Unsupervised Learning of Selectional Restrictions and Detection of Argument Coercions

Kirk Roberts and Sanda M. Harabagiu
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083, USA
{kirk,sanda}@hlt.utdallas.edu

Abstract

Metonymic language is a pervasive phenomenon. Metonymic type shifting, or argument *type coercion*, results in a selectional restriction violation where the argument's semantic class differs from the class the predicate expects. In this paper we present an unsupervised method that learns the selectional restriction of arguments and enables the detection of argument coercion. This method also generates an enhanced probabilistic resolution of logical metonymies. The experimental results indicate substantial improvements the detection of coercions and the ranking of metonymic interpretations.

1 Introduction

Metonymic language is pervasive in today's social interactions. For example, it is typical to find questions that require metonymic resolution:

(Q1) Did you enjoy War and Peace?
(Q2) Does anyone have any advice on how to start a bowling team?[1]

In order to process such questions and capture the intention of the person that posed them, coercions are needed. Question (Q1) is interpreted as whether you enjoyed *reading* "War and Peace", while (Q2) is interpreted as asking for advice on *organizing*, *forming*, or *registering* a bowling team. The quality of the answers therefore depends on the ability to (1) recognize when metonymic language is used, and (2) to produce coercions that capture the user's intention. One important step in this direction was taken by SemEval-2010 Task 7, which focused on the ability to recognize (a) an argument's selectional restriction for predicates such as *arrive at*, *cancel*, or *hear*, and (b) the type of coercion that licensed a correct interpretation of the metonymy. Details of the task are reported in (Pustejovsky et al., 2010). Approaches to metonymy based on this task are limited, however, because (a) the task is focused only on semantically non-ambiguous predicates and (b) the selectional restrictions of the arguments were chosen from a pre-defined set of six semantic classes (artifact, document, event, location, proposition, and sound). However, metonymy coercion systems capable of providing the interpretations of questions (Q1) and (Q2) clearly cannot operate with the simplifications designed for this task.

Inspired by recent advances in modeling selectional preferences with latent-variable models (Ritter et al., 2010; Ó Séaghdha, 2010), we propose an unsupervised model for learning selectional restrictions. The model assumes that (1) arguments have a single selected class exemplified by the selectional restriction, and (2) the selected class can be inferred from the data, in part by modeling how coercive each predicate is. The model is capable of operating with both ambiguous and disambiguated predicates, producing superior results for predicates that have been disambiguated. The selectional restrictions and coercions detected by the model reported in this paper can be used to enhance the logical metonymy approach reported in Lapata and Lascarides (2003). The experimental results show a significant improvement in the ranking of interpretations.

[1] Both questions taken from Yahoo Answers.

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 980–990,
Edinburgh, Scotland, UK, July 27–31, 2011.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 details unsupervised models that inform detection of metonymies. Section 4 outlines a method for disambiguating ambiguous predicates. Section 5 describes the enhanced interpretation of logical metonymies when conventional constraints are known. Section 6 outlines our implementation and experimental design. Section 7 presents our experimental results in three broad tasks: (i) semantic class induction, (ii) coercion detection, and (iii) logical metonymy interpretation. Section 8 summarizes the conclusions.

2 Previous Work

Lapata and Lascarides (2003) propose a probabilistic ranking model for logical metonymies. They estimate these probabilities using co-occurrence frequencies of predicate-argument pairs in a corpus. Shutova (2009) extends this approach to provide sense-disambiguated interpretations from WordNet (Fellbaum, 1998) by using the alternative interpretations to disambiguate polysemous words. Shutova and Teufel (2009) extend this approach further by clustering these sense-disambiguated interpretations into distinct groups of meaning (e.g., {*read*, *browse*, *look through*} and {*write*, *produce*, *work on*} for "*enjoy book*"). Not only do these approaches assume logical metonymies have already been identified, but they are susceptible to providing interpretations that are themselves logical metonymies (e.g., *finish book*). In this paper, we propose an enhancement to resolving logical metonymies by ruling out event-invoking predicates in order to provide more semantically valid interpretations.

Recently, the resolution of several linguistic problems has benefited from Latent Dirichlet Allocation (LDA) (Blei et al., 2003) models. Ó Séaghdha (2010) examines several selectional preference models based on LDA in predicting human judgements on predicate-argument plausibility. Both LDA and an extension, ROOTH-LDA (based on Rooth et al. (1999)), perform well at predicting plausibility on unseen predicate-argument pairs. Inspired by these results, we propose to extend selectional preference models in order to learn selectional restrictions.

Alternatively, unsupervised algorithms exist that both induce semantic classes (Rooth et al., 1999; Lin and Pantel, 2001) and cluster predicates by their selectional restrictions (Rumshisky et al., 2007) but none of these provide a sufficient framework for determining if a specific argument violates its predicate's selectional restriction.

3 Unsupervised Learning of Selectional Restrictions

In predicate-argument structures, predicates impose selectional restrictions in the form of semantic expectations on their arguments. Whenever the semantic class of the argument meets these constraints a *selection* occurs. For example, the predicate "*hear*" imposes the semantics related to sound on the argument "*voice*". Because the semantic class for "*voice*" conforms to these constraints, we call its semantic class the *selected class*. However, when the semantic class of the argument violates these constraints, we follow Pustejovsky et al. (2010) and refer to this as a *coercion*. In this case, we call the argument's semantic class the *coerced class*. For example, "*hear speaker*" is a coercion where the argument class, person, is implicitly coerced into the voice of the speaker, a sound.

