Adapting Topic Models using Lexical Associations with Tree Priors

Weiwei Yang  
Computer Science (CS)  
University of Maryland  
College Park, MD  
wwyang@cs.umd.edu

Jordan Boyd-Graber  
CS, iSchool, LSC, and UMIACS  
University of Maryland  
College Park, MD  
jbg@umiacs.umd.edu

Philip Resnik  
Linguistics and UMIACS  
University of Maryland  
College Park, MD  
resnik@umd.edu

Abstract

Models work best when they are optimized taking into account the evaluation criteria that people care about. For topic models, people often care about interpretability, which can be approximated using measures of lexical association. We integrate lexical association into topic optimization using tree priors, which provide a flexible framework that can take advantage of both first-order word associations and the higher-order associations captured by word embeddings. Tree priors improve topic interpretability without hurting extrinsic performance.

1 Introduction

Goodman (1996) introduces a key insight for machine learning models in natural language processing: if you know how performance on a problem is evaluated, it makes more sense to optimize using that evaluation metric, rather than others. Goodman applies his insight to parsing algorithms, but this insight has had an even larger impact in machine translation, where the introduction of the fully automatic BLEU metric makes it possible to tune systems using a score correlated with human rankings of MT system performance (Papineni et al., 2002).

Chang et al. (2009) provide a similar insight for topic models (Blei et al., 2003, LDA): if what you care about is the interpretability of topics, the standard objective function for parameter inference (likelihood) is not only poorly correlated with a human-centered measurement of topic coherence, but inversely correlated. Nonetheless, most topic models are still trained using methods that optimize likelihood (McAuliffe and Blei, 2008; Nguyen et al., 2013).

We take the logical next step suggested when you bring together the insights of Goodman (1996) and Chang et al. (2009), namely incorporating an approximation of human topic interpretability into the topic model optimization process in a way that is effective and more straightforward than previous methods (Newman et al., 2011). We take advantage of the human-centered evaluation of Chang et al. (2009), which can be reasonably approximated using an automatic metric based on word associations derived from a large, more general corpus (Lau et al., 2014). We exploit LDA and its Bayesian formulation by bringing word associations into the picture using a prior—specifically, we use external lexical association to create a tree structure and then use tree LDA (Boyd-Graber et al., 2007, tLDA), which derives topics using a given tree prior.

We construct tree priors with combinations of two types of word association scores (skip-gram probability (Mikolov et al., 2013) and G2 likelihood ratio (Dunning, 1993)) and three construction algorithms (two-level, hierarchical clustering with and without leaf duplication). Then tLDA identifies topics with these tree priors in Amazon reviews and the 20NewsGroups datasets. tLDA topics are more coherent compared with “vanilla” LDA topics, while retaining and often slightly improving topics’ extrinsic performance as features for supervised classification. Our approach can be viewed as a form of adaptation, and the flexibility of the tree prior approach—a menable to any kind of association score—suggests that there are many directions to pursue beyond the two flavors of association explored here.
Figure 1: An example of a tree prior (the tree structure) and gold posterior edge and word probabilities learned by tLDA. Numbers beside the edges denote the probability of moving from the parent node to the child node. A word’s probability, i.e., the number below the word, is the product of probabilities moving from the root to the leaf node.

2 Tree LDA: LDA with Tree Priors

Tree priors organize the vocabulary of a dataset in a tree structure, contrasting with introducing topic correlations (Blei and Lafferty, 2007; He et al., 2017). Words are located at the leaf level and share ancestor internal nodes. In our use of tree priors, if two words have a lower association score, their common ancestor node will be closer to the root node, e.g., contrast (orbit, satellite) with (orbit, launch) in Figure 1.

