Constraint-Based Pattern Mining

Constraint-Based Pattern Mining

- Why Constraint-Based Mining?
- Different Kinds of Constraints: Different Pruning Strategies
- Constrained Mining with Pattern Anti-Monotonicity
- Constrained Mining with Pattern Monotonicity
- Constrained Mining with Data Anti-Monotonicity
- Constrained Mining with Succinct Constraints
- Constrained Mining with Convertible Constraints
- Handling Multiple Constraints
- Constraint-Based Sequential-Pattern Mining

Why Constraint-Based Mining?

Why Constraint-Based Mining?

- Finding all the patterns in a dataset autonomously?—unrealistic!
  - Too many patterns but not necessarily user-interested!
- Pattern mining in practice: Often a user-guided, interactive process
  - User directs what to be mined using a data mining query language (or a graphical user interface), specifying various kinds of constraints
- What is constraint-based mining?
  - Mine together with user-provided constraints
- Why constraint-based mining?
  - User flexibility: User provides constraints on what to be mined
  - Optimization: System explores such constraints for mining efficiency
    - E.g., Push constraints deeply into the mining process

Various Kinds of User-Specified Constraints in Data Mining

- Knowledge type constraint—Specifying what kinds of knowledge to mine
  - Ex.: Classification, association, clustering, outlier finding, …
- Data constraint—using SQL-like queries
  - Ex.: Find products sold together in NY stores this year
- Dimension/level constraint—similar to projection in relational database
  - Ex.: In relevance to region, price, brand, customer category
- Interestingness constraint—various kinds of thresholds
  - Ex.: Strong rules: min_sup $\geq$ 0.02, min_conf $\geq$ 0.6, min_correlation $\geq$ 0.7
- Rule (or pattern) constraint ⇐ The focus of this study
  - Ex.: Small sales (price < \$10) triggers big sales (sum > \$200)

Constrained Mining with Pattern Anti-Monotonicity

Pattern Space Pruning with Pattern Anti-Monotonicity

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

min_sup = 2

Item	Price	Profit
a	100	40
b	40	0
c	150	−20
d	35	−15
e	55	−30
f	45	−10
g	80	20
h	10	5

- A constraint *c* is ***anti-monotone***
  - If an itemset S **violates** constraint *c*, so does any of its superset
  - That is, mining on itemset S can be terminated
- Ex. 1: c_1: $sum(S.price) \leq v$ is anti-monotone
- Ex. 2: c_2: range(S.profit) $\leq$ 15 is anti-monotone
  - Itemset *ab* violates c_2 (range(ab) = 40)
  - So does every superset of *ab*
- Ex. 3. c_3: $sum(S.Price) \geq v$ is not anti-monotone
- Ex. 4. Is c_4: $support(S) \geq \sigma$ anti-monotone?
  - Yes! Apriori pruning is essentially pruning with an anti-monotonic constraint!

Note: item.price > 0
Profit can be negative

Constrained Mining with Pattern Monotonicity

Pattern Monotonicity and Its Roles

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

min_sup = 2

Item	Price	Profit
a	100	40
b	40	0
c	150	−20
d	35	−15
e	55	−30
f	45	−10
g	80	20
h	10	5

- A constraint *c* is *monotone*: If an itemset S **satisfies** the constraint c, so does any of its superset
  - That is, we do not need to check *c* in subsequent mining
- Ex. 1: c_1: $sum(S.Price) \geq v$ is monotone
- Ex. 2: c_2: $min(S.Price) \leq v$ is monotone
- Ex. 3: c_3: range(S.profit) $\geq$ 15 is monotone
  - Itemset *ab* satisfies c_3
  - So does every superset of *ab*

Note: item.price > 0
Profit can be negative

Constrained Mining with Data Anti-Monotonicity

Data Space Pruning with Data Anti-Monotonicity

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

min_sup = 2

Item	Price	Profit
a	100	40
b	40	0
c	150	−20
d	35	−15
e	55	−30
f	45	−10
g	80	20
h	10	5

- A constraint c is *data anti-monotone*: In the mining process, if a data entry *t* cannot satisfy a pattern *p* under *c*, *t* cannot satisfy *p*'s superset either
  - Data space pruning: Data entry *t* can be pruned
- Ex. 1: c_1: $sum(S.Profit) \geq v$ is data anti-monotone
  - Let constraint c_1 be: $sum(S.Profit) \geq 25$
    - T_{30}: {b, c, d, f, g} can be removed since none of their combinations can make an S whose sum of the profit is ≥ 25
- Ex. 2: c_2: $min(S.Price) \leq v$ is data anti-monotone
  - Consider $v = 5$ but every item in a transaction, say T_{50}, has a price higher than 10
- Ex. 3: c_3: $range(S.Profit) > 25$ is data anti-monotone

Note: item.price > 0
Profit can be negative

Data Space Pruning Should Be Explored Recursively

- Example. c_3: *range*(*S.Profit*) > *25*
  - We check b's projected database
    - But item "a" is infrequent (sup = 1)
  - After removing "a (40)" from T_{10}
    - T_{10} cannot satisfy c_3 any more
      - Since "b (0)" and "c (−20), d (−15), f (−10), h (5)"
    - By removing T_{10}, we can also prune "h" in T_{20}

b's-proj. DB

TID	Transaction
10	a, c, d, f, h
20	c, d, f, g, h
30	c, d, f, g

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

Item	Profit
a	40
b	0
c	−20
d	−15
e	−30
f	−10
g	20
h	5

min_sup = 2

price(item) > 0

Constraint: range{S.profit} > 25

b's-proj. DB

TID	Transaction
~~10~~	~~a, c, d, f, h~~
20	c, d, f, g, ~~h~~
30	c, d, f, g

Recursive Data Pruning

b's FP-tree

single branch: cdfg: 2

Only a single branch "cdfg: 2" to be mined in b's projected DB

- Note: c_3 prunes T_{10} effectively only after "a" is pruned (by min-sup) in b's projected DB