3.1 A Baseline Model

We consider the LDA-based selectional preference model reported in Ó Séaghdha (2010) as a baseline for modeling selectional restrictions. Formally, we define our LDA baseline model as follows. Let V be the predicate vocabulary size, let A be the argument vocabulary size, and let K be the number of argument classes. Let a_i^v be the i^{th} (non-unique) argument realized by predicate v. Let c_i^v be the class for a_i^v. Let θ^v be the class distribution for predicate v and ϕ^k be the argument distribution for class k. The graphical model for this LDA is shown in Figure 1(a). The generative process for LDA is:

For each argument class $k = 1..K$:
1. Choose $\phi^k \sim$ Dirichlet(β)

For each unique predicate $v = 1..V$:
2. Choose $\theta^v \sim$ Dirichlet(α)

For every argument $i = 1..n^v$:
3. Choose $c_i^v \sim$ Multinomial(θ^v)
4. Choose $a_i^v \sim$ Multinomial($\phi^{c_i^v}$)

Following Griffiths and Steyvers (2004), we collapse θ and ϕ and estimate the model using Gibbs

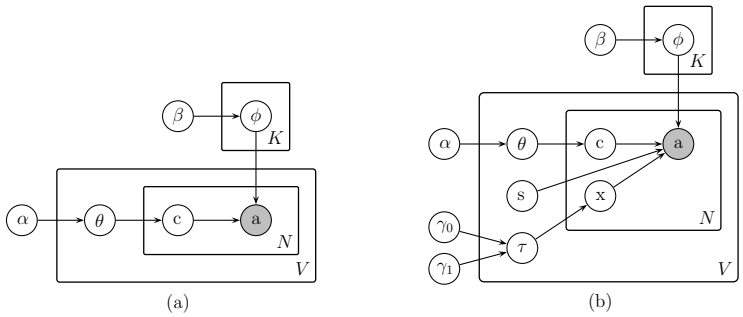

Figure 1: Graphical models for (a) LDA, and (b) coercion LDA (cLDA).

Sampling. This yields the update equation:

$$p(c_i^v = k|\mathbf{a}^v; \alpha, \beta) \propto \frac{f_{vk} + \alpha}{f_v + K\alpha} \frac{f_{ak} + \beta}{f_k + A\beta} \quad (1)$$

Where f_{ak} is the frequency of argument a being assigned class k; f_k is the frequency of class k being assigned to any argument; f_{vk} is the frequency of predicate v having an argument of class k; and f_v is the total number of non-unique arguments for predicate v.

3.2 A Coercion Model

We now incorporate our assumptions for selectional restriction modeling. Namely: (1) there is one selected class per predicate, and (2) the predicate's selected class can be chosen from the classes of its arguments. To accomplish this, we must also account for the coerciveness of each predicate. We assign a latent variable τ^v for each predicate v that controls how coercive v should be. The additional hyper-parameters γ_0 and γ_1 act as priors on τ^v. The generative process for this coercion LDA model, which we denote cLDA, is:

For each argument class $k = 1..K$:
1. Choose $\phi^k \sim$ Dirichlet(β)

For each unique predicate $v = 1..V$:
2. Choose $s^v \sim$ Uniform(1,K)
3. Choose $\theta^v \sim$ Dirichlet(α)[2]
4. Choose $\tau^v \sim$ Beta(γ_0, γ_1)

For every argument $i = 1..n^v$:
5. Choose $c_i^v \sim$ Multinomial(θ^v)
6. Choose $x_i^v \sim$ Bernoulli(τ^v)
7. If $x_i^v = 1$, Choose $a_i^v \sim$ Multinomial($\phi^{c_i^v}$)
   Else Choose $a_i^v \sim$ Multinomial(ϕ^{s^v})

The model variable s^v represents the selected class for predicate v. The coerced class is represented for each argument i by c_i^v, where x_i^v chooses between the selected and coerced class. The variable x_i^v is similar to switching variables in other graphical models such as Chemudugunta et al. (2007) and Reisinger and Mooney (2010), where switching variables are used to choose between a background distribution and a document-specific distribution. In this case, the switching variable chooses between a specific class and a predicate-specific distribution. The graphical model for cLDA is shown in Figure 1(b). Note that cLDA is virtually equivalent to LDA when τ^v is 1 and γ_1 is small because the selected class will be ignored. In this way, highly coercive predicates have less of an impact on the argument clustering because they are more reliant on the multinomial θ. We use Gibbs sampling to perform model inference and collapse θ, ϕ, and τ, integrating them out using multinomial-Dirichlet conjugacy (the Beta distribution used by τ is just a special case of the Dirichlet with only two parameters).

