Large-Scale Hierarchical Topic Models

Jay Pujara
Department of Computer Science
University of Maryland
College Park, MD 20742
jay@cs.umd.edu

Peter Skomoroch
LinkedIn Corporation
2029 Stierlin Ct.
Mountain View, CA 94043
pskomoroch@linkedin.com

Abstract

In the past decade, a number of advances in topic modeling have produced sophisticated models that are capable of generating hierarchies of topics. One challenge for these models is scalability: they are incapable of working at the massive scale of millions of documents and hundreds of thousands of terms. We address this challenge with a technique that learns a hierarchy of topics by iteratively applying topic models and processing subtrees of the hierarchy in parallel. This approach has a number of scalability advantages compared to existing techniques, and shows promising results in experiments assessing runtime and human evaluations of quality. We detail extensions to this approach that may further improve hierarchical topic modeling for large-scale applications.

1 Motivation

With massive datasets and corresponding computational resources readily available, the Big Data movement aims to provide deep insights into real-world data. Realizing this goal can require new approaches to well-studied problems. Complex models that, for example, incorporate many dependencies between parameters have alluring results for small datasets and single machines but are difficult to adapt to the Big Data paradigm.

Topic models are an interesting example of this phenomenon. In the last decade, a number of sophisticated techniques have been developed to model collections of text, from Latent Dirichlet Allocation (LDA)[1] through extensions using statistical machinery such as the nested Chinese Restaurant Process [2][3] and Pachinko Allocation[4]. One strength of such approaches is the ability to model topics in a hierarchical fashion. Coarse and fine topics are learned jointly, creating a synergy at multiple levels from shared parameters. However, past experiments have focused on small corpora, such as sampled abstracts from journals with only thousands of documents and terms. Operating on the scale of text collections the size of Wikipedia - millions of documents with millions of terms - is beyond such models using current inference techniques.

While complex models have shown strong analytical results, simpler models that use fewer parameters and produce less structured output are being used at scale. In recent years, many parallel versions of LDA (eg. [5][6][7]) have been developed, capable of summarizing web-scale collections. Although these tools are a boon for analyzing massive amounts of data, they lack the ability to learn a hierarchical representation of topics. This situation highlights a common dilemma in Big Data: whether we can have our cake (models with rich output) and eat it too (operate on massive datasets).

We propose a solution that addresses this dilemma, reusing infrastructure from an existing parallel implementation of LDA in a novel way. Our method provides a scalable mechanism for learning hierarchies from large text collections. We learn top-down hierarchies, first learning topic models at a coarse level, then splitting the corpus into these learned topics and iteratively learning subtopics in parallel. In experiments on large datasets from Wikipedia and TREC, we show fast training times and favorable human interpretability results, supporting exploratory data analysis at web-scale.

2 Background

Latent Dirichlet Allocation (LDA)[1] is a sophisticated topic model that has served as the foundation for much recent work in topic modeling. The method, defined for k topics and d documents, models topics as distributions over words (β_k), and documents as mixtures of topics (θ_d). For each word in a document d, $w_{d,n}$, a topic $z_{d,n}$ is chosen from the document's topic distribution (θ_d), and a word is chosen from the topic's distribution over words (β_k). The Dirichlet distribution of θ_d and β_k are parameterized by terms α and λ respectively. This generative story is reflected in the joint distribution shown in Equation 1.

$$p(w, z, \theta, \beta|\alpha, \lambda) = \prod_k p(\beta_k|\lambda) \prod_d p(\theta_d|\alpha) \prod_n p(z_{d,n})p(w_{d,n}|\beta_{z_{d,n}}) \tag{1}$$

Topic models are learned through inference on the generative model described. This inference problem is intractable at scale, and approximate techniques such as Gibbs Sampling or using variational methods are used during modeling. Zhai et al. [7] have argued that variational inference is best suited to learning topic models in large-scale settings, as intra-document dependencies can be minimized through the choice of a variational distribution. Supporting this claim, they've provided an open-source, Apache-licensed implementation for topic modeling called Mr.LDA for Hadoop.

Hierarchical topic models [2] are often in the form of a DAG where nodes correspond to topics at varying levels of granularity. A number of models have been proposed for learning hierarchical topic models, but a common strategy is to choose a *set of topics* for each word in the document, corresponding to a path in the hierarchy. Using such models allows us to understand the relationship between topics at differing levels of the hierarchy, but at a cost. Instead of storing parameters for each term for each of k topics, we must now consider parameters for each *path* in the hierarchy, or k^l parameters for an l-level hierarchy with constant branching factor (and potentially more in techniques such as PAMs). Each of these parameters must maintain values for each term in our vocabulary V. The domains where hierarchies might prove most useful - massive, diverse collections using natural language - are likely to have very large vocabularies.