Constrained Mining with Succinct Constraints

Succinctness: Pruning Both Data and Pattern Spaces

- Succinctness: If the constraint *c* can be enforced by directly manipulating the data
- Ex. 1: To find those patterns without item *i*
  - Remove *i* from DB and then mine (pattern space pruning)
- Ex. 2: To find those patterns containing item *i*
  - Mine only *i*-projected DB (data space pruning)
- Ex. 3: c_3: $min(S.Price) \leq v$ is succinct
  - Start with only items whose price $\leq$ v and remove transactions with high-price items only (pattern + data space pruning)
- Ex. 4: c_4: $sum(S.Price) \geq v$ is not succinct
  - It cannot be determined beforehand since sum of the price of itemset S keeps increasing

Constrained Mining with Convertible Constraints

Convertible Constraints: Ordering Data in Transactions

TID	Transaction
10	a, b, c, d, f, h
20	a, b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

min_sup = 2

Item	Price	Profit
a	100	40
b	40	0
c	150	−20
d	35	−15
e	55	−30
f	45	−5
g	80	30
h	10	5

- Convert tough constraints into (anti-)monotone by proper ordering of items in transactions
- Examine c_1: avg(*S*.profit) > 20
  - Order items in (profit) value-descending order
    - *<a, g, f, b, h, d, c, e>*
  - An itemset *ab* violates c_1 (avg(ab) = 20)
    - So does *ab** (i.e., *ab-projected DB)*
    - C_1: anti-monotone if patterns grow in the right order!
- Can item-reordering work for Apriori?
  - Level-wise candidate generation requires multi-way checking!
  - *avg*(*agf*) = 21.7 > 20, but avg(*gf*) = 12.5 < 20
  - Apriori will not generate “agf” as a candidate

Different Kinds of Constraints: Different Pruning Strategies

Different Kinds of Constraints Lead to Different Pruning Strategies

- In summary, constraints can be categorized as
  - Pattern space pruning constraints vs. data space pruning constraints
- Pattern space pruning constraints
  - Anti-monotonic: If constraint *c* is violated, its further mining can be terminated
  - Monotonic: If *c* is satisfied, no need to check *c* again
  - Succinct: If the constraint *c* can be enforced by directly manipulating the data
  - Convertible: *c* can be converted to monotonic or anti-monotonic if items can be properly ordered in processing
- Data space pruning constraints
  - Data succinct: Data space can be pruned at the initial pattern mining process
  - Data anti-monotonic: If a transaction *t* does not satisfy *c*, then *t* can be pruned to reduce data processing effort

Handling Multiple Constraints

How to Handle Multiple Constraints?

- It is beneficial to use multiple constraints in pattern mining
- But different constraints may require potentially conflicting item-ordering
  - If there exists conflict ordering between c_1 and c_2
    - Try to sort data and enforce *one constraint* first (which one?)
    - Then enforce the other constraint when mining the projected databases
- Ex. c_1: avg(*S*.profit) **>** 20, and c_2: avg(S.price) < 50
  - Assume c_1 has more pruning power
    - Sort in profit descending order and use c_1 first
  - For each project DB, sort trans. in price ascending order and use c_2 at mining

Constraint-Based Sequential-Pattern Mining

Constraint-Based Sequential-Pattern Mining

- Share many similarities with constraint-based itemset mining
- Anti-monotonic: If S violates *c*, the super-sequences of S also violate *c*
  - sum(S.price) < 150; min(S.value) > 10
- Monotonic: If S satisfies *c*, the super-sequences of S also do so
  - element_count (S) > 5; S $\supseteq$ {PC, digital_camera}
- Data anti-monotonic: If a sequence s_1 with respect to S violates c_3, s_1 can be removed
  - c_3: sum(S.price) ≥ v
- Succinct: Enforce constraint c by explicitly manipulating data
  - S $\supseteq$ {i-phone, MacAir}
- Convertible: Projection based on the sorted value not sequence order
  - value_avg(S) < 25; profit_sum (S) > 160
  - max(S)/avg(S) < 2; median(S) – min(S) > 5

Timing-Based Constraints in Seq.-Pattern Mining

- Order constraint: Some items must happen before the other
  - {algebra, geometry} → {calculus} (where "→" indicates ordering)
  - Anti-monotonic: Constraint-violating sub-patterns pruned
- Min-gap/max-gap constraint: Confines two elements in a pattern
  - E.g., mingap = 1, maxgap = 4
  - Succinct: Enforced directly during pattern growth
- Max-span constraint: Maximum allowed time difference between the 1^{st} and the last elements in the pattern
  - E.g., maxspan (S) = 60 (days)
  - Succinct: Enforced directly when the 1^{st} element is determined
- Window size constraint: Events in an element do not have to occur at the same time: Enforce max allowed time difference
  - E.g., window-size = 2: Various ways to merge events into elements

Episodes and Episode Pattern Mining

- Episodes and regular expressions: Alternative to seq. patterns
  - Serial episodes: $A \rightarrow B$
  - Parallel episodes: $A \mid B$

    Indicating partial order relationships
  - Regular expressions: $(A|B)C^*(D \rightarrow E)$
- Ex. Given a large shopping sequence database, one may like to find
  - A, B, C, D, E, such as it follows two constraints
  - Ordering following the template $(A|B)C^*(D \rightarrow E)$, and
  - Sum of the prices of A, B, C*, D, and E is greater than $100, where C* means C appears *-times
  - How to efficiently mine such sequential patterns?