The update formula for the selected class s^v is:

$$\begin{aligned} p(s^v = k|\mathbf{a}^v, \mathbf{c}^v, \mathbf{x}^v; \alpha, \beta) \\ &\propto \prod_i^{n^v} P(a_i^v|s^v = k; \beta) \\ &\propto \prod_{i \in S^v} \frac{f_{a_i^v k} + \beta}{f_k + A\beta} \end{aligned} \quad (2)$$

Where n^v is the number of argument observations for predicate v; S^v is the set of arguments of v that are selections; and $f_{a_i^v k}$ is the frequency of word a_i^v being assigned to class k for any predicate. We then sample c_i^v and x_i^v jointly:

$$\begin{aligned} p(c_i^v = k, x_i^v = q|s^v, \mathbf{c}_{\bar{i}}^v, \mathbf{x}_{\bar{i}}^v, \mathbf{a}^v; \alpha, \beta, \gamma) \\ &\propto p(c_i^v{=}k; \alpha) p(x_i^v{=}q; \gamma) p(a_i^v|s^v, \mathbf{c}_{\bar{i}}^v, \mathbf{x}_{\bar{i}}^v, \mathbf{a}^v; \beta) \\ &\propto \frac{f_{vk} + \alpha}{f_v + K\alpha} \frac{f_{vq} + \gamma_q}{f_{v0} + \gamma_0 + f_{v1} + \gamma_1} \frac{f_{az} + \beta}{f_z + A\beta} \end{aligned} \quad (3)$$

[2]With the exception that the probability of drawing the selected class s^v is zero. This can be seen as drawing the multinomial θ^v from a Dirichlet distribution with K-1 components.

Where f_{vq}, f_{v0}, and f_{v1} is the frequency of x values that equal q, 0, and 1, respectively, for predicate v; f_{az} is the frequency of word a being in class z and f_z is the frequency all words being in class z, where z is defined as being equal to k when $x_i^v = 1$, or s^v when $x_i^v = 0$.

Note that Equation (2) results in a sampling of the selected class for v proportional to the number of arguments in each class for v, fulfilling our second assumption. Also note from Equation (3), the second term corresponds to the coerciveness of the predicate. When the predicate is very coercive, the marginal probability associated with $x_i^v = 0$ will be very low. If all predicates become entirely coercive, most x values will become 1 and the cLDA will become almost equivalent to an LDA model.

3.3 Coercion Detection

After the latent parameters have been estimated, we still require a method to determine if a given predicate-argument pair is a coercion or not. We assign a score in $[0, 1]$ instead of a binary value. Higher scores (near 1) indicate high likelihood of selection, while lower scores (near 0) indicate coercion. The LDA model must rely on a scoring method using the predicate-class and argument-class mixtures:

$$C_1(v, a) = \sum_k^K P(k|v)P(a|k) = \sum_k^K \theta_k^v \phi_a^k \quad (4)$$

Where θ_k^v represents the probability of any argument of v being in the class k and ϕ_a^k represents the probability of the argument a being in class k for any predicate. C_1 is also available as a scoring method for cLDA by including the proportion of the selected class s^v in θ. Note that since θ and ϕ are integrated out for both LDA and cLDA, we instead use their frequencies smoothed with α and β, respectively, which is their maximum likelihood estimate.

The cLDA model contains two useful parameters that can identify selections and coercions: the selected class s and the coercion indicator x. This yields two more coercion scoring metrics:

$$C_2(v, a) = P(a|s^v) = \phi_a^{s^v} \quad (5)$$

$$C_3(v, a) = P(x_a^v = 0|v, a) = 1.0 - \frac{\sum_{i \in I_a^v} x_i^v}{|I_a^v|} \quad (6)$$

Where s^v is the selected class for predicate v; I_a^v is the set of predicate-argument instances for predicate v and argument a; and x_i^v is 0 for a selection and 1 for a coercion. Of the three metrics, C_3 is the most direct measure of a coercion as it represents the average decision the model learned on the same predicate-argument pair. However, C_3 requires a large sample of instances for a particular predicate and argument, and so may be quite sparse. In practice, these different metrics have their own strengths and weaknesses and the best performing method often depends on the final task.

4 Predicate Sense Induction

Our assumption of a single selected class per predicate ignores predicate polysemy. However, the same lexical item may have multiple meanings, each with a separate selected class. We therefore propose a method of partitioning a predicate's arguments by the induced senses of the predicate. This allows separate induced predicates to each select a separate argument class. Consider the verb *fire*, which has at least two distinct common senses: (1) to shoot or propel an object (e.g., to fire a gun), and (2) to lay someone off (e.g., to fire an employee). The first sense selects a weapon (e.g., gun, bullet, rocket), while the second sense selects a person (e.g., employee, coach, apprentice).

