Shared Components Topic Models with Application to Selectional Preference

Matthew R. Gormley **Mark Dredze** **Benjamin Van Durme** **Jason Eisner**
Center for Language and Speech Processing
Human Language Technology Center of Excellence
Department of Computer Science
Johns Hopkins University, Baltimore, MD
{mrg,mdredze,vandurme,jason}@cs.jhu.edu

Introduction Predicate argument *selectional preference*[1] is the notion that the roles, or argument positions, of a given predicate tend to prefer some arguments to others. Automatically inferring these preferences has been a topic of interest within the computational linguistics community since the early 1990's, with Resnik [3] giving examples such as: *Mary drank some* {*wine, gasoline, pencils, sadness*}, where the provided nouns in the syntactic object position of the verb *drink* are of various levels of semantic acceptability. Here we are motivated by statements such as the following by Resnik [3]:

> Although selectional preference is traditionally formalized in terms of feature agreement using notations like [+Animate], such formalizations often fail to specify the set of allowable features, or to capture the gradedness of qualitative difference [. . .],

Like Resnik, we would like a preference model that retained this notion of *feature agreement*, but also allowed for the *gradedness* that contemporary computational linguists have come to assume. A related intuition can be found in recent work on category learning of Griffiths and colleagues, such as [4], which is partially motivated by the notion of linguistic ontologies [5].

We introduce the Shared Components Topic Model (SCTM), which expresses selectional preferences as soft disjunctions of conjunctions of semantic features. The model assumes there exist underlying semantic features, e.g. `[+liquid]`, `[+solid]`, each of which defines a distribution over nouns. The model conjoins these features into semantic classes, such as ⟨`[+liquid]` & `[+comestible]`⟩; each class gives a distribution over nouns that all possess the same set of semantic features. In this way, we aim to model the acceptable nouns of *eat* as (⟨`[+comestible]` & `[+solid]`⟩ OR ⟨`[+meal]`⟩). In the SCTM, because the features are distributions, the classes are constructed as products (the soft variant of conjunctions) and preferences are encoded by mixtures (the soft variant of disjunctions).

Model Latent Dirichlet Allocation (LDA) [6] has been used to learn selectional preferences as soft disjunctions over flat semantic classes [7, 8, 9]. Our model, the SCTM, also learns the structure of each class as a soft conjunction of high-level semantic features. Figure 1a provides the generative process for topic modeling and a mapping of terminology to selectional preference. Here, we further describe the SCTM by analogy to LDA. In both LDA and the SCTM, each verb is modeled as a mixture over K semantic classes; for each noun token, this mixture generates a semantic class assignment, and the noun's type is sampled from that semantic class. In LDA, a semantic class is a multinomial over words sampled from a shared Dirichlet prior. By contrast, the SCTM models each semantic class as a normalized product of a subset of C underlying semantic features (which we call components).

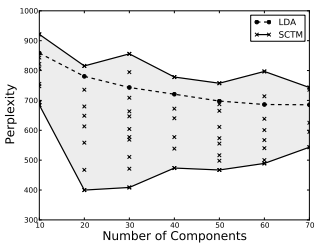
[1] The term selectional preference has a variety of (near) synonyms. Chomsky [1] used the term *selectional rules*, giving the alternatives: *selectional restrictions* and *restrictions of cooccurrence*. Semanticists such as Thomason [2] referred to the problem as *sortal (in)correctness*, with Thomason providing variants including: *category mistake*, *selectionally incorrect*, *type crossing*, *semi-grammatical*, and *semantically anomalous*.

Figure 1

For each component $c \in \{1, \ldots, C\}$:
  $\boldsymbol{\phi}_c \sim \text{Dir}(\boldsymbol{\beta})$
  $\pi_c \sim \text{Beta}(\frac{\gamma}{C}, 1)$
  For each topic $k \in \{1, \ldots, K\}$:
    $b_{kc} \sim \text{Bernoulli}(\pi_c)$
For each document $m \in \{1, \ldots, M\}$
  $\boldsymbol{\theta}_m \sim \text{Dir}(\boldsymbol{\alpha})$
  For each word $n \in \{1, \ldots, N_m\}$
    $z_{mn} \sim \text{Mult}(1, \boldsymbol{\theta}_m)$
    $x_{mn} \sim p(\cdot \,|b_{z_{mn}}, \boldsymbol{\phi})$ from Eq. (1)

Topic Modeling ↔Selectional Preference
document ↔verb, word ↔noun,
topic ↔class, component ↔feature

(a) The SCTM generative process



(b) Topic Modeling

model	C	K	PwA	Perp
CPM			89.99	194.34
LDA (K=C)	10		77.98	597.59
	11		78.10	583.42
	20		81.92	462.80
	21		82.51	460.32
	30		83.84	406.42
	40		85.28	363.80
SCTM	10	10	75.01	684.32
	10	20	78.91	543.80
	10	40	80.97	461.40
	10	80	81.90	432.35
	20	20	78.23	615.09
	20	40	83.04	423.67
	20	80	85.19	334.36
	20	160	86.52	283.39

(c) Selectional Preference

The kth semantic class in the SCTM is a Product of Experts (PoE) [10] model, where the subset of semantic features included in the product is determined by a binary vector generated by a beta-bernoulli model, the finite counterpart of the Indian Buffet Process (IBP) [11].

$$p(x|\boldsymbol{b}_k, \boldsymbol{\phi}) = \frac{\prod_{c=1}^{C} \phi_{cx}^{b_{kc}}}{\sum_{v=1}^{V} \prod_{c=1}^{C} \phi_{cv}^{b_{kc}}} \tag{1}$$

Here, $\boldsymbol{\phi}_c$ is the cth semantic feature, a distribution over words. $\boldsymbol{b}_k$ is the binary vector defining the structure of this semantic class. The model is closely related to SAGE [12] and the IOMM [13].

