Learning a Concept Hierarchy from Multi-labeled Documents

Viet-An Nguyen[1], Jordan Boyd-Graber[2], Philip Resnik[1,3,4] and Jonathan Chang[5]

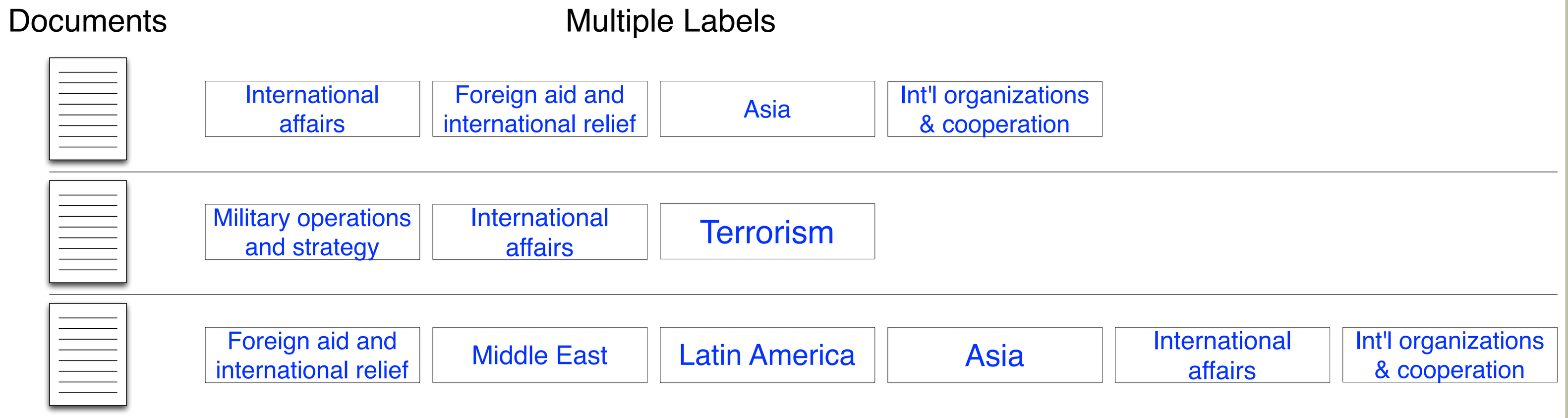
[1]Computer Science, [3]Linguistics, [4]UMIACS
University of Maryland, College Park, MD