Specifically, we employ tiered clustering (Reisinger and Mooney, 2010) using the words in the predicate's context. Tiered clustering is a discrete clustering method, as opposed to methods such as (Brody and Lapata, 2009) that assign a distribution of word senses to each word instance. Tiered clustering has several advantages over other discrete clustering approaches. First, tiered clustering learns a background word distribution in addition to the clusters. This reduces the impact that words common to most senses have on the clustering process and allow clusters to form around only the most salient words. Second, tiered clustering

Cluster 1 (18,391)	Cluster 2 (16,651)	Cluster 3 (18,749)	Cluster 4 (11,833)
shots	ball	hire	gun
gun	puck	letter	imagination
Israeli	hired	Yeltsin	grill
missiles	owner	minister	laser
rockets	shots	workforce	cells
officers	coaches	executives	engine
soldiers	net	employee	brain
rounds	circle	managers	!
bullets	Johnson	hired	engines
weapons	Williams	union	fire

Table 1: Context word clusters resulting from tiered clustering for the verb *fire* (includes the number of unique words belonging to each cluster).

uses a Chinese Restaurant Process (CRP) prior to control both the formation of new clusters (senses) and the bias toward larger clusters (more common senses). This conforms with our intuition of how word senses are distributed: a few common senses with a gradual transition to a long tail of rare senses. When deciding which cluster to use for a given predicate-argument pair, we use the cluster most associated with the argument.

We use a 10-token window around the predicate as features. The result of predicate induction on the verb *fire* is shown in Table 4. The first three clusters can be interpreted to be about (1) firing weapons, (2) sport-related shots (e.g., "*fired the puck*"), and (3) lay-offs. One must be careful in choosing the parameters for induction, however, as it is possible to partition a unique word sense such that coercions and selections are placed in a separate clusters. Section 6 discusses our parameter selection experiments.

5 Logical Metonymy Interpretation

Logical metonymies are a unique class of coercions due to the fact that their eventive interpretation can be derived from verbal predicates. For instance, for the logical metonymy "*enjoy book*", we know that *read* is a good candidate interpretation because (1) books are objects whose purpose is to be read and (2) reading is an event that may be enjoyed. We therefore expect to see many instances of both "*read book*" and "*enjoy reading*" (Lapata and Lascarides, 2003). Conversely, for coercions with non-eventive interpretations, such as "*arrive at meeting*", the interpretation (*location of*) is more dependent on the predicate (*arrive*) than the function of its argument (*meeting*).

In this section, we limit our discussion of logical metonymy to the verb-object case, its corresponding baseline for ranking interpretations, and our proposed enhancements. However, similar baselines exist for other types of logical metonymy, such as adjective-noun and noun-noun. Since our enhancement does not depend on any syntactic information beyond the predicate-argument instances needed for Section 3.2, it could easily be applied to those as well.

Lapata and Lascarides (2003) propose a probabilistic ranking model where the probability of an interpretation e for a verb-object pair (v, o) is proportional to the probability of all three in a verb-interpretation-object pattern.[3] For example, the probability that *read* is the correct interpretation of "*enjoy book*" is proportional to the likelihood of seeing "*enjoy reading book*" expressed as a syntactic dependency in a sufficiently large corpus. Due to data sparsity, they approximate this likelihood of seeing the object given the verb and interpretation to simply the likelihood of seeing the object given the interpretation. We denote this logical metonymy ranking method as LM_{LL}, formally defined as:

$$\begin{aligned} LM_{LL}(e; v, o) &= P_c(v, e, o) \\ &= P_c(e)P_c(v|e)P_c(o|e, v) \\ &\approx P_c(e)P_c(v|e)P_c(o|e) \\ &\approx \frac{f_c(v, e)f_c(o, e)}{N f_c(e)} \end{aligned} \quad (7)$$

Where P_c and f_c indicate probability and frequency, respectively, derived from corpus counts. See Lapata and Lascarides (2003) for a detailed explanation of how these frequencies are obtained.

This model, which we consider our baseline, is only partially correct as the corpus will contain coercions that form invalid interpretations. Consider the phrases "*enjoy finishing a book*" and "*enjoy discussing a book*". Both "*finish book*" and "*discuss book*" are coercions (and logical metonymies) themselves, and do not form a valid interpretation.[4]

[3] They use two patterns: "*v e*-ing *o*" and "v to e *o*", where e is tagged as a verb.

[4] For evidence of the frequency of these phrases, at the time of this writing, "*enjoy finishing a book*" and "*enjoy finishing the book*" have a combined 728 Google hits, while "*enjoy discussing a book*" and "*enjoy discussing the book*" have a com-

Thus, when discovering interpretations for logical metonymies, we must be aware of the selectional restrictions of candidate interpretations.

We propose to incorporate the coercion probability learned by our cLDA model in order to rank only those interpretations that are considered selections:

$$LM'(e; v, o) = P(v, e, o, x_o^e = 0) \quad (8)$$

However, due to the approximations made to estimate $P_c(v, e, o)$, this probability cannot be directly calculated as not all the frequencies reflect verb-object counts. Instead, we can combine the corpus probability $P_c(v, e, o)$ with the probability that the verb-object pair (e, o) is a coercion in our model. We denote this probability $P_x(e, o)$, and it may be derived from the scoring metrics in Equations (4), (5), or (6) above. We further propose three methods for enhancing the LM_{LL} baseline using $P_x(e, o)$ to approximate Equation 8.

A naive method for including information from our cLDA model is to consider the corpus probability, $P_c(v, e, o)$ and the coercion probability, $P_x(e, o)$, to be independent:

$$LM_{IND}(e; v, o) = P_c(v, e, o)P_x(e, o) \quad (9)$$

In other words, the rank of an interpretation is dictated by the unweighted combination of its corpus probability P_c and its coercion probability P_x. However, these two quantities are not likely to be independent. Most instances where e is used with either v or o are in fact selective.[5] We therefore experiment with two shallow learning methods for combining these two quantities.