Tree LDA (Boyd-Graber et al., 2007, tLDA) is an LDA extension that creates topics from a tree prior. A topic in tLDA is a multinomial distribution over the paths from the root to leaves. An internal node, i.e., the circles in Figure 1, is a multinomial distribution over its child nodes. The probability of a path is the product of probabilities of picking the nodes in the path, e.g., \( P_r(\text{satellite}) = 0.614 \times 0.962 \times 0.427 \approx 0.252 \). Thus two paths with shared nodes have correlated weights in a topic. The generative process of tLDA is:

1. For topics \( k \in \{1, \ldots, K\} \) and internal nodes \( n_i \)
   - (a) Draw child distribution \( \pi_{k,i} \sim \text{Dir}(\beta) \)
2. For each document \( d \in \{1, \ldots, D\} \)
   - (a) Draw topic distribution \( \theta_d \sim \text{Dir}(\alpha) \)
   - (b) For each token \( t_{d,n} \) in document \( d \)
     - i. Draw topic assignment \( z_{d,n} \sim \text{Mult}(\theta_d) \)
     - ii. Draw path \( y_{d,n} \) to word \( w_{d,n} \) with probability \( \prod_{(z,i) \in y_{d,n}} \pi_{z_{d,n}, i} \)

The root node is not considered a level.

All tree prior examples are real sub-trees of the priors built on Gigaword.

3 Tree Prior Construction from Word Association Scores

A two-level tree is the most straightforward construction.\(^2\) Each internal node, \( n_i \), is a concept associated with a word \( v_i \) in the vocabulary. Then we sort all other words in descending order of their association scores with \( v_i \) and select the top \( N \) words (we use \( N = 10 \)) as \( n_i \)’s child leaf nodes. \( n_i \) has an additional child node which represents \( v_i \), to ensure that every word appears at the leaf level at least once (Figure 2).\(^3\) Thus, if the vocabulary size is \( V \), there will be a total of \( (N + 1)V \) leaf nodes.

3.1 Hierarchical Clustering (HAC)

While a two-level tree is bushy (high branching factor) and flat, hierarchical agglomerative clustering (Lukasová, 1979, HAC) reduces the number of leaf nodes and encodes levels of word association information in its hierarchy (Figure 1).

The HAC process starts from \( V \) clusters representing the \( V \) words in the vocabulary. It then repeatedly merges the two clusters with the highest association score until there is only one cluster left. If at least one of the two clusters, \( c_i \) and \( c_j \), has multiple words, their association score is the average association score of the pairwise words from the two clusters:

\[
S(c_i, c_j) = \frac{1}{|c_i||c_j|} \sum_{w_j \in c_i} \sum_{w_j' \in c_j} S(w_j, w_j').
\]

3.2 HAC with Leaf Duplication (HAC+LD)

HAC might merge words with multiple senses. For example, the word “spring” could mean either a season (similar to “summer”) or a place with water

\(^1\)Unlike other tree-based topic models such as Andrzejewski et al. (2009), all Dirichlet hyperparameters are the same for all internal nodes. Regardless of cardinality, all Dirichlet parameters are the same scalar \( \beta \).

\(^2\)The root node is not considered a level.

\(^3\)All tree prior examples are real sub-trees of the priors built on Gigaword.
Figure 3: An example of HAC-LD for the words “spring”, “summer”, and “lake”, whose paired words are shaded in gray. HAC-LD alleviates the problem in HAC that a word with multiple senses can only be assigned to a single cluster close to one of its senses.

(similar to “lake”). Assigning “spring” to either side will cause information loss on the other side.

To alleviate this problem, we first pair every word with its most similar word and create a cluster with the pair. Thus “spring” is paired with “summer” and “lake” simultaneously (Figure 3).

4 Experiments

We compute two versions of word association scores from Gigaword, using word2vec (Mikolov et al., 2013) and G2 likelihood ratio (Dunning, 1993).\footnote{https://catalog.ldc.upenn.edu/ldc2011t07.} Given the word vectors \( v_i \) and \( v_j \), which represent words \( w_i \) and \( w_j \), their word2vec association score is

\[
S(w_i, w_j) = \frac{\exp (v_i \cdot v_j)}{\sum_k \exp (v_i \cdot v_k)}. \tag{2}
\]

Then we apply the three tree construction algorithms to construct six tree priors. In the two-level trees, the value of \( N \), i.e., the number of child nodes per internal node, is ten.