Summary

Summary: Constraint-Based Pattern Mining

- Why Constraint-Based Mining?
- Different Kinds of Constraints: Different Pruning Strategies
- Constrained Mining with Pattern Anti-Monotonicity
- Constrained Mining with Pattern Monotonicity
- Constrained Mining with Data Anti-Monotonicity
- Constrained Mining with Succinct Constraints
- Constrained Mining with Convertible Constraints
- Handling Multiple Constraints
- Constraint-Based Sequential-Pattern Mining

Recommended Readings

- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques (3rd ed)*. Morgan Kaufmann. Chapter 7: Advanced Pattern Mining
- Ng, R., Lakshmanan, L.V.S.,Han, J., & Pang, A. (1998). Exploratory mining and pruning optimizations of constrained association rules. *SIGMOD'98.*
- Pei, J., Han, J., & Lakshmanan, L. V. S. (2001). Mining frequent itemsets with convertible constraints. *ICDE'01.*
- Pei, J., Han, J., & Wang, W. (2007). Constraint-based sequential pattern mining: The pattern-growth methods. *J. Int. Inf. Sys., 28*(2).

Additional References

- Bonchi, F., Giannotti, F., Mazzanti, A., & Pedreschi, D. (2003). ExAnte: Anticipated data reduction in constrained pattern mining. *PKDD'03*.
- Garofalakis, M. N., Rastogi, R., & Shim, L. (2002). Mining sequential patterns with regular expression constraints. *IEEE Trans. Knowl. Data Eng*, *14*(3).
- Grahne, G., Lakshmanan, L., & Wang, X. (2000). Efficient mining of constrained correlated sets. *ICDE'00.*
- Mannila, H., Toivonen, H., & Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*.
- Srikant, R., Vu, Q., & Agrawal, R. (1997). Mining association rules with item constraints. *KDD'97.*
- Zhu, F., Yan, X., Han, J., & Yu, P. S. (2007). gPrune: A constraint pushing framework for graph pattern mining. *PAKDD'07.*

Pattern Mining Applications: Mining Quality Phrases from Text Data

Pattern Mining Applications: Mining Quality Phrases from Text Data

- From Frequent Pattern Mining to Phrase Mining
- Previous Phrase Mining Methods
- ToPMine: Phrase Mining without Training Data
- SegPhrase: Phrase Mining with Tiny Training Sets
- AutoPhrase: Phrase Mining with Distant Supervision

Thanks to Ahmed El-Kishky@Facebook, Jialu Liu@Google Research, Jingbo Shang@UCSD, Xiang Ren@USC, Chi Wang@MSR, and Marina Danilevsky@IBM Research for their contributions

From Frequent Pattern Mining to Phrase Mining

Why Phrase Mining?

- Unigrams vs. phrases
  - **Unigrams** (single words) are often *ambiguous*
    - Example: “United”: United States? United Airline? United Parcel Service?
  - **Phrase**: A natural, meaningful, *unambiguous* semantic unit
    - Example: “United States” vs. “United Airline”
- Mining semantically meaningful phrases
  - Transform text data from *word granularity* to *phrase granularity*
  - Enhance the power and efficiency at manipulating unstructured data

From Frequent Pattern Mining to Phrase Mining

- General principle
  - Exploit information redundancy and data-driven criteria to determine phrase boundaries and salience
- Methodology: Exploring three ideas
  - Frequent pattern mining and colocation analysis
  - Phrasal segmentation
  - Quality phrase assessment
- Recent developments of phrase mining methods
  - ToPMine: Mining quality phrase without training (A. El-Kishky, et al., 2015)
  - SegPhrase: Mining quality phrase with tiny training sets (J. Liu, et al., 2015)
  - AutoPhrase: Mining quality phrases with distant supervision (e.g., Wikipedia) (Shang, et al., 2018)

Previous Phrase Mining Methods

Phrase Mining: Can We Reduce Annotation Cost?

- Phrase mining: Originated from the NLP community—"Chunking"
  - Model it as a sequence labeling problem (B-NP, I-NP, O, …)
- Need annotation and training
  - Annotate hundreds of documents as training data
  - Train a supervised model based on part-of-speech features
- Recent trend:
  - Use distributional features based on web n-grams (Bergsma et al., 2010)
  - State-of-the-art performance: ~95% accuracy, ~88% phrase-level F-score
- Limitations
  - High annotation cost, not scalable to a new language, a new domain/genre
  - May not fit domain-specific, dynamic, emerging applications
    - Scientific domains, query logs, or social media (e.g., Yelp and Twitter data)

Unsupervised Phrase Mining and Topic Modeling

- Many studies of unsupervised phrase mining are linked with topic modeling
- Topic modeling
  - Represents documents by multiple topics in different proportions
    - Each topic is represented by a word distribution
  - Does not require any prior annotations or labeling of the documents
- Statistical topic modeling algorithms
  - The most common algorithm: LDA (Latent Dirichlet Allocation) [Blei, et al., 2003]
- Three strategies on phrase mining with topic modeling
  - Strategy 1: Generate bag-of-words → generate sequence of tokens
  - Strategy 2: Post bag-of-words model inference, visualize topics with n-grams
  - Strategy 3: Prior bag-of-words model inference, mine phrases and impose on the bag-of-words model

