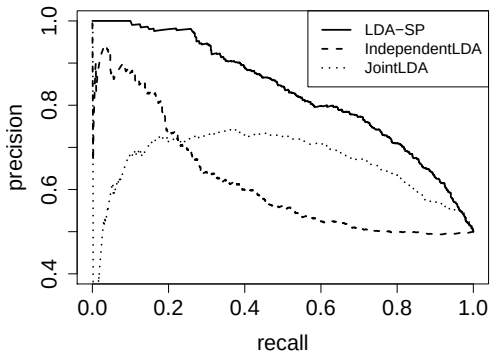

Figure 3: Comparison of LDA-based approaches on the pseudo-disambiguation task. LDA-SP (LinkLDA) substantially outperforms the other models.

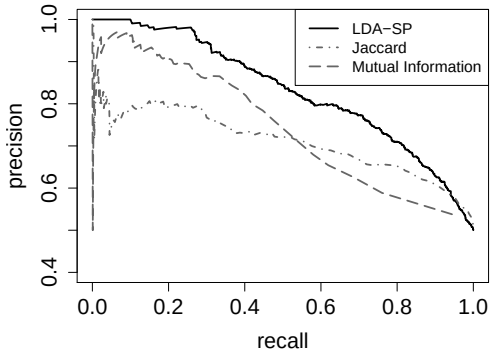

Figure 4: Comparison to similarity-based selectional preference systems. LDA-SP obtains 85% higher recall at precision 0.9.

computation of similarities. We find LDA-SP significantly outperforms these methods. Its edge is most noticed at high precisions; it obtains 85% more recall at 0.9 precision compared to mutual information. Overall LDA-SP obtains an 15% increase in the area under precision-recall curve over mutual information. All three systems' AUCs are shown in Table 2; LDA-SP's improvements over both Jaccard and mutual information are highly significant with a significance level less than 0.01 using a paired t-test.

In addition to a superior performance in selectional preference evaluation LDA-SP also produces a set of coherent topics, which can be useful in their own right. For instance, one could use them for tasks such as set-expansion (Carlson et al., 2010) or automatic thesaurus induction (Etzioni et al., 2005; Kozareva et al., 2008).

	LDA-SP	MI-Sim	Jaccard-Sim
AUC	0.833	0.727	0.711

Table 2: Area under the precision recall curve. LDA-SP's AUC is significantly higher than both similarity-based methods according to a paired t-test with a significance level below 0.01.

4.3 End Task Evaluation

We now evaluate LDA-SP's ability to improve performance at an end-task. We choose the task of improving textual entailment by learning selectional preferences for inference rules and filtering inferences that do not respect these. This application of selectional preferences was introduced by Pantel *et. al.* (2007). For now we stick to inference rules of the form $r_1(a_1, a_2) \Rightarrow r_2(a_1, a_2)$, though our ideas are more generally applicable to more complex rules. As an example, the rule (X *defeats* Y) $\Rightarrow$ (X *plays* Y) holds when X and Y are both sports teams, however fails to produce a reasonable inference if X and Y are *Britain* and *Nazi Germany* respectively.

4.3.1 Filtering Inferences

In order for an inference to be plausible, both relations must have similar selectional preferences, and further, the arguments must obey the selectional preferences of both the antecedent r_1 and the consequent r_2.[4] Pantel et al. (2007) made use of these intuitions by producing a set of class-based selectional preferences for each relation, then filtering out any inferences where the arguments were incompatible with the intersection of these preferences. In contrast, we take a probabilistic approach, evaluating the quality of a specific inference by measuring the probability that the arguments in both the antecedent and the consequent were drawn from the same hidden topic in our model. Note that this probability captures both the requirement that the antecedent and consequent have similar selectional preferences, and that the arguments from a particular instance of the rule's application match their overlap.