The first method is a filtering approach where a threshold is learned for P_x:

$$LM_{TH}(e; v, o) = \begin{cases} P_c(v, e, o) & \text{if } P_x(e, o) \geq \delta \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Where the threshold δ is learned from a development set. We expect this model could suffer from noisy P_x values or to simply choose a threshold of zero due to the prominence of P_c.

Finally, we include a weighted linear model to discover the relative value of P_c and P_x:

$$LM_{WT}(e; v, o) = w_1 P_c(v, e, o) + w_2 P_x(e, o) \quad (11)$$

Where w_1 and w_2 are learned weights. We discuss how the parameters for LM_{TH} and LM_{WT} are learned in the experimental setup below.

6 Experimental Setup

We use the NYT subsection of the English Gigaword Fourth Edition (Parker et al., 2009) for a total of 1.8M newswire articles. The Stanford Dependency Parser (de Marneffe et al., 2006) is used to extract verb-object relations (*dobj*) that form the input to our model. To reduce noise, we keep only verbs listed in VerbNet (Kipper et al., 1998) with at least 100 argument instances, discarding *have* and *say*, which are too semantically flexible to select from clear semantic classes and so common they distort the class distributions. This results in 4,145 unique verbs with 51M argument instances (388K unique arguments). Additionally we use the dependency parser to extract open clausal complements of verbs (e.g., "*like to swim*") for use in logical metonymy interpretation. We believe this to be a more reliable alternative to the phrase chunk extraction patterns used in Lapata and Lascarides (2003). We keep clausal complements (*xcomp*) where the dependent is either a gerund or infinitive in order to estimate $P_c(v|e)$ in Equation (7).

For tiered clustering we use the same implementation as Reisinger and Mooney (2010)[6] to partition the surface form of the verb into one or more induced forms. Instead of using a fixed number of iterations, the clustering was run for 100 iterations past the best recorded log-likelihood in order to find the best possible fit to the data. We tuned the hyperparameters by maximizing the log-likelihood on a small held-out set of 20 predicate-argument pairs (10 selections, 10 coercions). The resulting partitions were fairly conservative, yielding 12,332 induced verbs or about 3 induced verb forms for every surface form, with 305 verbs not being partitioned at all.

We implemented both LDA and cLDA as described in Sections 3.1 and 3.2. For the α and β

bined 7,040 Google hits.

[5] For comparison, "*enjoy reading a book*" and "*enjoy reading the book*" have a combined 6.5 million Google hits

[6] Available at http://github.com/joeraii/UTML-Latent-Variable-Modeling-Toolkit

hyper-parameters, we used the MALLET (McCallum, 2002) defaults of 1.0 and 0.1, respectively, for both LDA and cLDA. We used the 20 predicate-argument pairs mentioned above to tune the γ hyper-parameters as well as the number of iterations. Both γ_0 and γ_1 were set to 100. We observed that for both LDA and cLDA, longer runs (in iterations) resulted in improved model log-likelihood but inferior results in terms of detecting coercions. It is not uncommon in topic modeling for model likelihood to not be completely correlated with the score on the task for which the topic model was intended (see Chang et al. (2009)). Both LDA and cLDA were found to perform best at 50 iterations on this data, after which their class distributions were less "smooth" and became rigidly associated with just a few classes, thus having a negative impact on coercion detection. While further iterations hurt coercion detection, only minor gains in model likelihood are seen. We believe the small number of iterations necessary for the model to converge is therefore a function of the data. In traditional topic modeling, documents are generally of similar size (i.e., within an order of magnitude). But in our data, many predicates have 10,000 times more instances than others. We have not yet empirically explored the impact of using a more uniform number of arguments for each predicate. This issue also makes it difficult to take multiple samples, which we experimented with unsuccessfully.

Our a priori intuition was that as the number of classes was increased, LDA would improve and cLDA would degrade due to its assumption of a single selected class. However, this did not always bear out in the results for every task described below. As such, instead of choosing a specific number of classes for each model, we describe results for each model with $K = 10$, 25, and 50.

For logical metonymy, both LM_{TH} and LM_{WT} require learned parameters. LM_{TH} needs a learned threshold while LM_{WT} needs two learned weights. For both, we split the data set into two partitions, learn the optimal threshold/weights on one partition, and use it as the parameters for the other partition. Both methods are trained on the final scoring metric, described in Section 7.3. For threshold learning, this involves finding the optimal cut-off to maximize the score. For weight learning, we use an exhaustive search over the range {1.0, 0.9, ..., 0.2, 0.1, 10^{-2}, 10^{-3}, ..., 10^{-14}} for both w_1 and w_2.

induced predicates?		N			Y		
# classes		10	25	50	10	25	50
LDA	NMI	.382	.448	.389	.435	.391	.383
	Rand	.717	.731	.721	.760	.723	.730
	F1	.425	.319	.192	.543	.311	.205
	B^3 (C)	.553	.513	.444	.525	.476	.341
	B^3 (E)	.453	.351	.223	.521	.324	.234
	MUC	.545	.545	.531	.500	.532	.544
cLDA	NMI	.446	.403	.360	**.510**	.430	.366
	Rand	.736	.719	.716	**.788**	.734	.711
	F1	.448	.291	.183	**.567**	.329	.184
	B^3 (C)	.575	.484	.312	**.593**	.495	.313
	B^3 (E)	.473	.321	.205	**.556**	.346	.205
	MUC	.500	.521	.507	**.595**	.541	.571

Table 2: Clustering scores for induced classes.

