ToPMine: Phrase Mining without Training Data

Strategy 3: First Phrase Mining then Topic Modeling

- Why first Phrase Mining then Topic Modeling?
  - With Strategy 2, tokens in the same phrase may be assigned to different topics
    - Ex. knowledge discovery using least squares support vector machine classifiers…
      - *Knowledge discovery* and *support vector machine* should have coherent topic labels
- Solution: switch the order of phrase mining and topic model inference

[knowledge discovery] using [least squares] [support vector machine] [classifiers] …	→	[knowledge discovery] using [least squares] [support vector machine] [classifiers] …

- Techniques for this strategy
  - Phrase mining, document segmentation, and phrase ranking
  - Topic model inference with phrase constraint

ToPMine: Phrase Mining before Topic Modeling

- **ToPMine** [El-Kishky et al. VLDB'15]: Phrase mining, then phrase-based topic modeling
- Phrase mining
  - Frequent ***contiguous pattern*** mining: Extract candidate phrases and their counts
  - Agglomerative merging of adjacent unigrams as guided by a **significance score**
  - Document segmentation to count phrase occurrence
    - Calculate rectified (i.e., true) phrase frequency
  - Phrase ranking (using the criteria proposed in KERT)
    - Popularity, concordance, informativeness, completeness
- Phrase-based topic modeling
  - The mined bag-of-phrases are passed as input to PhraseLDA, an extension of LDA, that constrains all words in a phrase to each sharing the same latent topic

Phrase	Raw frequency	Rectified frequency
[support vector machine]	90	80
[vector machine]	95	0
[support vector]	100	20

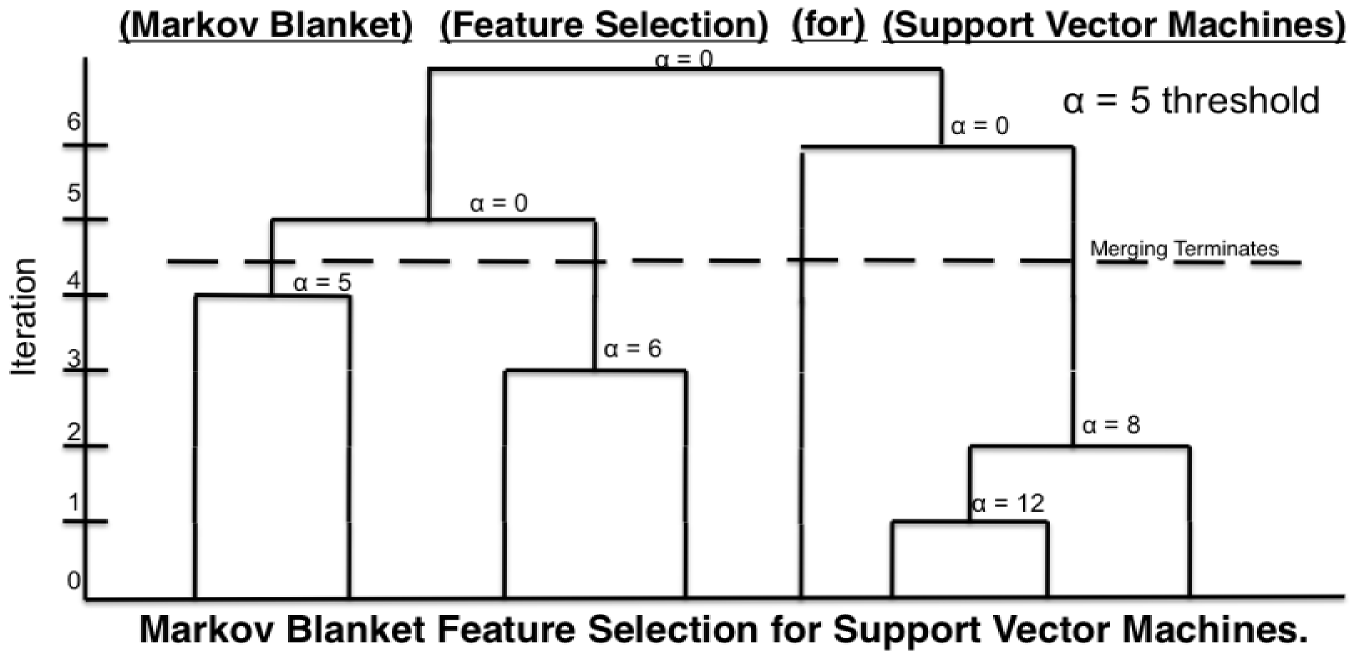
Collocation Mining

- Collocation: A sequence of words that occur more frequently than expected
  - Often “interesting”, relay information not portrayed by their constituent terms
    - Ex. “made an exception”, “strong tea”
- Many different measures used to extract collocations from a corpus [Dunning 93, Pederson 96]
  - E.g., mutual information, t-test, z-test, chi-squared test, likelihood ratio