3 Method and Discussion

Algorithm 1 ITERATIVE-TOPIC-HIERARCHY: iteratively learn a topic hierachy

Require: Dataset D
Require: Parameters k, LEVELS
Require: LEARN-TOPICS, learn topics from data, as in Mr.LDA : produces $\theta_{1:D}$ and $\beta_{1:k}$
Require: SPLIT-CORPUS, distribute dataset into k parts using topic-model M

```
LEARN-NODE(k, D, 0, LEVELS)

function LEARN-NODE(k, D, l, LEVELS)
    if l < LEVELS then
        ⟨θ_{1:D}, β_{1:k}⟩ = LEARN-TOPICS(k, D)
        D_{1:k} = SPLIT-CORPUS(θ_{1:D}, D)
        for i = 1 → k do
            LEARN-NODE(k, D_i, l + 1, LEVELS)
        end for
    end if
end function
```

Our approach learns a top-down hierarchy iteratively and is summarized in Algorithm 1. First, we learn a topic model of k topics using the entire corpus. Next, we create k new datasets and allocate each document in the corpus to zero, one, or more of the k datasets using method `SPLIT-CORPUS` and the θ values learned by the topic model. Finally, we learn k new topic models using the k

datasets constructed in the previous step. Each of these topic models corresponds to subtopics of one of the original k topics. This procedure can be applied iteratively, generating further levels of the hierarchy. Since our method doesn't add dependencies from subtopics to their parents (or siblings), the process is easily parallelizable by launching each learning task independently.

The consequence of these decisions allows us to leverage massive datasets while escaping some of the limitations of current hierarchical topic modeling. Existing models attempt to learn parameters *jointly* across levels of the hierarchy, while our approach learns at a single level, implicitly conditioning on ancestors. Instead of learning parameters for each *path* in the hierarchy, we learn parameters for each *node*, removing dependencies between nodes and allowing learning to take place in parallel. Finally, our method operates in a coarse-to-fine fashion, first learning a flat topic model and refining each topic, using information from the coarse model to make decisions about these refinements through its allocation of documents to subtasks.

4 Implementation Details

Our implementation uses the Mr.LDA package to learn the topic model (`LEARN-TOPICS` in Algorithm 1). We make customizations to this package to allow easier parallelization of learning and add support three models providing the functionality of `SPLIT-CORPUS`:

- `SPLIT-SINGLE` chooses the highest probability topic for each document d using θ_d and allocates the document to the corresponding topic: $[D_k = \{d : \text{mode}(\theta_d) = k\}]$
- `SPLIT-MULTIPLE` creates c copies of the document, and proportionally allocates those documents according to the probabilities in θ_d: $[D_k = \{d \times \text{cnt} : \text{cnt} = \text{round}(c * \theta_d[k])\}]$
- `SPLIT-SELECT` applies a threshold, t, on the entropy (H) of θ_d and then allocates documents below the threshold to the highest probability topic (as in `SPLIT-SINGLE`): $[D_k = \{d : \text{mode}(\theta_d) = k \wedge H(\theta_d) < t\}]$

`SPLIT-SINGLE` provides a very simple method to split the corpus, and has the added advantage that each document is used exactly once in each level of the hierarchy. `SPLIT-MULTIPLE` embodies the tenet of LDA topic modeling that each document is a mixture of topics by creating multiple copies of each document. A drawback of this approach is that the number of documents grows with each level, c^l, although this can be addressed with horizontal scaling. Another drawback is that, due to the allocation of less relevant documents to a topic, incoherent subtopics may be learned. Finally, `SPLIT-SELECT` chooses only documents that strongly map to a particular topic by using a threshold based on the entropy of the θ_d distribution. This allows us to cull irrelevant documents, but possibly at the cost of losing specialized subtopics that occur rarely in the dataset.

5 Datasets and Results

Our method was applied to two large-scale document collections - TREC and Wikipedia. The TREC collection consists of 473K documents from the Financial Times and LA Times[8]. Tokens were processed by stemming and removing terms that occur less than 20 times, yielding a vocabulary of 60K. The Wikipedia dataset consists of 3M documents with a vocabulary of 1.8M after removing tokens that either occurred in more than 90% of documents or less than 20 times. Illustrative results are also presented for LinkedIn member profiles in the Appendix.