[2]Computer Science
University of Colorado, Boulder, CO

[5]Facebook
Menlo Park, CA

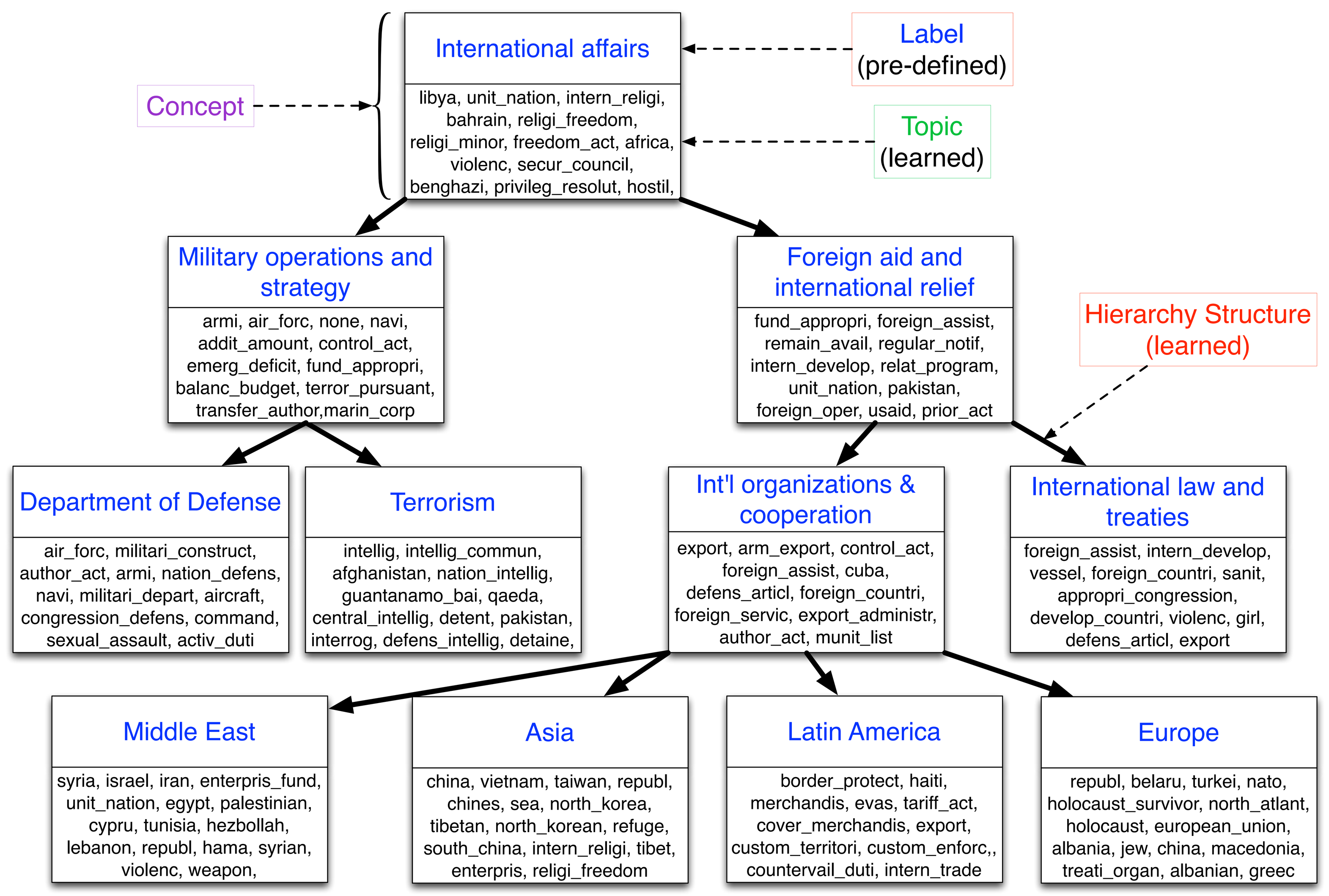
Input: Multi-labeled Documents

Each document is tagged with multiple labels.



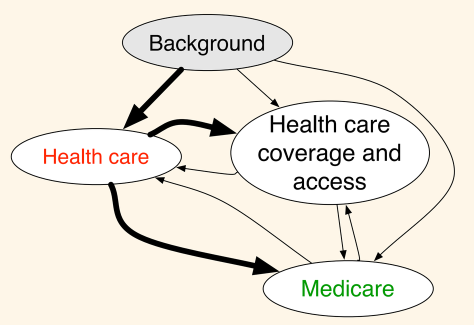
Output: Concept Hierarchy

1. Capturing the **dependency among labels**: using tree-structured hierarchy
2. Learning an **interpretable topic hierarchy**: associating
   - Label: pre-defined word/phrase
   - Topic: multinomial distribution over the vocabulary
   - Concept = Label + Topic



Label-to-Hierarchy (L2H): Learning an Interpretable Hierarchy from Multi-labeled Data

Creating label graph

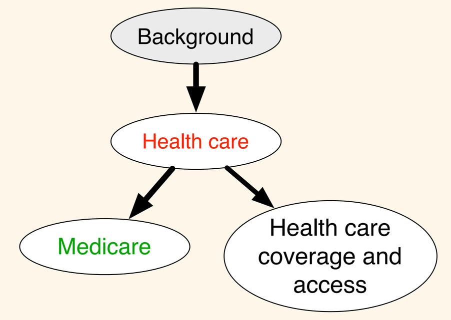


- Construct a complete weighted directed graph $\mathcal{G}$ where each label is a node and the edge weight is:

$$t_{\text{Health care}\rightarrow\text{Medicare}} = \frac{\text{No. docs tagged with both Health care \& Medicare}}{\text{No. docs tagged with Medicare}}$$