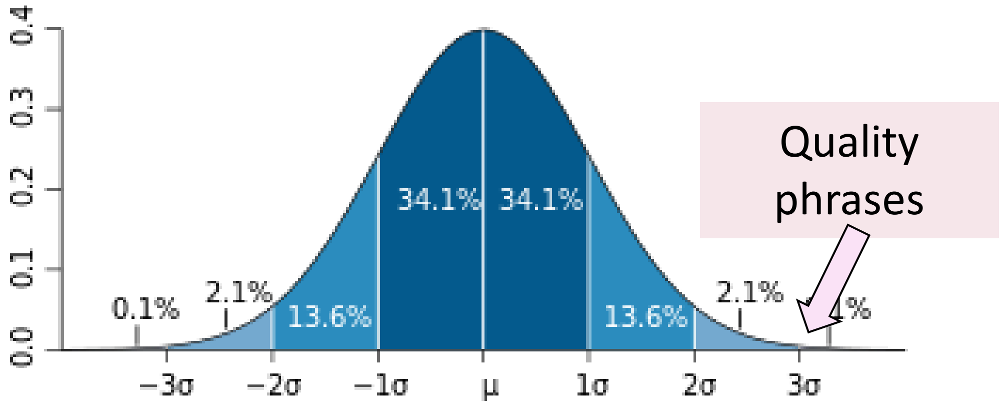
$$\mathrm{PMI}(x,y) = \log \frac{p(x,y)}{p(x)p(y)} \qquad sig = \frac{count(phr_{x+y}) - E[count(phr_{x+y})]}{\sqrt{count(phr_{x+y})}} \qquad \chi^2 = \sum \frac{(O-E)^2}{E}$$

- Many of these measures can be used to guide the agglomerative **phrase-segmentation** algorithm

Phrase Candidate Generation: Frequent Pattern Mining + Statistical Analysis



[Markov blanket] [feature selection] for [support vector machines]
[knowledge discovery] using [least squares] [support vector machine] [classifiers]
…[support vector] for [machine learning]…



Based on significance score [Church et al.'91]:

$$\alpha(P_1, P_2) \approx (f(P_1 \bullet P_2) - \mu_0(P_1,P_2))/\surd\, f(P_1 \bullet P_2)$$

Note for the first title:

- [feature selection] forms phrase but not [selection for] based on the significant scores computed
- [support vector machine] does not contribute to the counts of [support], [vector], [support vector], [vector machine]

ToPMine: Experiments on DBLP Abstracts

	Topic 1	*Topic 2*	*Topic 3*	*Topic 4*	*Topic 5*
unigrams	problem	word	data	programming	data
	algorithm	language	method	language	patterns
	optimal	text	algorithm	code	mining
	solution	speech	learning	type	rules
	search	system	clustering	object	set
	solve	recognition	classification	implementation	event
	constraints	character	based	system	time
	programming	translation	features	compiler	association
	heuristic	sentences	proposed	java	stream
	genetic	grammar	classifier	data	large
n-grams	genetic algorithm	natural language	data sets	programming language	data mining
	optimization problem	speech recognition	support vector machine	source code	data sets
	solve this problem	language model	learning algorithm	object oriented	data streams
	optimal solution	natural language processing	machine learning	type system	association rules
	evolutionary algorithm	machine translation	feature selection	data structure	data collection
	local search	recognition system	paper we propose	program execution	time series
	search space	context free grammars	clustering algorithm	run time	data analysis
	optimization algorithm	sign language	decision tree	code generation	mining algorithms
	search algorithm	recognition rate	proposed method	object oriented programming	spatio temporal
	objective function	character recognition	training data	java programs	frequent itemsets

ToPMine is efficient and generates high-quality topics and phrases without any training data

ToPMine: Experiments on Yelp Reviews

	Topic 1	*Topic 2*	*Topic 3*	*Topic 4*	*Topic 5*
unigrams	coffee	food	room	store	good
	ice	good	parking	shop	food
	cream	place	hotel	prices	place
	flavor	ordered	stay	find	burger
	egg	chicken	time	place	ordered
	chocolate	roll	nice	buy	fries
	breakfast	sushi	place	selection	chicken
	tea	restaurant	great	items	tacos
	cake	dish	area	love	cheese
	sweet	rice	pool	great	time
n-grams	ice cream	spring rolls	parking lot	grocery store	mexican food
	iced tea	food was good	front desk	great selection	chips and salsa
	french toast	fried rice	spring training	farmer's market	food was good
	hash browns	egg rolls	staying at the hotel	great prices	hot dog
	frozen yogurt	chinese food	dog park	parking lot	rice and beans
	eggs benedict	pad thai	room was clean	wal mart	sweet potato fries
	peanut butter	dim sum	pool area	shopping center	pretty good
	cup of coffee	thai food	great place	great place	carne asada
	iced coffee	pretty good	staff is friendly	prices are reasonable	mac and cheese
	scrambled eggs	lunch specials	free wifi	love this place	fish tacos

ToPMine works well for phrase and topic mining in social media data