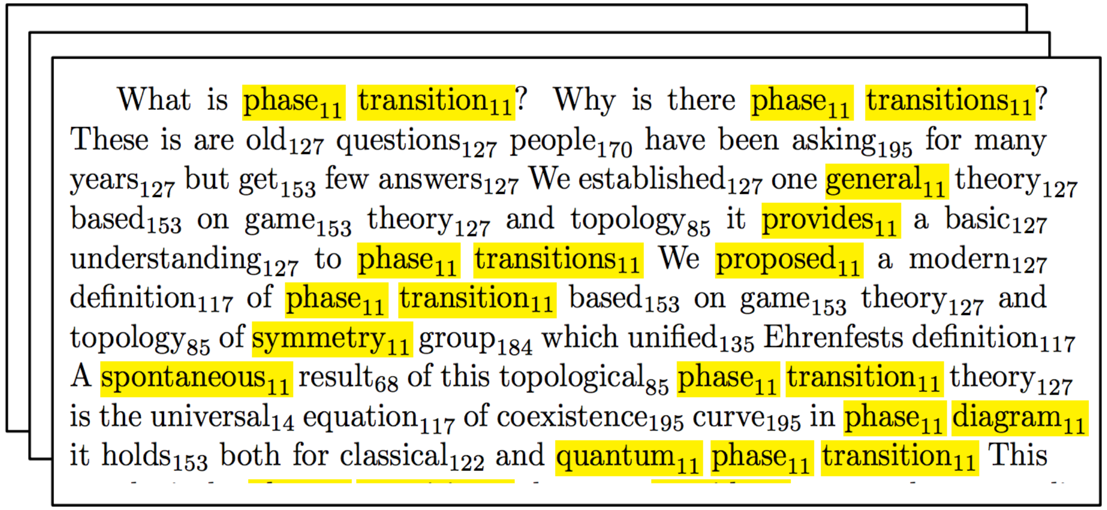
Strategy 1: Simultaneously Inferring Phrases and Topics

- Bigram Topic Model [Wallach'06]
  - Probabilistic generative model that conditions on previous word and topic when drawing next word
- Topical N-Grams (TNG) [Wang, et al.'07] (a generalization of Bigram Topic Model)
  - Probabilistic model that generates words in textual order
  - Create n-grams by concatenating successive bigrams
- Phrase-Discovering LDA (PDLDA) [Lindsey, et al.'12]
  - Viewing each sentence as a time-series of words, PDLDA posits that the generative parameter (topic) changes periodically
  - Each word is drawn based on previous m words (context) and current phrase topic
- Comments on this strategy
  - High model complexity: Tends to overfitting
  - High inference cost: Slow

Strategy 2: Post Topic-Modeling Phrase Construction (I): TurboTopics

- **TurboTopics** [Blei & Lafferty'09] – Phrase construction as a post-processing step to Latent Dirichlet Allocation
  - Perform Latent Dirichlet Allocation on corpus to assign each token a topic label
  - Merge adjacent unigrams with the same topic label by a distribution-free permutation test on arbitrary-length back-off model
  - End recursive merging when all significant adjacent unigrams have been merged

Annotated documents


What is phase$_{11}$ transition$_{11}$? Why is there phase$_{11}$ transitions$_{11}$? These is are old$_{127}$ questions$_{127}$ people$_{170}$ have been asking$_{195}$ for many years$_{127}$ but get$_{153}$ few answers$_{127}$ We established$_{127}$ one general$_{11}$ theory$_{127}$ based$_{153}$ on game$_{153}$ theory$_{127}$ and topology$_{85}$ it provides$_{11}$ a basic$_{127}$ understanding$_{127}$ to phase$_{11}$ transitions$_{11}$ We proposed$_{11}$ a modern$_{127}$ definition$_{117}$ of phase$_{11}$ transition$_{11}$ based$_{153}$ on game$_{153}$ theory$_{127}$ and topology$_{85}$ of symmetry$_{11}$ group$_{184}$ which unified$_{135}$ Ehrenfests definition$_{117}$ A spontaneous$_{11}$ result$_{68}$ of this topological$_{85}$ phase$_{11}$ transition$_{11}$ theory$_{127}$ is the universal$_{14}$ equation$_{117}$ of coexistence$_{195}$ curve$_{195}$ in phase$_{11}$ diagram$_{11}$ it holds$_{153}$ both for classical$_{122}$ and quantum$_{11}$ phase$_{11}$ transition$_{11}$ This

LDA topic #11

phase, transitions, phases, transition, quantum, critical, symmetry, field, point, model, order, diagram, systems, two, theory, system, study, breaking, spin, first

Turbo topic #11

phase transitions, model, symmetry, point, quantum, systems, phase transition, phase diagram, system, order, field, order, parameter, critical, two, transitions in, models, different, symmetry breaking, first order, phenomena

Post Topic-Modeling Phrase Construction (II): KERT

- **KERT** [Danilevsky et al.'14] – Phrase construction as a post-processing step to LDA
  - Run bag-of-words model inference and assign topic label to each token
  - Perform **frequent pattern mining** to extract candidate phrases within each topic
  - Perform **phrase ranking** based on four different criteria
    - **Popularity:** e.g., "information retrieval" vs. "cross-language information retrieval"
    - **Concordance**
      - "powerful tea" vs. "strong tea"
      - "active learning" vs. "learning classification"
    - **Informativeness:** e.g., "this paper" (frequent but not discriminative, not informative)
    - **Completeness:** e.g., "vector machine" vs. "support vector machine"