We use $z_{r_i,j}$ to denote the topic that generates the j^{th} argument of relation r_i. The probability that the two arguments a_1, a_2 were drawn from the same hidden topic factorizes as follows due to the conditional independences in our model:[5]

$$P(z_{r_1,1} = z_{r_2,1}, z_{r_1,2} = z_{r_2,2}|a_1, a_2) = P(z_{r_1,1} = z_{r_2,1}|a_1)P(z_{r_1,2} = z_{r_2,2}|a_2)$$

[4] Similarity-based and discriminative methods are not applicable to this task as they offer no straightforward way to compare the similarity between selectional preferences of two relations.

[5] Note that all probabilities are conditioned on an estimate of the parameters θ, β, γ from our model, which are omitted for compactness.

To compute each of these factors we simply marginalize over the hidden topics:

$$P(z_{r_1,j} = z_{r_2,j}|a_j) = \sum_{t=1}^{T} P(z_{r_1,j} = t|a_j)P(z_{r_2,j} = t|a_j)$$

where $P(z = t|a)$ can be computed using Bayes rule. For example,

$$\begin{aligned} P(z_{r_1,1} = t|a_1) &= \frac{P(a_1|z_{r_1,1} = t)P(z_{r_1,1} = t)}{P(a_1)} \\ &= \frac{\beta_t(a_1)\theta_{r_1}(t)}{P(a_1)} \end{aligned}$$

4.3.2 Experimental Conditions

In order to evaluate LDA-SP's ability to filter inferences based on selectional preferences we need a set of inference rules between the relations in our corpus. We therefore mapped the DIRT Inference rules (Lin and Pantel, 2001), (which consist of pairs of dependency paths) to TEXTRUNNER relations as follows. We first gathered all instances in the generalization corpus, and for each $r(a_1, a_2)$ created a corresponding simple sentence by concatenating the arguments with the relation string between them. Each such simple sentence was parsed using Minipar (Lin, 1998). From the parses we extracted all dependency paths between nouns that contain only words present in the TEXTRUNNER relation string. These dependency paths were then matched against each pair in the DIRT database, and all pairs of associated relations were collected producing about 26,000 inference rules.

Following Pantel et al. (2007) we randomly sampled 100 inference rules. We then automatically filtered out any rules which contained a negation, or for which the antecedent and consequent contained a pair of antonyms found in WordNet (this left us with 85 rules). For each rule we collected 10 random instances of the antecedent, and generated the consequent. We randomly sampled 300 of these inferences to hand-label.

4.3.3 Results

In figure 5 we compare the precision and recall of LDA-SP against the top two performing systems described by Pantel et al. (ISP.IIM-$\vee$ and ISP.JIM, both using the CBC clusters (Pantel, 2003)). We find that LDA-SP achieves both higher precision and recall than ISP.IIM-$\vee$. It is also able to achieve the high-precision point of ISP.JIM and can trade precision to get a much larger recall.

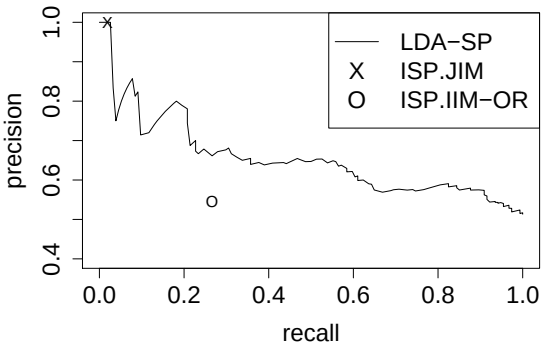

Figure 5: Precision and recall on the inference filtering task.

Top 10 Inference Rules Ranked by LDA-SP		
antecedent	consequent	KL-div
will begin at	will start at	0.014999
shall review	shall determine	0.129434
may increase	may reduce	0.214841
walk from	walk to	0.219471
consume	absorb	0.240730
shall keep	shall maintain	0.264299
shall pay to	will notify	0.290555
may apply for	may obtain	0.313916
copy	download	0.316502
should pay	must pay	0.371544
Bottom 10 Inference Rules Ranked by LDA-SP		
antecedent	consequent	KL-div
lose to	shall take	10.011848
should play	could do	10.028904
could play	get in	10.048857
will start at	move to	10.060994
shall keep	will spend	10.105493
should play	get in	10.131299
shall pay to	leave for	10.131364
shall keep	return to	10.149797
shall keep	could do	10.178032
shall maintain	have spent	10.221618

Table 3: Top 10 and Bottom 10 ranked inference rules ranked by LDA-SPafter automatically filtering out negations and antonyms (using WordNet).