For both of these collections, we learned a 2-level topic hierarchy on a modest Hadoop cluster. For the TREC corpus, we used a branching factor of 5 (5 topics at the root, 5 subtopics for each topic). For each learning phase, we used 40 mappers and 20 reducers. The root topic model ran for 22 iterations of variational inference with an average iteration time of 360s, after which the model converged (log likelihood changed by less than a factor of 10^{-6}). Subtopics from the root were run for 20 iterations, with an average iteration time of 230s using `SPLIT-SINGLE` corpus allocation. The total time for learning the full hierarchy was approximately 3.5 hours. For the larger Wikipedia corpus, we used a branching factor of 10 (10 root topics with 10 subtopics each). The root topic model converged after 41 iterations, with an average iteration time of 2300s. Subtopics from the root were run for 20 iterations, with an average iteration time of 700s using `SPLIT-SINGLE`

SINGLE	japan korea moscow nuclear russia	los metro angel orang san	bosnia serb london attack kill	america panama colombia prevent latin	hong kong li provinci comrad
MULTIPLE	report 94 forc govern militari	mr govern time industri financi	column counti citi who part	94 report 1994 english peopl	parti govern 94 report state
SELECT	industri london moslem ft product	traffick santa mexico park beach	tass africa itar sov herzegovina	list noriega won seat known	evolutionari comrad idea chuch procuratori

Table 1: Three sets of subtopics of topic 4 from the TREC corpus (brief/investig/korea/beij) generated by different document-allocation techniques. (Full TREC hierarchy in Appendix)

Measure	SINGLE	MULTIPLE	SELECT
Total Documents	472524	2362303	333693
Iteration time (s)	230	405	77
Intruders Found (%)	27.4	13.0	25.5

Table 2: Evaluation of scalability and quality of document-allocation techniques

corpus allocation. The total time for learning the full hierarchy was approximately 30 hours. The full hierarchy, with the top 5 IDF-normalized terms for each topic, is shown in the Appendix.

We proposed three methods for splitting the dataset when allocating documents to subtrees, and sample output for each is shown in Table 1. SPLIT-MULTIPLE was run with $c = 5$, so that 5 copies of each document were generated and allocated among the 5 possible subtopics. SPLIT-SELECT was run with $t = .72$, where documents where the entropy of θ_d was less than .72 were assigned to the most-probable topic in the multinomial and all others were ignored.

Speed and quality for these methods are assessed in Table 2. Adding multiple copies of documents increased running time by 75%, while only choosing low-entropy documents decreased running time by 66%. A growing trend in topic models is to focus on interpretablity rather than model log likelihood [9], using *word intrusion* as a metric. Humans are presented with the top terms for a topic as well as an intruder - a term that has low probability for the current topic but high probability for some other topic. We conducted such an evaluation with six participants, using the 10 top terms in each topic as well as an intruder term, so that randomly choosing a term would have a 9.1% success rate. The intruder term was among the top 100 terms for another topic in the same subtree, but ranked between 250-500 for the chosen topic. Terms in each list were presented in random order, and the term lists were randomized across methods. The results suggest that the MULTIPLE (15.9%) strategy lags behind both SINGLE (27.4%) and SELECT (25.5%) which have similar results.

6 Conclusion and Future Work

The hierarchical topic modeling approach we've presented fulfills two goals outlined - rich output and scalability. The hierarchies learned are interpretable, both qualitatively and based on a word-intrusion evaluation. The method is scalable, operating on millions of documents and terms and producing results on the scale of hours and days. One aspect of our approach is allocating documents to subtasks learning different branches of the hierarchy. Our evaluation suggests that single-allocation or a selective-allocation both perform well, with a small quality-scalability tradeoff. In future work, we hope to perform more extensive evaluation, including a comparison of these techniques to prior models trained on a smaller sample of data. We are also interested in implementing a parallel variational inference algorithm for hierarchical LDA that uses sketches or hashing to reduce the number of parameters.

References

[1] David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[2] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, 2003.

[3] Chong Wang and David M. Blei. Variational inference for the nested chinese restaurant process. In *NIPS*, 2009.

[4] Wei Li and Andrew Mccallum. Pachinko Allocation: Scalable Mixture Models of Topic Correlations. *Journal of Machine Learning Research*, 2008.