Comparability property: directly compare phrases of mixed lengths

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	5

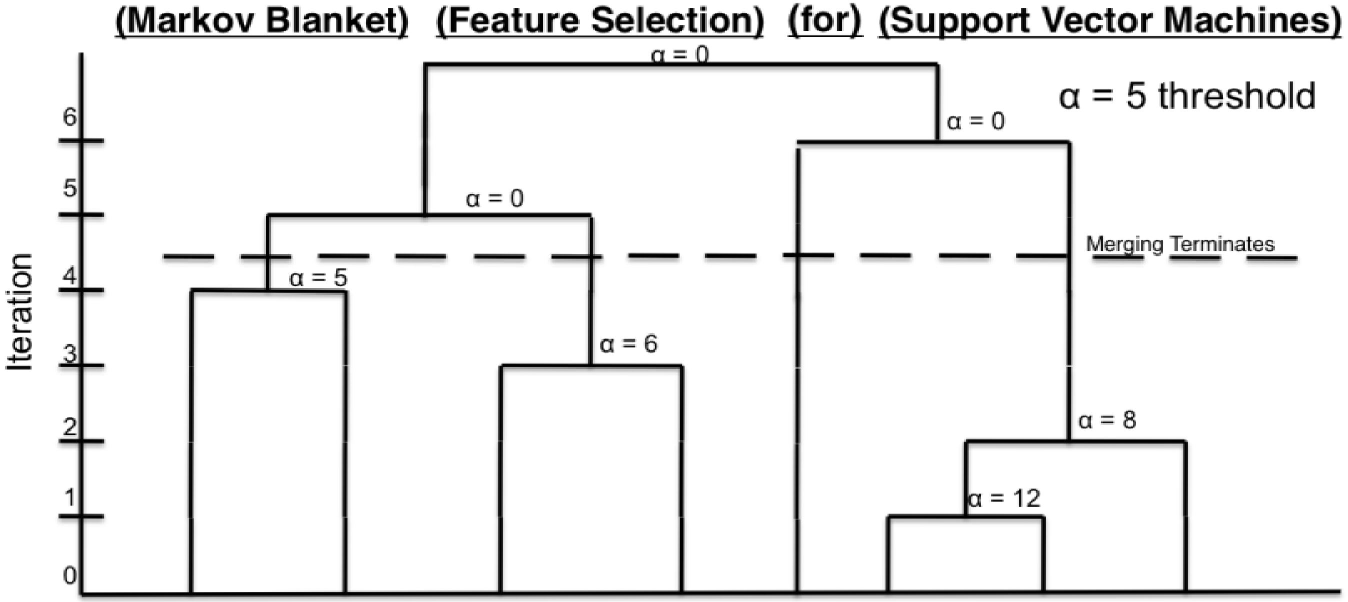
Collocation Mining

- 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

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

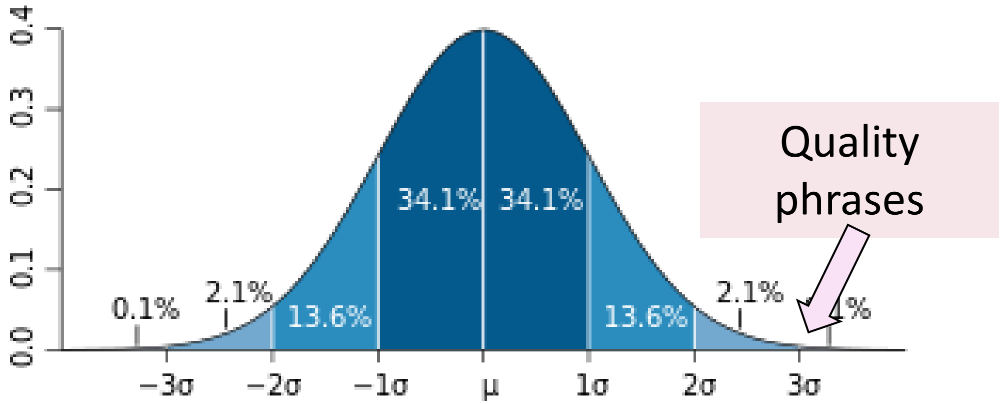
- Many of these measures can be used to guide the agglomerative **phrase-segmentation** algorithm

Phrase Candidate Generation: Frequent Pattern Mining + Statistical Analysis



Markov Blanket Feature Selection for Support Vector Machines.

[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]:

$$\alpha(P_1, P_2) \approx (f(P_1 \bullet P_2) - \mu_0(P_1, P_2)) / \sqrt{f(P_1 \bullet P_2)}$$

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

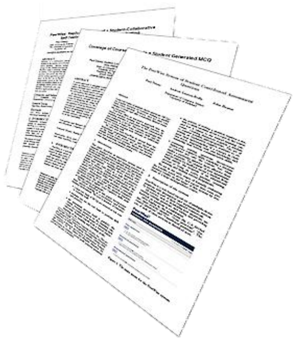
SegPhrase: Phrase Mining with Tiny Training Sets