7 Results and Discussion

7.1 Semantic Class Induction

For the evaluation of the argument classes induced by our method, we use a subset of the WordNet lexicographer files, which correspond to coarse-grained semantic classes. We chose this form of evaluation because, unlike a named entity corpus, no sentential context is required and is therefore more consistent with the information available to our model. We use six of the larger, more semantically coherent WordNet classes: artifact, person, plant, animal, location, and food. We consider each of these a cluster and compare them to clusters composed of the top ten non-polysemous words (according to WordNet) in each of the classes generated by both the baseline (LDA) and our model (cLDA). Words not in both sets of clusters are removed. The result of this evaluation, compared with six clustering metrics, is shown in Table 2. For descriptions of NMI, Rand, and cluster F-measure, see Manning et al. (2008); for the B^3 metrics (Cluster and Element), see Bagga and Baldwin (1998); for the MUC metric, see Vilain et al. (1995). Each metric has different strengths and biases in regards to the number and distribution of clusters, so all are provided to give a general picture of class induction performance.

The best performing model on all metrics is cLDA with induced predicates using 10 classes. However, as the number of classes is increased and the granularity of the induced classes becomes more fine-grained, LDA (predictably) outperforms cLDA on most metrics. This is consistent with our intuition

induced predicates?		N			Y		
# classes		10	25	50	10	25	50
LDA	C_1	74.4	78.7	80.5	69.7	70.1	73.4
cLDA	C_1	80.6	**81.2**	80.9	76.2	78.4	77.5
	C_2	75.4	75.9	78.9	73.5	68.3	**80.8**
	C_3	67.8	70.8	67.4	70.9	67.4	74.1

Table 3: Accuracy on SemEval-2010 Task 7 data.

that a single-class assumption degrades as the number of classes increases.

For this evaluation, predicate induction also improved LDA for smaller numbers of classes, but not to the degree that it improved cLDA. Without predicate induction, LDA outperforms cLDA on all six metrics for 25 and 50 classes. With predicate induction, LDA outperforms cLDA on only one metric for 25 classes and five metrics for 50 classes. Thus the induced predicates do reduce the negative impact caused by the single selected class assumption for semantic class induction.

7.2 Coercion Detection

For the evaluation of coercion detection, we use the SemEval-2010 Task 7 data (Pustejovsky et al., 2010). This data uses the most common sense for each of five predicates (*arrive*, *cancel*, *deny*, *finish*, and *hear*) with a total of 2,070 sentences annotated with the argument's source type (the argument's semantic class) and target type (the predicate's selected class for that argument). We ignore the actual argument classes and evaluate on the coercion type, which is a selection when the source and target type match, and a coercion otherwise.

In order to evaluate unsupervised systems on this data, we use the corresponding training set (1,031 examples) to learn a threshold for coercion detection. At test time, if the model output is below the threshold, a coercion is inferred. Otherwise it is considered a selection. Therefore, the better a model can rank selections over coercions, the more accurate threshold it will learn. The results for this evaluation are shown in Table 3. The baseline for this task (threshold = 0, or all selections) is 67.4.

The best overall model on this data is cLDA using the C_1 coercion scoring method (Equation (4)). This method consistently outperforms the baseline LDA, especially for smaller numbers of classes, performing best with $K = 25$. The second metric, C_2, was not as reliable. The third metric, C_3, performed poorly on the task. As discussed in Section 3.3, C_3 is a direct result of the sampling for the predicate-argument pair in question and can thus be expected to perform poorly on rare predicate-argument pairs. Given that many of the arguments in this data are rare or unseen in the Gigaword data (e.g., "*cancel Renault*"), C_3's poor performance is understandable.

The use of predicate sense induction based on tiered clustering to overcome the single-class assumption caused significant degradation in performance on this task. Using automatically induced predicates instead of the surface form caused an average degradation of 2.6 points across the twelve tests. A potential explanation for this is that the evaluated predicates have a single dominant sense, meaning the single class assumption may be valid for these predicates (the task-defined selected classes are: location for *arrive*, event for *cancel* and *finish*, proposition for *deny*, and sound for *hear*). Therefore it would be interesting to evaluate it on a set of highly polysemous predicates with multiple dominant senses. Furthermore, the introduction of predicate sense induction was designed to help cLDA, and the performance degradation for these nine tests was not as large as it was for LDA. For cLDA, C_1 had an average degradation of 3.5 points compared to LDA's C_1 average degradation of 6.5 points. cLDA's C_2 had an average degradation of only 2.5 points and C_3 was actually improved by 2.1 points. This suggests that there is value in assigning different selected classes via sense induction, but that the two-step approach is not beneficial for these common predicates. This could be overcome by a joint approach of inducing predicate classes while simultaneously detecting coercions, as the presence of many coercions would be an indicator that more induced predicates are necessary.