In addition we demonstrate LDA-SP's ability to rank inference rules by measuring the Kullback Leibler Divergence[6] between the topic-distributions of the antecedent and consequent, θ_{r_1} and θ_{r_2} respectively. Table 3 shows the top 10 and bottom 10 rules out of the 26,000 ranked by KL Divergence after automatically filtering antonyms (using WordNet) and negations. For slight variations in rules (*e.g.*, symmetric pairs) we mention only one example to show more variety.

[6]KL-Divergence is an information-theoretic measure of the similarity between two probability distributions, and defined as follows: $KL(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$.

4.4 A Repository of Class-Based Preferences

Finally we explore LDA-SP's ability to produce a repository of human interpretable class-based selectional preferences. As an example, for the relation *was born in*, we would like to infer that the plausible arguments include (*person*, *location*) and (*person*, *date*).

Since we already have a set of topics, our task reduces to mapping the inferred topics to an equivalent class in a taxonomy (*e.g.*, WordNet). We experimented with automatic methods such as Resnik's, but found them to have all the same problems as directly applying these approaches to the SP task.[7] Guided by the fact that we have a relatively small number of topics (600 total, 300 for each argument) we simply chose to label them manually. By labeling this small number of topics we can infer class-based preferences for an arbitrary number of relations.

In particular, we applied a semi-automatic scheme to map topics to WordNet. We first applied Resnik's approach to automatically shortlist a few candidate WordNet classes for each topic. We then manually picked the best class from the shortlist that best represented the 20 top arguments for a topic (similar to Table 1). We marked all incoherent topics with a special symbol $\emptyset$. This process took one of the authors about 4 hours to complete.

To evaluate how well our topic-class associations carry over to unseen relations we used the same random sample of 100 relations from the pseudo-disambiguation experiment.[8] For each argument of each relation we picked the top two topics according to frequency in the 5 Gibbs samples. We then discarded any topics which were labeled with $\emptyset$; this resulted in a set of 236 predictions. A few examples are displayed in table 4.

We evaluated these classes and found the accuracy to be around 0.88. We contrast this with Pantel's repository,[9] the only other released database of selectional preferences to our knowledge. We evaluated the same 100 relations from his website and tagged the top 2 classes for each argument and evaluated the accuracy to be roughly 0.55.

[7] Perhaps recent work on automatic coherence ranking (Newman et al., 2010) and labeling (Mei et al., 2007) could produce better results.

[8] Recall that these 100 were not part of the original 3,000 in the generalization corpus, and are, therefore, representative of new "unseen" relations.

[9] http://demo.patrickpantel.com/Content/LexSem/paraphrase.htm

arg1 class	relation	arg2 class
politician#1	was running for	leader#1
people#1	will love	show#3
organization#1	has responded to	accusation#2
administrative_unit#1	has appointed	administrator#3

Table 4: Class-based Selectional Preferences.

We emphasize that tagging a pair of class-based preferences is a highly subjective task, so these results should be treated as preliminary. Still, these early results are promising. We wish to undertake a larger scale study soon.

5 Conclusions and Future Work

We have presented an application of topic modeling to the problem of automatically computing selectional preferences. Our method, LDA-SP, learns a distribution over topics for each relation while simultaneously grouping related words into these topics. This approach is capable of producing human interpretable classes, however, avoids the drawbacks of traditional class-based approaches (poor lexical coverage and ambiguity). LDA-SP achieves state-of-the-art performance on predictive tasks such as pseudo-disambiguation, and filtering incorrect inferences.

Because LDA-SP generates a complete probabilistic model for our relation data, its results are easily applicable to many other tasks such as identifying similar relations, ranking inference rules, *etc.* In the future, we wish to apply our model to automatically discover new inference rules and paraphrases.