- Add a Background node to the graph

Generating tree-structured hierarchy



- Generate a spanning tree

$$p(\mathcal{T}\,|\,\mathcal{G}) = \prod_{\text{all edges }(i\rightarrow j)} t_{i\rightarrow j}$$

- Associate each node with a topic:

$$\begin{cases}\phi_{\text{Background}} \sim \text{Dir}(\beta u)\\ \phi_{\text{Medicare}} \sim \text{Dir}(\beta\phi_{\text{Health care}})\end{cases}$$

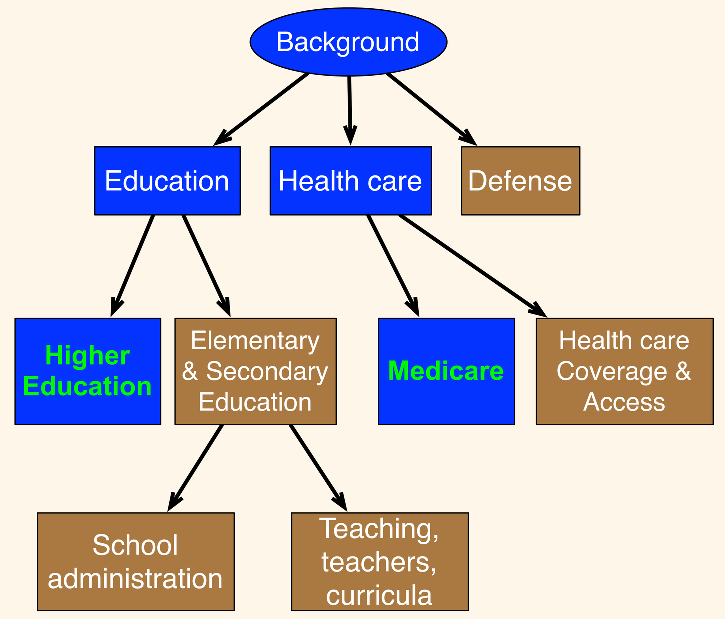
Generating documents

Given a document tagged with labels Higher Education and Medicare, define two label sets:

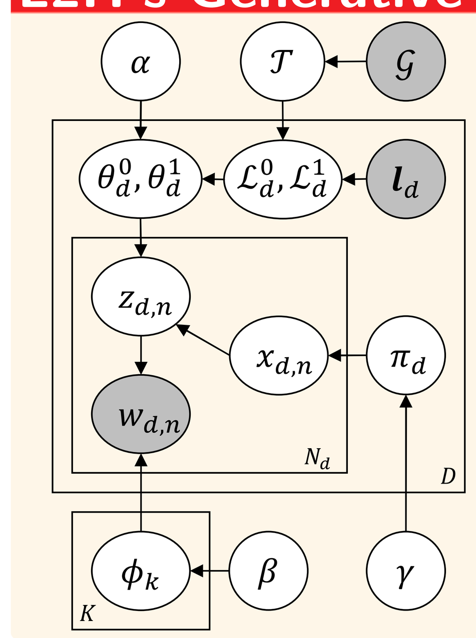
- Candidate set
- Complementary set

For each token

- Choose a label set using a binary switching variable
- Draw a node from the chosen set
- Draw a word type from the node's topic



L2H's Generative Process



1. Create label graph $\mathcal{G}$ and draw a spanning tree $\mathcal{T}$ from $\mathcal{G}$
2. For each node $k \in [1, K]$ in $\mathcal{T}$
   - If k is the root, draw background topic $\phi_k \sim \text{Dir}(\beta u)$
   - Otherwise, draw topic $\phi_k \sim \text{Dir}(\beta\phi_{\sigma(k)})$ where $\sigma(k)$ is node k's parent.
3. For each document $d \in [1, D]$ having labels l_d, define $\mathcal{L}_d^0$ and $\mathcal{L}_d^1$ using $\mathcal{T}$ and l_d
   - Draw $\theta_d^0 \sim \text{Dir}(\mathcal{L}_d^0 \times \alpha)$ and $\theta_d^1 \sim \text{Dir}(\mathcal{L}_d^1 \times \alpha)$
   - Draw a stochastic switching variable $\pi_d \sim \text{Beta}(\gamma_0, \gamma_1)$
   - For each token $n \in [1, N_d]$
     - Draw set indicator $x_{d,n} \sim \text{Bern}(\pi_d)$
     - Draw topic indicator $z_{d,n} \sim \text{Mult}(\theta_d^{x_{d,n}})$
     - Draw word $w_{d,n} \sim \text{Mult}(\phi_{z_{d,n}})$