[5] Ramesh Nallapati, William W. Cohen, and John D. Lafferty. Parallelized variational em for latent dirichlet allocation: An experimental evaluation of speed and scalability. In *ICDM Workshops*, 2007.

[6] Feng Yan, Ningyi Xu, and Yuan Qi. Parallel inference for latent dirichlet allocation on graphics processing units. In *NIPS*, 2009.

[7] Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad Alkhouja. Mr. LDA: A flexible large scale topic modeling package using variational inference in mapreduce. In *WWW*, 2012.

[8] National Institute of Standards and Technology. Special db 22 and nist trec document database. `http://www.nist.gov/srd/text.cfm`.

[9] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

7 Appendix

The customizations to Mr.LDA we detail are freely available at `https://github.com/puuj/Mr.LDA`

Terms shown in the tables below are normalized using a formula similar to TF-IDF. The probability of the term is used in lieu of term frequency, and inverse *topic* frequency is used to decrease the weight of terms that appear in multiple topics at the same level of the hierarchy. In the output shown, the top 500 terms from each topic were used for this normalization.

To save space, hierarchies are shown as tables in the Appendix. The first row of each table corresponds to a topic at the root level. Subsequent entries in each column are subtopics: children of the topic seen in the first row.

topic 0	**topic 1**	**topic 2**	**topic 3**	**topic 4**
design	dollar	game	al	brief
materi	share	team	arab	investig
use	uk	sport	palestinian	korea
qualiti	profit	season	israel	beij
space	stock	photo	isra	letter
subtopic 0_0	**subtopic** 1_0	**subtopic** 2_0	**subtopic** 3_0	**subtopic** 4_0
comput	appoint	shop	resourc	japan
softwar	airlin	museum	task	korea
electron	sir	restaur	materi	moscow
machin	labour	buy	labor	nuclear
user	court	fashion	properti	russia
subtopic 0_1	**subtopic** 1_1	**subtopic** 2_1	**subtopic** 3_1	**subtopic** 4_1
music	pension	inning	uk	los
artist	risk	pitch	pound	metro
news	life	goal	93	angel
garden	save	touchdown	gatt	orang
british	mortgag	led	round	san
subtopic 0_2	**subtopic** 1_2	**subtopic** 2_2	**subtopic** 3_2	**subtopic** 4_2
beij	index	russian	battalion	bosnia
farmer	00	concert	artilleri	serb
daili	se	moscow	ukrainian	london
fbis	qtr	type	enemi	attack
provinc	90	radio	zhirinovski	kill
subtopic 0_3	**subtopic** 1_3	**subtopic** 2_3	**subtopic** 3_3	**subtopic** 4_3
ecolog	pre	driver	arafat	america
command	interim	feet	plo	panama
weapon	fin	accid	jordan	colombia
coordin	5m	grade	jan	prevent
combat	3m	ride	cairo	latin
subtopic 0_4	**subtopic** 1_4	**subtopic** 2_4	**subtopic** 3_4	**subtopic** 4_4
communiti	percent	attorney	bosnia	hong
district	china	law	muslim	kong
resid	nuclear	council	polic	li
board	korea	murder	human	provinci
metro	english	judg	kill	comrad

Table 3: Topic Hierarchy with branching factor 5 for TREC-60K corpus using SPLIT-SINGLE