Finally, our repository of selectional preferences for 10,000 relations is available at http://www.cs.washington.edu/research/ldasp.

Acknowledgments

We would like to thank Tim Baldwin, Colin Cherry, Jesse Davis, Elena Erosheva, Stephen Soderland, Dan Weld, in addition to the anonymous reviewers for helpful comments on a previous draft. This research was supported in part by NSF grant IIS-0803481, ONR grant N00014-08-1-0431, DARPA contract FA8750-09-C-0179, a National Defense Science and Engineering Graduate (NDSEG) Fellowship 32 CFR 168a, and carried out at the University of Washington's Turing Center.

References

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *ACL-08: HLT*.

Shane Bergsma, Dekang Lin, and Randy Goebel. 2008. Discriminative learning of selectional preference from unlabeled text. In *EMNLP*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*

Samuel Brody and Mirella Lapata. 2009. Bayesian word sense induction. In *EACL*, pages 103–111, Morristown, NJ, USA. Association for Computational Linguistics.

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *WSDM 2010*.

Harr Chen, S. R. K. Branavan, Regina Barzilay, and David R. Karger. 2009. Global models of document structure using latent permutations. In *NAACL*.

Stephen Clark and David Weir. 2002. Class-based probability estimation using a semantic hierarchy. *Comput. Linguist.*

Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. 1999. Similarity-based models of word cooccurrence probabilities. In *Machine Learning*.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

Hal Daume III. 2007. hbc: Hierarchical bayes compiler. http://hal3.name/hbc.

Katrin Erk. 2007. A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

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*.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana maria Popescu, Tal Shaked, Stephen Soderl, Daniel S. Weld, and Alex Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Comput. Linguist.*

T. L. Griffiths and M. Steyvers. 2004. Finding scientific topics. *Proc Natl Acad Sci U S A*.

Frank Keller and Mirella Lapata. 2003. Using the web to obtain frequencies for unseen bigrams. *Comput. Linguist.*

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *ACL-08: HLT*.

Hang Li and Naoki Abe. 1998. Generalizing case frames using a thesaurus and the mdl principle. *Comput. Linguist.*

Dekang Lin and Patrick Pantel. 2001. Dirt-discovery of inference rules from text. In *KDD*.

Dekang Lin. 1998. Dependency-based evaluation of minipar. In *Proc. Workshop on the Evaluation of Parsing Systems*.

Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. 2007. Automatic labeling of multinomial topic models. In *KDD*.

David Mimno, Hanna M. Wallach, Jason Naradowsky, David A. Smith, and Andrew McCallum. 2009. Polylingual topic models. In *EMNLP*.

David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2009. Distributed algorithms for topic models. *JMLR*.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *NAACL-HLT*.

Diarmuid Ó Séaghdha. 2010. Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard H. Hovy. 2007. Isp: Learning inferential selectional preferences. In *HLT-NAACL*.

Patrick Andre Pantel. 2003. *Clustering by committee*. Ph.D. thesis, University of Alberta, Edmonton, Alta., Canada.

Joseph Reisinger and Marius Pasca. 2009. Latent variable models of concept-attribute attachment. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.

P. Resnik. 1996. Selectional constraints: an information-theoretic model and its computational realization. *Cognition*.

Philip Resnik. 1997. Selectional preference and sense disambiguation. In *Proc. of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*

Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via em-based clustering. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*.

Lenhart Schubert and Matthew Tong. 2003. Extracting and evaluating general world knowledge from the brown corpus. In *In Proc. of the HLT-NAACL Workshop on Text Meaning*, pages 7–13.

Benjamin Van Durme and Daniel Gildea. 2009. Topic models for corpus-centric knowledge generalization. In *Technical Report TR-946, Department of Computer Science, University of Rochester, Rochester*.

Tae Yano, William W. Cohen, and Noah A. Smith. 2009. Predicting response to political blog posts with topic models. In *NAACL*.

L. Yao, D. Mimno, and A. Mccallum. 2009. Efficient methods for topic model inference on streaming document collections. In *KDD*.