Posterior Inference: MCMC

Initialization: the hierarchy is initialized by the *maximum spanning tree* on $\mathcal{G}$ (Chu-Liu/Edmonds' algorithm)

Gibbs sampling:

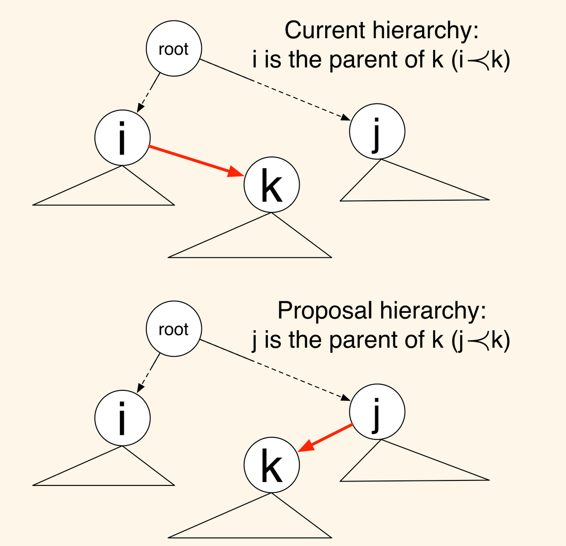
1. Sampling node assignment for each token: $p(x_{d,n} = i, z_{d,n} = k \,|\, x^{-d,n}, z^{-d,n}, \phi, \mathcal{L}_d^i) \propto$

$$\underbrace{\frac{C_{d,i}^{-d,n} + \gamma_i}{C_{d,\cdot}^{-d,n} + \gamma_0 + \gamma_1}}_{p(\text{label set})} \times \underbrace{\frac{N_{d,k}^{-d,n} + \alpha}{C_{d,i}^{-d,n} + \alpha|\mathcal{L}_d^i|}}_{p(\text{node}\,|\,\text{set})} \times \underbrace{\phi_{k,w_{d,n}}}_{p(\text{word}\,|\,\text{node})}, \text{ where } \begin{cases} C_{d,i}, & \text{no. tokens in } d \text{ assigned to label set } i\\ N_{d,k}, & \text{no. toekns in } d \text{ assigned to node } k\end{cases}$$

2. Sampling topic ϕ at each node: two passes over the hierarchy
   - Bottom-up smoothing: estimate the counts propagated from children nodes $\tilde{m}_k$ using the *maximal path assumption*
   - Top-down sampling: sample $\phi_k \sim \text{Dir}(m_k + \tilde{m}_k + \beta\phi_{\sigma(k)})$ using the node's actual counts m_k, propagated counts from its children $\tilde{m}_k$ and its parent's topic $\phi_{\sigma(k)}$