topic 0	**topic 1**	**topic 2**	**topic 3**	**topic 4**	**topic 5**	**topic 6**	**topic 7**	**topic 8**	**topic 9**
words	records	species	officers	plot	route	64	catholic	league	election
physical	recorded	genus	enemy	episodes	railway	cdp	bishop	football	committee
cannot	festival	food	combat	friend	lake	44	lord	cup	institute
might	theatre	plants	ordered	told	municipality	bureau	sir	championship	student
individua	musical	fish	armed	woman	km	poverty	charles	teams	professor
subtopic 0_0	**subtopic** 1_0	**subtopic** 2_0	**subtopic** 3_0	**subtopic** 4_0	**subtopic** 5_0	**subtopic** 6_0	**subtopic** 7_0	**subtopic** 8_0	**subtopic** 9_0
lord	scenes	birds	gas	wwe	la	eighteens	sweden	pitcher	means
wife	takes	males	cases	wwf	del	eighteen	anne	era	treatment
movie	uses	female	scholars	tag	spanish	sixty	swedish	asteroid	cannot
asks	tells	tail	argued	raw	brazil	4	husband	sox	drug
scene	idea	male	worlds	triple	el	6	denmark	pitched	therefore
subtopic 0_1	**subtopic** 1_1	**subtopic** 2_1	**subtopic** 3_1	**subtopic** 4_1	**subtopic** 5_1	**subtopic** 6_1	**subtopic** 7_1	**subtopic** 8_1	**subtopic** 9_1
search	dancers	served	film	hollywood	storm	elementary	theatre	punt	film
database	dancing	tea	singh	roles	tropical	valley	opera	kick	television
email	bollywood	chinese	music	starred	hurricane	republican	plays	song	polish
page	repertoire	style	love	festival	damage	post	musical	kicker	writer
client	dancer	dish	art	filming	structure	nearby	song	qb	newspapers
subtopic 0_2	**subtopic** 1_2	**subtopic** 2_2	**subtopic** 3_2	**subtopic** 4_2	**subtopic** 5_2	**subtopic** 6_2	**subtopic** 7_2	**subtopic** 8_2	**subtopic** 9_2
gun	game	waste	li	band	peak	railway	gardens	renamed	bar
planets	electronic	consumption	formula	album	rock	wind	rooms	seats	vice
spacecraft	indie	concrete	wang	guitar	lakes	coal	roof	construction	illinois
hebrew	fulllength	emissions	liu	gameplay	columbia	railways	features	seating	democrat
launch	labels	farm	han	playstation	summit	plants	top	empire	jersey
subtopic 0_3	**subtopic** 1_3	**subtopic** 2_3	**subtopic** 3_3	**subtopic** 4_3	**subtopic** 5_3	**subtopic** 6_3	**subtopic** 7_3	**subtopic** 8_3	**subtopic** 9_3
disorder	ragam	subtropical	irish	windows	india	fuselage	ship	wickets	scientific
actions	scale	flowers	uniform	users	germany	mk	ships	firstclass	phd
cognitive	iranian	moist	singapore	web	poland	altitude	fleet	manchester	economics
self	revolution	moth	ireland	data	literacy	ft	hms	cricketer	physics
consciousness	fwv	flowering	ira	user	prefecture	kg	admiral	batsman	environmental
subtopic 0_4	**subtopic** 1_4	**subtopic** 2_4	**subtopic** 3_4	**subtopic** 4_4	**subtopic** 5_4	**subtopic** 6_4	**subtopic** 7_4	**subtopic** 8_4	**subtopic** 9_4
guitar	interview	protein	economy	emperor	ride	internet	regiment	tag	labour
tv	wanted	cell	asia	prince	brand	wireless	brigade	poker	seats
advertising	really	gene	african	li	companys	communications	battalion	riders	parliamentary
announced	asked	proteins	taiwan	liu	went	video	awarded	wrestler	cabinet
stations	saying	dna	armenian	chinese	livery	microsoft	medal	nwa	constituency
subtopic 0_5	**subtopic** 1_5	**subtopic** 2_5	**subtopic** 3_5	**subtopic** 4_5	**subtopic** 5_5	**subtopic** 6_5	**subtopic** 7_5	**subtopic** 8_5	**subtopic** 9_5
angle	signal	tambon	divisions	cars	china	taxes	refer	olympics	communities
connected	frequency	amphoe	battalions	contestants	chinese	revenue	surname	olympic	hong
curve	transmitter	villages	offensive	round	hong	rates	arabic	silver	kong
index	operated	town	positions	contestant	kong	health	manuscript	bronze	providing
let	site	eruption	armoured	aircraft	oil	pay	codex	metres	needs
subtopic 0_6	**subtopic** 1_6	**subtopic** 2_6	**subtopic** 3_6	**subtopic** 4_6	**subtopic** 5_6	**subtopic** 6_6	**subtopic** 7_6	**subtopic** 8_6	**subtopic** 9_6
particle	starred	railway	greece	gets	parkway	heat	cathedral	ferrari	billion
cancer	season	services	berlin	tries	turnpike	requirements	papal	schumacher	americans
electron	cast	business	occupation	asks	toll	greater	santa	mclaren	budget
acid	productions	founded	serbia	decides	corridor	battery	portugal	f1	samesex
electrons	broadway	launched	hungary	leaves	bypass	speeds	venice	hamilton	increased
subtopic 0_7	**subtopic** 1_7	**subtopic** 2_7	**subtopic** 3_7	**subtopic** 4_7	**subtopic** 5_7	**subtopic** 6_7	**subtopic** 7_7	**subtopic** 8_7	**subtopic** 9_7
chinese	poland	star	la	trial	terminal	airfield	christians	fifa	bishop
verb	gmina	symptoms	promoted	ride	platform	assigned	prayer	uefa	command
verbs	district	syndrome	charles	arrested	airlines	squadrons	gods	midfielder	commander
nouns	voivodeship	bone	sir	criminal	navy	corps	divine	serie	seminary
gender	administrative	muscle	puerto	train	airline	missions	gospel	defender	navy
subtopic 0_8	**subtopic** 1_8	**subtopic** 2_8	**subtopic** 3_8	**subtopic** 4_8	**subtopic** 5_8	**subtopic** 6_8	**subtopic** 7_8	**subtopic** 8_8	**subtopic** 9_8
season	peaked	x	wing	cable	team	racing	boys	nba	football
super	listing	f	flying	channels	league	v8	tibetan	ncaa	teams
xmen	rb	points	fighter	affiliate	stadium	chassis	guru	assists	basketball
nintendo	contest	theorem	engine	satellite	teams	cc	pupils	tackles	athletic
ball	grammy	equation	submarine	stores	play	sports	girls	coached	clubs
subtopic 0_9	**subtopic** 1_9	**subtopic** 2_9	**subtopic** 3_9	**subtopic** 4_9	**subtopic** 5_9	**subtopic** 6_9	**subtopic** 7_9	**subtopic** 8_9	**subtopic** 9_9
university	museum	spoken	violence	homer	income	equation	ministry	fight	arrested
students	fiction	dialects	trial	author	median	p	meeting	boxing	told
education	gallery	speakers	al	simpsons	households	theorem	seminary	stakes	alleged
professor	science	geological	lebanon	century	65	values	episcopal	decision	killed
critical	editor	earths	syria	bart	size	vector	theological	64	refused