We use Amazon reviews (Jindal and Liu, 2008) and 20NewsGroups (Lang, 1995, 20NG). We apply the same tokenization and stopword removal methods. We then sort the words by their document frequencies and return the top words, while also removing words that appear in more than 30% of the documents (Table 1).

Both corpora are split into five folds. For classification tasks, each fold is further equally split into a development set and a test set. All the results are averaged across five-fold cross-validation using 20 topics with hyper-parameters \( \alpha = \beta = 0.01 \). For

<table>
<thead>
<tr>
<th>Corpus</th>
<th>#Vocabulary</th>
<th>#Docs</th>
<th>#Tokens</th>
<th>#Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>20NG</td>
<td>9,194</td>
<td>18,769</td>
<td>1.75M</td>
<td>20</td>
</tr>
<tr>
<td>Amazon</td>
<td>9,410</td>
<td>39,392</td>
<td>1.51M</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1: Corpus Statistics

Table 2: The average perplexity results on the test sets by various models. LDA gives the lowest perplexity, because tLDA models have constraint from the tree priors and sacrifice the perplexity.

20NewsGroups classification a post’s newsgroup is its label. For Amazon reviews, 4–5 star reviews have positive labels, 1–2 stars negative, and reviews with 3 stars are discarded.

4.1 Perplexity

Before evaluating topic quality, we conduct a sanity check of the models’ average perplexity on the test sets (Table 2).

LDA achieves the lowest perplexity among all models on both corpora while tLDA models yield suboptimal perplexity results owing to the constraints given by tree priors. As shown in the following sections, the sacrifice in perplexity brings improvement in topic coherence, while not hurting or slightly improving extrinsic performance using topics as features in supervised classification.

Tree priors built from word2vec generally outperform the ones built using the G2 likelihood ratio. Among the three tree prior construction algorithms, the two-level is the best on the 20NewsGroups corpus. However, there is no such consistent pattern on Amazon reviews.

4.2 Topic Coherence

Instead of manually evaluating topic quality using word intrusion (Chang et al., 2009), we use an automatic alternative to compute topic coherence (Lau et al., 2014). For every topic, we extract its top ten words and compute average pairwise PMI on a reference corpus (Wikipedia as of October 8, 2014).

We include LDA and the latent concept topic model (Hu and Tsujii, 2016, LCTM) as baselines. LCTM also incorporates prior knowledge from word embeddings. It assumes that latent concepts exist in the embedding space and are Gaussian distributions over word embeddings, and a topic is a multinomial distribution over these concepts. We
Table 3: We sort topics into thirds by Kullback-Leibler divergence (KLD): low, medium, and high divergence between vanilla LDA and tLDA. Unique coherent words are in **black and bold**. Unique incoherent words are in *red and italic*. tLDA brings in more topic-relevant words.

<table>
<thead>
<tr>
<th>Topic</th>
<th>KLD</th>
<th>Model</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Christian</td>
<td>0.709</td>
<td>LDA</td>
<td>god, jesus, church, christ, christian, bible, man, christians, lord, sin</td>
</tr>
<tr>
<td>Security</td>
<td>0.720</td>
<td>LDA</td>
<td>key, encryption, chip, clipper, keys, government, public, security, system, law</td>
</tr>
<tr>
<td>Middle East</td>
<td>0.765</td>
<td>tLDA</td>
<td>israel, jews, war, israeli, jewish, arab, people, world, peace, muslims</td>
</tr>
<tr>
<td>Sports</td>
<td>1.212</td>
<td>LDA</td>
<td>hockey, team, game, play, la, nhl, ca, period, pit, cup</td>
</tr>
<tr>
<td>University Research</td>
<td>1.647</td>
<td>LDA</td>
<td>university, information, national, april, states, year, research, number, united, american</td>
</tr>
<tr>
<td>Health</td>
<td>1.914</td>
<td>LDA</td>
<td>medical, people, disease, health, cancer, <em>food, sex, crammer, men, drug</em></td>
</tr>
<tr>
<td>Images</td>
<td>1.995</td>
<td>LDA</td>
<td>file, image, jpeg, graphics, files, format, <em>bit, color</em>, program</td>
</tr>
<tr>
<td>Hardware</td>
<td>2.127</td>
<td>tLDA</td>
<td>drive, card, mb, scsi, disk, <em>mac, system, pc, apple</em>, bit</td>
</tr>
<tr>
<td>People</td>
<td>2.512</td>
<td>LDA</td>
<td>armenian, people, turkish, armenians, <em>armenia, turkey, turks, didn</em>, soviet, <em>time</em></td>
</tr>
</tbody>
</table>