7.3 Logical Metonymy Interpretation

For the evaluation of logical metonymy, we use both an existing data set and a newly created data set. Shutova and Teufel (2009) annotated 10 verb-object logical metonymies from Lapata and Lascarides (2003) with sense-disambiguated interpretations and organized the interpretations into clusters representing different possible meanings. For evaluation purposes we ignore the sense annotations and clusters and consider all lexical matchings of one of the annotated interpretations to be correct. The

	induced predicates?		N			Y		
	# classes		10	25	50	10	25	50
LM_{LL}			0.381			0.365		
LM_{IND}	LDA	C_1	0.415	0.406	0.383	0.386	0.412	0.395
	cLDA	C_1	0.408	0.412	0.412	0.407	**0.468**	0.439
		C_2	0.415	0.447	0.419	0.414	0.415	0.434
		C_3	0.416	0.453	**0.455**	0.395	0.416	0.402
LM_{TH}	LDA	C_1	0.599	0.568	0.588	0.479	0.520	0.551
	cLDA	C_1	0.571	0.644	0.751	0.497	0.620	**0.708**
		C_2	0.544	0.496	0.633	0.457	0.635	0.660
		C_3	0.601	0.677	**0.767**	0.472	0.622	0.571
LM_{WT}	LDA	C_1	0.383	0.381	0.379	0.365	0.356	0.361
	cLDA	C_1	0.380	0.387	0.381	0.386	0.377	0.321
		C_2	0.317	0.342	0.350	0.338	0.340	0.345
		C_3	0.378	0.370	0.350	0.387	0.382	0.384

Table 4: Mean average precision (MAP) scores on the Shutova and Teufel (2009) data set. The bold items indicate the best scores with/without induced predicates as well as using/not using a threshold-based interpretation method.

	induced predicates?		N			Y		
	# classes		10	25	50	10	25	50
LM_{LL}			0.274			0.248		
LM_{IND}	LDA	C_1	0.291	0.286	0.294	0.263	0.267	0.255
	cLDA	C_1	0.296	0.298	0.285	0.280	0.274	0.288
		C_2	0.291	0.287	0.288	0.283	0.271	0.285
		C_3	0.318	0.317	**0.333**	0.298	0.285	**0.307**
LM_{TH}	LDA	C_1	0.478	0.534	0.534	0.414	0.495	0.479
	cLDA	C_1	0.449	0.504	0.541	0.391	0.495	**0.513**
		C_2	0.505	0.478	0.456	0.398	0.429	0.440
		C_3	0.449	0.496	**0.577**	0.382	0.439	0.446
LM_{WT}	LDA	C_1	0.276	0.270	0.271	0.248	0.251	0.249
	cLDA	C_1	0.271	0.272	0.270	0.257	0.259	0.265
		C_2	0.274	0.274	0.266	0.250	0.259	0.261
		C_3	0.271	0.273	0.274	0.253	0.262	0.259

Table 5: Mean average precision (MAP) scores on 100 logical metonymies manually annotated with interpretations. The bold items indicate the best scores with/without induced predicates as well as using/not using a threshold-based interpretation method.

data contains an average of 11 interpretations per metonymy and has a reported 70% recall.

In order to create a larger data set, we identified 100 verb-object logical metonymies, including those used in Lapata and Lascarides (2003). Three annotators were asked to provide up to five interpretations for each metonymy (they were not provided with any verbs from which to choose, only the verb-object pair). The annotators provided an average of 4.6 interpretations per metonymy. Because our goal was recall, inter-annotator agreement was necessarily low, and each logical metonymy had an average of 11.7 unique interpretations. All annotators agreed on at least one interpretation for 40 metonymies, while for 14 they had no interpretations in common.[7]

Since logical metonymy interpretation is usually evaluated as a ranking task, we score our methods using mean average precision (MAP):

$$\text{MAP} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{\sum_{n=1}^{N}\text{prec}(n)\times\text{rel}(n)}{\text{interps}(q)} \qquad (12)$$

Where Q is the number of metonymies evaluated; N is the number of interpretations ranked; prec(n) is the precision at rank n; rel(n) = 1 if interpretation n is valid, 0 otherwise; and interps(q) is the number of valid interpretations for the metonymy q. We rank all 4,145 verbs as interpretations except for those removed by the threshold technique, as they have a score of zero. This can give LM_{TH} artificially high MAP scores since it may remove some valid interpretations that are low-ranking. However, since a smaller, higher precision list may be useful for many applications we still consider MAP a valid metric and indicate both the highest scoring method and the highest scoring non-threshold method. The results on the Shutova and Teufel (2009) data are

[7] Data available at http://www.hlt.utdallas.edu/∼kirk/data/lmet.zip

shown in Table 4. The results on our own data are shown in Table 5.

The scores reported in the Shutova and Teufel (2009) data are noticeably higher than the data we annotated. Since the metonymies in our data are a super-set of those in their data, and since for those metonymies our annotators provided approximately the same number of interpretations (110 versus 120), this likely indicates the remaining metonymies in our data are more difficult.