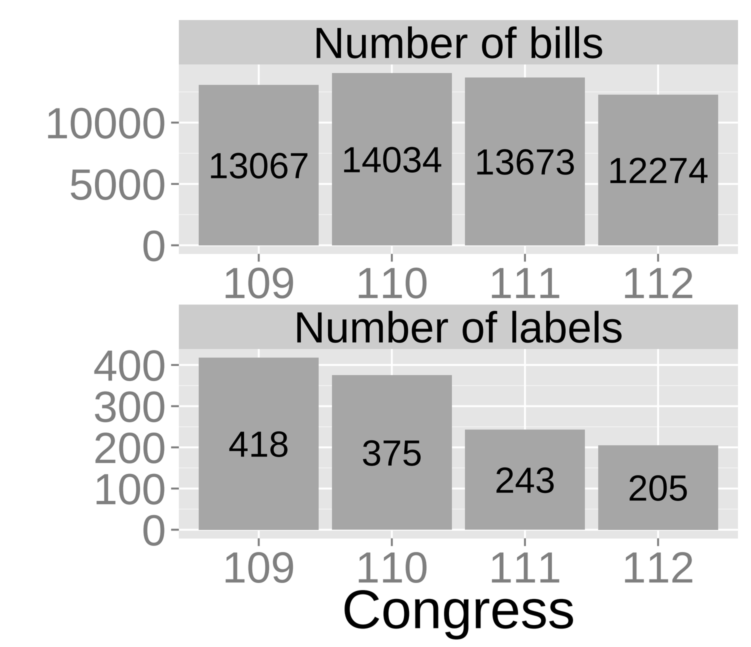
3. Updating tree structure: propose a new parent node for each node, reject if it creates cycle, otherwise accept with Metropolis-Hastings probability $\min\left(1, \frac{Q(i\prec k)}{Q(j\prec k)}\frac{P(j\prec k)}{P(i\prec k)}\right)$.
   The proposal probability is proportional to the edge weight $\frac{Q(i\prec k)}{Q(j\prec k)} = \frac{t_{i,k}}{t_{j,k}}$ and

$$\frac{P(j\prec k)}{P(i\prec k)} = \frac{t_{j,k}}{t_{i,k}} \underbrace{\prod_{d\in\mathcal{D}_{\triangle_k}} \frac{p(z_d\,|\,j\prec k)\,p(x_d\,|\,j\prec k)\,p(w_d\,|\,j\prec k)}{p(z_d\,|\,i\prec k)\,p(x_d\,|\,i\prec k)\,p(w_d\,|\,i\prec k)}}_{\text{for documents having tokens assigned to any node in subtree } \triangle_k \text{ rooted at } k} \prod_{l=1}^{K} \frac{p(\phi_l\,|\,j\prec k)}{p(\phi_l\,|\,i\prec k)}$$