SagPhrase: Phrase Mining with Tiny Training Sets

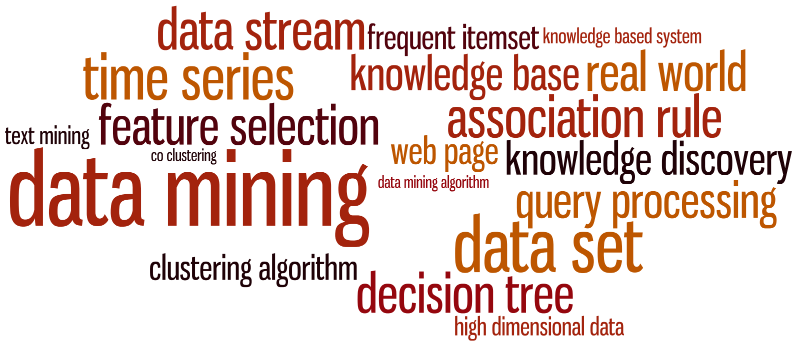
- A small set of training data may enhance the quality of phrase mining

J. Liu et al., Mining Quality Phrases from Massive Text Corpora. In *SIGMOD'*15

Raw Corpus



Quality Phrases



Segmented Corpus

Document 1
Citation recommendation is an interesting but challenging research problem in data mining area.

Document 2
In this study, we investigate the problem in the context of heterogeneous information networks using data mining technique.

Document 3
Principal Component Analysis is a linear dimensionality reduction technique commonly used in machine learning applications.

+ A small set of labels by human or a general KB

Input Raw Corpus → **Quality Phrases** ↔ **Segmented Corpus**

Phrase Mining **Phrasal Segmentation**

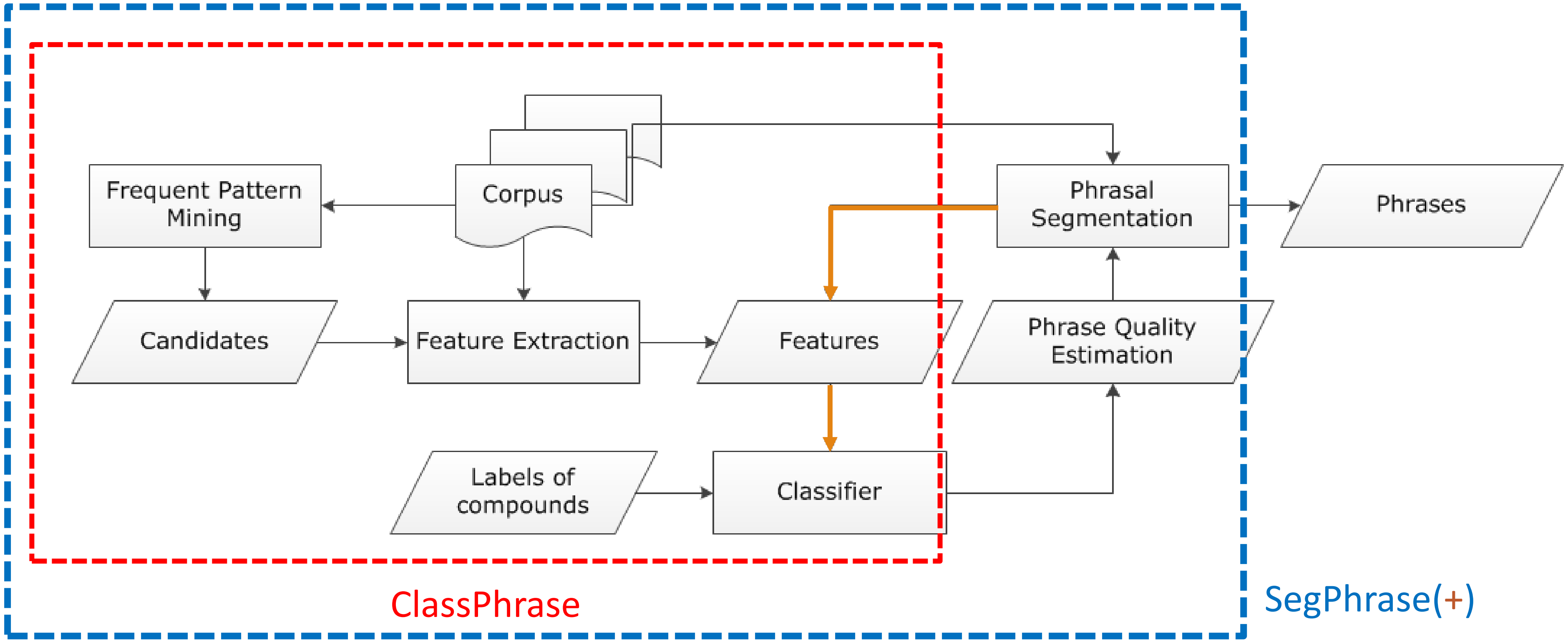
Integrating phrase mining with phrasal segmentation and classification

SegPhrase+: The Overall Framework

- ClassPhrase: Frequent pattern mining, feature extraction, classification
- SegPhrase: Phrasal segmentation and phrase quality estimation
- SegPhrase+: One more round to enhance mined phrase quality

SegPhrase (a classifier is used)

Small labeled dataset provided by experts
or
a distant supervised KB (e.g., Wikipedia / DBPedia)



SegPhrase: Pattern Mining and Feature Extraction

- **Pattern Mining for Candidate Set**
  - Build a candidate phrases set by frequent pattern mining
    - Mining frequent *k*-grams (*k* is typically small, e.g., 6 in the experiments)
    - **Popularity** measured by *raw* frequent words and phrases mined from the corpus
- **Feature Extraction: Concordance**
  - Partition a phrase into two parts to check whether the co-occurrence is significantly higher than pure random
- **Feature Extraction: Informativeness**
  - Quality phrases typically start and end with a non-stopword
    - "machine learning is" vs. "machine learning"
  - Use average IDF over words in the phrase to measure the semantics
  - Usually, the probabilities of a quality phrase in quotes, brackets, or connected by hyphen should be higher (punctuations information)
    - e.g., "state-of-the-art"