In all cases the best reported scores use cLDA. Unlike coercion detection on the SemEval data, C_3 performs very well, achieving the highest scores when no predicate sense induction is used. Also unlike coercion detection, LDA scores do not increase as the number of classes increase. We suspect both these differences have to do with the fact that the arguments in this data are far more common. Since LDA is a selectional preference model and its coercion scores correspond roughly to the plausibility of seeing a predicate-argument pair, it is less able to distinguish coercions in common arguments.

Of the logical metonymy ranking methods, LM_{TH} consistently produces the highest MAP scores. However, as stated before, by using a cut-off and removing low-ranking valid interpretations, the MAP score is increased, which might not be applicable to some applications. The best non-thresholded ranking method is LM_{IND}, which naively combines the LM_{LL} score with the coercion probability. In almost every case this beats out LM_{WT}. Upon inspection, we observed that the range of scale for the LM_{LL} scores are very inconsistent. This can make it difficult to learn a linear model using these scores as features, and as a result the learned weights were forced to ignore the coercion score and rely entirely on LM_{LL}. We attempted other scaling methods, such as a rank-based method, but these had poor results as well, so we leave the problem of the supervised learning these weights to future work.

Using induced senses did not result in the drastic and consistent degradation in performance seen on the SemEval data, and the highest non-threshold result for the Shutova and Teufel (2009) data used predicate induction. Both metonymy data sets were limited to the verbs found in Lapata and Lascarides (2003), which are still quite common (*attempt*, *begin*, *enjoy*, *expect*, *finish*, *prefer*, *start*, *survive*, *try*, *want*). However, the verbs used in our data set had a greater number of WordNet senses attested in a corpus than the SemEval data (an average of 4.4 senses for our data versus 3.0 senses for the SemEval data). This suggests the potential value of sense induction for highly polysemous predicates and further motivates the integration of sense induction within a selectional restriction model.

8 Conclusion

We have presented a novel topic model that extends an unsupervised selectional preference model (LDA) to an unsupervised selectional restriction model (cLDA) using two assumptions. For the first assumption, that each predicate has a single selected class, we proposed a predicate induction method to overcome predicate polysemy. This improved results for semantic class induction but proved harmful for detecting coercions on common predicates with a single, dominant sense. For the second assumption, that the selected class can be inferred from the data, we proposed a sampling method based on the classes of the predicate's arguments. Superior performance on coercion detection shows the merit of this assumption.

Additionally, we proposed methods for improving an existing task, logical metonymy interpretation, using the learned parameters of our model, showing positive results.

It is clear that our model may be improved by more accurate predicate sense induction. To this end, we plan to develop a model that simultaneously induces predicates and learns coercions, using knowledge of a predicate's coerciveness to inform the induction mechanism.

Acknowledgements

We would like to thank Diarmuid Ó Séaghdha, Bryan Rink, and Anna Rumshisky for several helpful conversations during the course of this work. We thank Mirella Lapata and Ekaterina Shutova for making the data from their experiments available as well as the organizers of SemEval-2010 Task 7 for the associated data set. Additionally, we thank Srikanth Gullapalli, Aileen McDermott, and Bryan Rink for annotating the data set used in our experiment. Finally, we thank the anonymous reviewers for their suggestions on improving this work.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *In Proceedings of the First International Conference on Language Resources and Evaluation Workshop on Linguistic Coreference*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Samuel Brody and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–111.

Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, pages 1–9.

Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. 2007. Modeling General and Specific Aspects of Documents with a Probabilistic Topic Model. In *Advances in Neural Information Processing Systems 19*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the Fifth International Language Resources and Evaluation*.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 1998. Class-based construction of a verb lexicon. In *Proceedings of AAAI/IAAI*.

Maria Lapata and Alex Lascarides. 2003. A Probabilistic Account of Logical Metonymy. *Computational Linguistics*, 21(2):261–315.

Dekang Lin and Patrick Pantel. 2001. Induction of Semantic Classes from Natural Language Text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 317–322.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit.

Diarmuid Ó Séaghdha. 2010. Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 435–444.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2009. English Gigaword Fourth Edition. *The LDC Corpus Catalog.*, LDC2009T13.

James Pustejovsky, Anna Rumshisky, Alex Plotnick, Elisabetta Jezek, Olga Batiukova, and Valeria Quochi. 2010. SemEval-2010 Task 7: Argument Selection and Coercion. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 27–32.

Joseph Reisinger and Raymond J. Mooney. 2010. A Mixture Model with Sharing for Lexical Semantics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182.

Alan Ritter, Mausam, and Oren Etzioni. 2010. A Latent Dirichlet Allocation method for Selectional Preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 424–434.

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*.

Anna Rumshisky, Victor A. Grinberg, and James Pustejovsky. 2007. Detecting selectional behavior of complex types in text. In *Fourth International Workshop on Generative Approaches to the Lexicon*.

Ekaterina Shutova and Simone Teufel. 2009. Logical Metonymy: Discovering Classes of Meaning. In *Proceedings of the CogSci 2009 Workshop on Semantic Space Models*.

Ekaterina Shutova. 2009. Sense-based interpretation of logical metonymy using a statistical method. In *Proceedings of the ACL 2009 Student Workshop*.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings fo the 6th Message Understanding Conference*, pages 45–52.