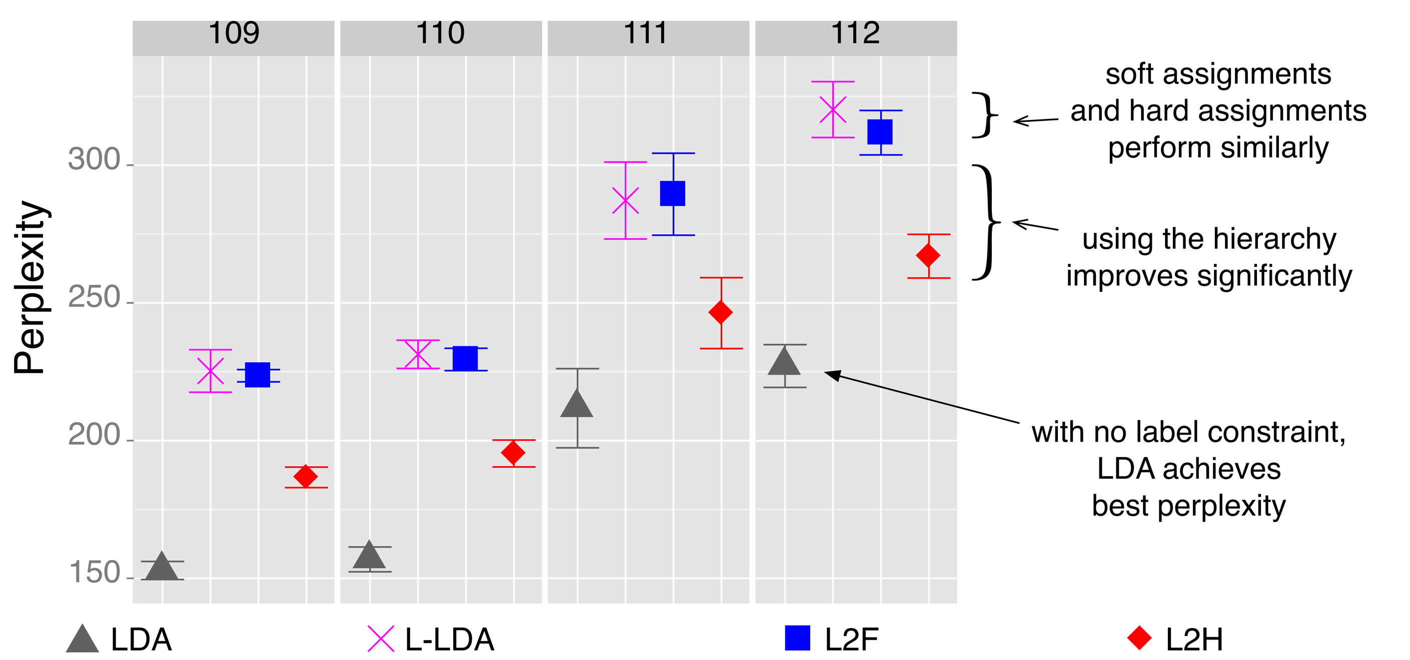
Data

- **Documents**: Congressional bill text of four Congresses (109^{th}–112^{th}) from GovTrack.
- **Labels**: Each bill is labeled with multiple issues by the Congressional Research Service.



Document Modeling

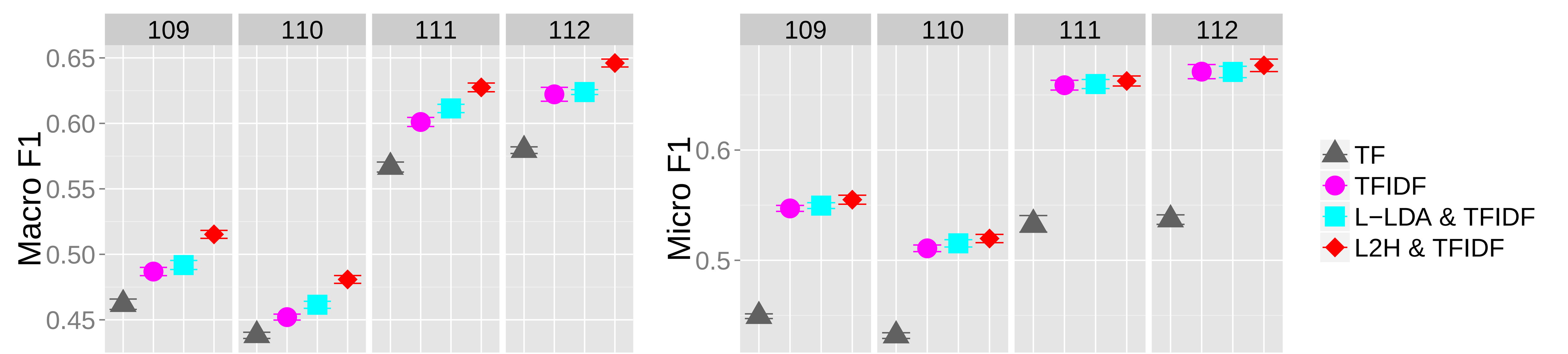
- **Task**: predicting words in held-out documents
- **Evaluation**: perplexity (lower is better)



LDA	Labeled-LDA	Label-to-Flat hierarchy	Label-to-Hierarchy
Unsupervised topic model, no label constraint	(Ramage et al '09) one topic per label, flat structure, *hard assignment* (i.e., document can only be generated from its fixed set of topics)	simplified version of L2H with a fixed flat hierarchy; allows *soft assignments* in contrast to Labeled-LDA	Our proposed model

Multi-label Prediction

The hierarchy improves the performance on multi-label classification.



- **Task**: predicting a set of labels for each test document
- **Evaluation**: Macro F1 and Micro F1
- **Setup**: using M3L–an efficient max-margin multi-label classifier (Hariharan et al. 2012) to study the effectiveness of different sets of features.

Features:

- TF: uses term frequency
- TF-IDF: use TF-IDF instead of raw frequency
- L-LDA&TF-IDF: combines Labeled LDA features with TF-IDF
- L2H&TF-IDF: combines L2H features with TF-IDF

vietan@cs.umd.edu, Jordan.Boyd.Graber@colorado.edu, resnik@umd.edu, jonchang@fb.com