Table 4: Topic Hierarchy with branching factor 10 for Wikipedia corpus using SPLIT-SINGLE

topic 0	**topic 1**	**topic 2**
C++	Microsoft Word	Supply Chain
Java	Teachers	Supply Chain Management
PHP	Teaching	Business Strategy
CSS	Students	Purchasing
C#	Microsoft Excel	Warehousing
C	English	Contract Negotiation
MySQL	Curriculum Design	Inventory Management
.NET	Courses	Procurement
HTML	Editing	Negotiation
XML	Academia	New Business Development
subtopic 0_0	**subtopic** 1_0	**subtopic** 2_0
Research	English Composition	Lean Manufacturing
Algorithms	British Literature	Six Sigma
Matlab	Discourse	Continuous Improvement
Analysis	World Literature	Manufacturing
Data	Graduate Record Examinations	Process Improvement
subtopic 0_1	**subtopic** 1_1	**subtopic** 2_1
Java EE	Community Colleges	New Business Development
Hibernate	Greek Life	Marketing Strategy
Spring	Counselor Education	Sales Management
Tomcat	First Year Experience	Business Strategy
Java	Personality Development	Strategic Planning
subtopic 0_2	**subtopic** 1_2	**subtopic** 2_2
Microsoft SQL Server	Science Communication	Logistics
C#	Human Physiology	Transportation
Data	Laboratory Skills	Warehousing
Applications	Dissection	Shipping
Client	Invertebrates	Freight
subtopic 0_3	**subtopic** 1_3	**subtopic** 2_3
Electrical Engineering	Electronic Resources	Retail
Embedded Systems	Special Collections	Sales
VHDL	Academic Libraries	Brand
Electronics	Virtual Reference	Marketing
FPGA	Library Research	Merchandising
subtopic 0_4	**subtopic** 1_4	**subtopic** 2_4
PHP	School Psychology	Procurement
CSS	Abnormal Psychology	Purchasing
MySQL	Cognitive Neuroscience	Logistics
Javascript	Communication Disorders	Supply Chain Management
HTML	Health Psychology	Supply Chain

Table 5: Subset of hierarchy on LinkedIn member profiles. A 2-level hierarchy with branching factor 15 was learned using SPLIT-SINGLE. Each "document" was a member profile, and the terms in each document were explicitly or implicitly specified skills identified by LinkedIn (see `http://www.linkedin.com/skills/`). There were tens of millions of documents and tens of thousands of terms, with a training time of approximately one day using hundreds of mappers and reducers.