Learning To perform parameter estimation, we use an algorithm that follows the outline of the Monte Carlo EM (MCEM) algorithm [14]. In the Monte Carlo E-step, we sample the class assignments z_{mn} and the binary vectors b_{kc} based on current parameters $\boldsymbol{\phi}$ and observed data X. In the M-step, we find new components $\boldsymbol{\phi}$. Since these are the parameters of the PoEs, we replace the usual maximization of data log-likelihood with a contrastive divergence (CD) objective [15], popular for PoE training. Normally, CD only estimates the parameters of the product distributions. However, in our model, which features are included in the product change based on the E-step. Since we can generate multiple samples in the E-step, we modify the CD objective to compute the gradient for each E-step sample and take the average to approximate the expectation under b_{kc} and z_{mn}. This approximate algorithm is much more efficient than a pure MCEM (or a pure MCMC) approach.

Experiments We present results on two tasks: selectional preference and topic modeling. For selectional preference, our data comes from the part-of-speech (POS) tagged n-grams corpus of [16]. Using POS tag patterns we produce a corpus of selectional preference examples of the form $(\text{verb}_{\text{dependency type}}, \text{noun})$. For example, the pattern VBD (PRP\$|DT) NN would match *sold the car*. Figure 1c presents pseudo-word accuracy (PwA) and per-noun perplexity (Perp) on test data. We consider the tradeoff of model compactness vs. accuracy. In the case where C=K=10, the performance of LDA is better than the SCTM in both PwA and Perp. Yet, if we increase the size of both models and consider the case of LDA with K=11 and of the SCTM with C=10, K=40, then the PwA of the former is 78.10 and the latter is 80.97. Yet, LDA has added a multinomial the size of the vocabulary (1000) while the SCTM has added only a few binary vectors of length C.

We also apply the SCTM to topic modeling to explore its potential as a more compact representation of topics. We use 1,000 randomly selected articles from the 20 Newsgroups dataset.[2] We evaluate the average perplexity per word on held out test data using the *left-to-right* approximation of [17]. Figure 1b shows the results for LDA and the SCTM for the same number of components and varying K (SCTM). For LDA (K=C), this is a single (dashed) line. For SCTM, the x markers each correspond to a different K at the shown C. The shaded region shows the full SCTM perplexity range for different K. Observe that for each number of components, LDA falls within the upper portion of the shaded region. This shows that, while for some (small) values of K for the SCTM, LDA does better, the SCTM can easily include more K (requiring few new parameters) to achieve better results.

Discussion The main goal of this work is to learn features which can compactly describe selectional preference and the ways in which they combine into semantic classes. While quantitatively our model performs as well as and learns classes (topics) that are similar in appearance to LDA, our current work is focussed on biasing the SCTM to prefer more interpretable underlying features (components).

[2] http://people.csail.mit.edu/jrennie/20Newsgroups/

References

[1] Noam Chomsky. *Aspects of the Theory of Syntax*. 1965.

[2] R. H. Thomason. A semantic theory of sortal incorrectness. *Journal of Philosophical Logic*, 1:209–258, 1972.

[3] Philip Resnik. Semantic classes and syntactic ambiguity. In *Proceedings of the workshop on Human Language Technology*, pages 278–283, 1993.

[4] K.R. Canini and T.L. Griffiths. A nonparametric bayesian model of Multi-Level category learning. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence.*, 2011.

[5] F.C. Keil. *Semantic and conceptual development: An ontological perspective*. Harvard University Press, Cambridge, MA, 1979.

[6] David Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[7] Benjamin Van Durme and Daniel Gildea. Topic models for corpus-centric knowledge generalization. *Technical Report*, (TR-946), June 2009.

[8] Alan Ritter, Mausam, and Oren Etzioni. A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

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

[10] Geoffrey Hinton. Products of experts. In *International Conference on Artificial Neural Networks (ICANN)*, 1999.

[11] Thomas Griffiths and Zoubin Ghahramani. Infinite latent feature models and the indian buffet process. In *Advances in Neural Information Processing Systems (NIPS)*, volume 18, 2006.

[12] Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. Sparse additive generative models of text. In *International Conference on Machine Learning (ICML)*, 2011.

[13] K.A. Heller and Z. Ghahramani. A nonparametric bayesian approach to modeling overlapping clusters. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS07)*, page 187194, 2007.

[14] Greg Wei and Martin Tanner. A monte carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

[15] Geoffrey Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[16] Dekang Lin, Kenneth Church, Heng Ji, Satoshi Sekine, David Yarowsky, Shane Bergsma, Kailash Patil, Emily Pitler, Rachel Lathbury, Vikram Rao, Kapil Dalwani, and Sushant Narsale. New tools for web-scale n-grams. In *Language Resources and Evaluation (LREC)*, 2010.

[17] Hanna Wallach, Ian Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *International Conference on Machine Learning (ICML)*, pages 1105–1112, 2009.