Most tLDA models yield more coherent topics (Figure 4). Among all tLDA models, the two-level tree built on word2vec improves the most. LCTM performs poorly: after marginalizing out the concepts on 20NewsGroups, all its topics consist of words like “don”, “dodgers”, “au”, “alot”, “people”, “alicea”, “uw”, “arabia”, “sps”, and “entry” with slight differences in ordering.

To show how subjective topic quality improves over LDA, we extract the topics given by LDA and tLDA (with two-level tree built on word2vec scores) on 20NewsGroups, pair them, and sort the pairs based on KL divergence (KLD). In Table 3, we select and present three topics from each of the top, middle, and bottom third of the sorted topics.

Topics with low KLD (Christian, Security, and Middle East) do not differ significantly. Although the topics of Sports have medium KLD and quite different words, they are generally coherent. As the KLD increases, tLDA topics have more coherent words. In University Research topics, tLDA includes more research-related words, e.g., “center”, “science”, and “institute”. In Health topics, the tLDA topic has more coherent words like “patients”, “insurance”, “aids”, and “treatment”, while LDA includes less relevant words, e.g., “food”, “sex”, and “cramer”.

In the topics with large KLD, tLDA topics are also more coherent. For instance, in the Images topics, the LDA topic contains less relevant words like “mail” and “data”, while the tLDA topic mostly consists of words related to images, and even includes words like “jpeg”, “color”, and “bit” that are not among the top words in the LDA topic. In the topics for Hardware, there are more words closer to the hardware level for tLDA, e.g., “drives”, “dos”, “controller”, and “ide”, in contrast to LDA, e.g., “mac”, “pc”, and “apple”. tLDA also ranks hardware-related words higher. For instance, “scsi” and “disk” come before “mb”. The words in the topics for People are generally coherent, except “didn” and “time” in the LDA topic.

### 4.3 Extrinsic Classification

To extrinsically evaluate topic quality, we use binary and multi-class classification on Amazon reviews and 20NewsGroups corpora using SVM-light (Joachims, 1998) and SVM-multiclass. We tune the parameter $C$, the trade-off between training error and margin, on the development set and apply the trained model with the best performance on the development set to the test set. The classification accuracies are given in Table 4.

We compare the accuracies of features of bag-of-words (BOW) and LDA/LCTM/tLDA topics. For the tLDA models with two-level and HAC-LD tree priors, the path assignment is an additional feature. We also include the features of BOW
Combining topic models and vector space models is an emerging area. We introduce a method that is simpler and more flexible than previous work (Hu and Tsujii, 2016), and although we extract prior knowledge from word vectors, our model is not restricted to this and can use any word association scores. Our model yields more coherent topics and maintains extrinsic performance, and in addition it is less computationally costly.\footnote{TLDA Java implementation converges in twelve hours; LCTM needs sixty hours (2.8GHz Intel Xeon and 110GB RAM).}

We plan to merge tree prior construction and the topic modeling into a unified framework (Teh et al., 2007; Görür and Teh, 2009; Hu et al., 2013). This will allow tree priors to change along with the topics they produce instead of using a static one constructed \textit{a priori}.
Acknowledgement

We thank the anonymous reviewers for their insightful and constructive comments. This research has been supported in part, under subcontract to Raytheon BBN Technologies, by DARPA award HR0011-15-C-0113. Boyd-Graber is also supported by NSF grants IIS-1320538, IIS-1409287, IIS-1564275, IIS-1652666, and NCSE-1422492. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsors.

References