SegPhrase: Classification Using Tiny Training Sets

- Use tiny training sets (300 labels for 1GB corpus; can also use phrases extracted from KBs)
  - Label: indicating whether a phrase is a high quality one
    - E.g., "support vector machine": 1; "the experiment shows": 0
- Classification: Construct models to distinguish quality phrases from poor ones
  - Use *Random Forest* algorithm to bootstrap different datasets with limited labels
- Phrasal segmentation can tell which phrase is more appropriate
  - Ex: "A standard [feature vector] [machine learning] setup is used to describe ....."

    Not counted towards the rectified frequency
  - Partition a sequence of words by maximizing the likelihood
  - Consider length penalty and filter out phrases with low rectified frequency
- Process: Classification → Phrasal segmentation // SegPhrase

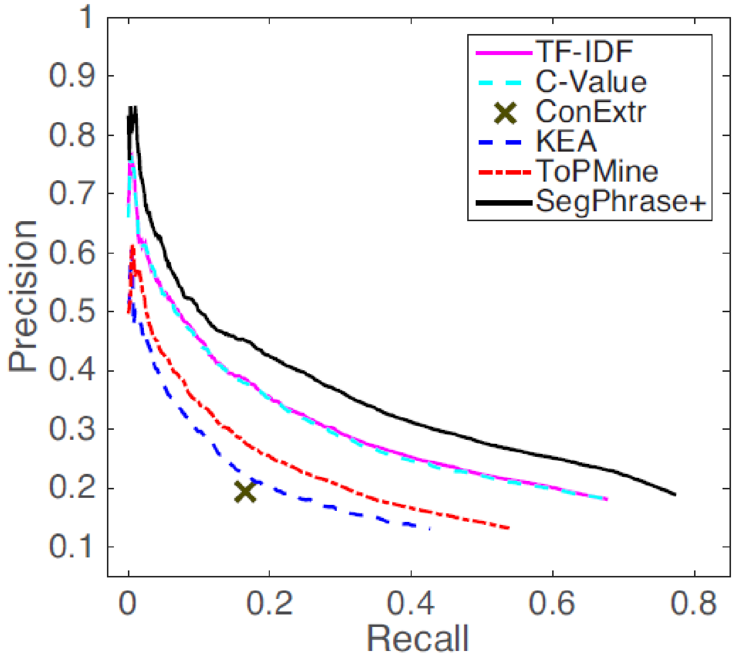
  → Classification → Phrasal segmentation // SegPhrase+

Performance: Precision Recall Curves on DBLP

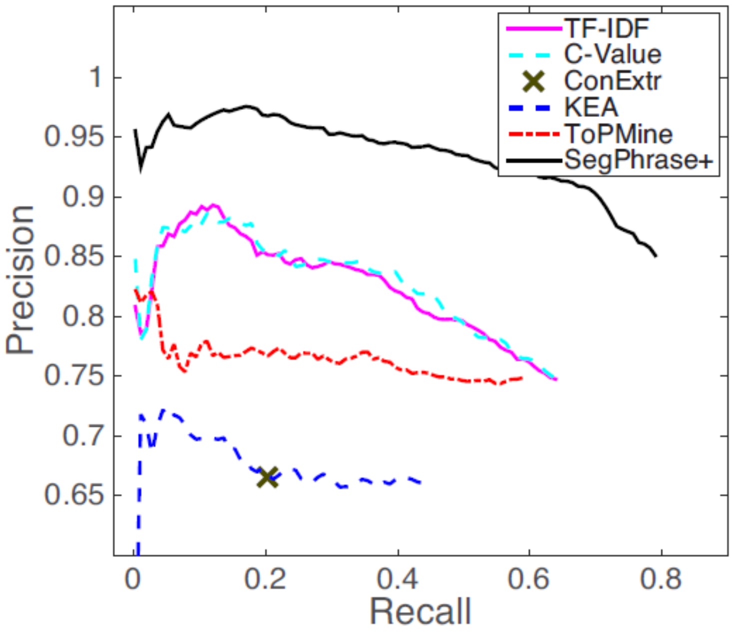
- Datasets:
- Evaluation
  - Wiki Phrases (based on internal links, ~7K high quality phrases)
  - Sampled 500*7 Wiki-uncovered phrases: Results evaluated by 3 reviewers
- Compared with other phrase-mining methods
  - TF-IDF, C-Value, ConExtr, KEA, and ToPMine
- Also, Segphrase+ is efficient, linearly scalable

Dataset	#docs	#words	#labels
DBLP	2.77M	91.6M	300
Yelp	4.75M	145.1M	300

Use only 300 human labeled phrases for training



Precision-Recall Curves on DBLP Data (Wiki Phrases)



Precision-Recall Curves on DBLP Data (Non Wiki-phrases)

Experimental Results: Interesting Phrases Generated (From Titles & Abstracts of SIGKDD)

Query	SIGKDD	
Method	SegPhrase+	Chunking (TF-IDF & C-Value)
1	data mining	data mining
2	data set	association rule
3	association rule	knowledge discovery
4	knowledge discovery	frequent itemset
5	**time series**	decision tree
…	…	…
51	association rule mining	search space
52	rule set	domain knowledge
53	concept drift	**important problem**
54	knowledge acquisition	concurrency control
55	**gene expression data**	conceptual graph
…	…	…
201	web content	optimal solution
202	**frequent subgraph**	semantic relationship
203	intrusion detection	**effective way**
204	**categorical attribute**	space complexity
205	user preference	**small set**
…	…	…

Only in SegPhrase+

Only in Chunking

Mining Quality Phrases in Multiple Languages

- Both ToPMine and SegPhrase+ are extensible to mining quality phrases in multiple languages
- SegPhrase+ on Chinese (From Chinese Wikipedia)
- ToPMine on Arabic (From Quran (Fus7a Arabic)(no preprocessing)
  - Experimental results of Arabic phrases:

كفروا→ Those who disbelieve

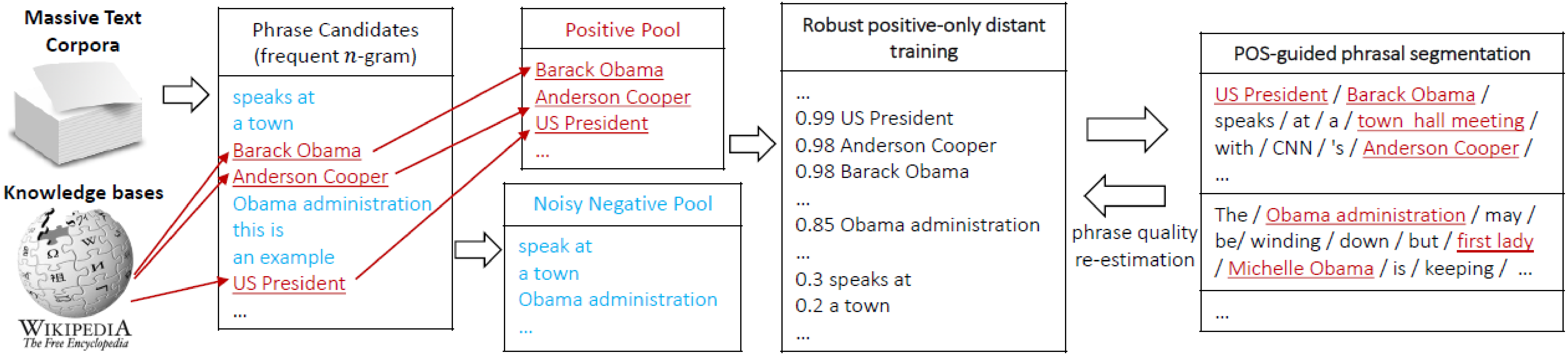
بسم الله الرحمن الرحيم→ In the name of God the Gracious and Merciful

Rank	Phrase	In English
…	…	…
62	首席_执行官	CEO
63	中间_偏右	Middle-right
…	…	…
84	百度_百科	Baidu Pedia
85	热带_气旋	Tropical cyclone
86	中国科学院_院士	Fellow of Chinese Academy of Sciences
…	…	…
1001	十大_中文_金曲	Top-10 Chinese Songs
1002	全球_资讯网	Global News Website
1003	天一阁_藏_明代_科举_录_选刊	A Chinese book name
…	…	…
9934	国家_戏剧_院	National Theater
9935	谢谢_你	Thank you
…	…	…

AutoPhrase: Phrase Mining with Distant Supervision

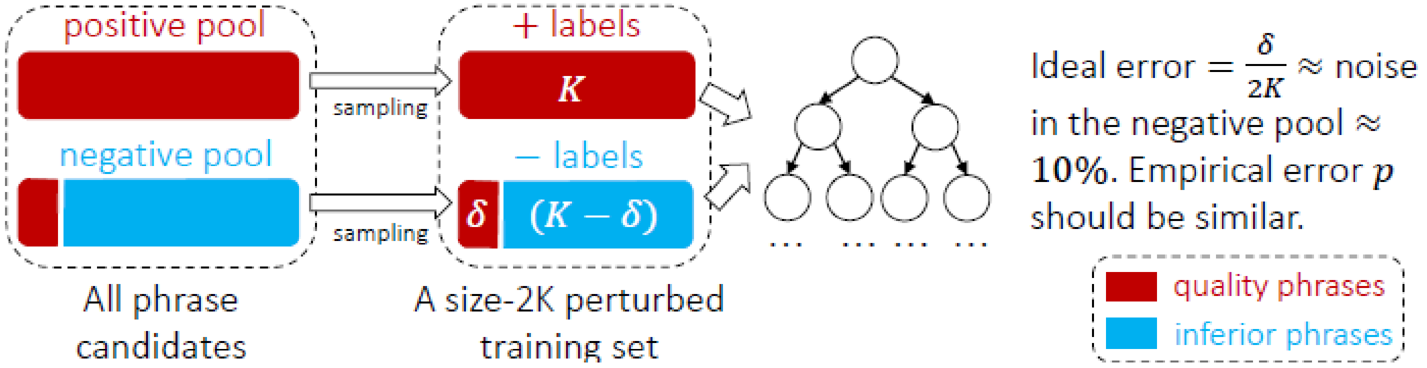
AutoPhrase: Automated Phrase Mining by Distant Supervision

- AutoPhrase: *Automatic* extraction of high-quality phrases (e.g., scientific terms and general entity names) in a given corpus (e.g., research papers and news)
- **Major features**:
  - No human efforts; multiple languages; high performance—precision, recall, efficiency
  - **Distant training**: Utilize quality phrases in KBs (e.g., Wiki) as *positive* phrase labels
- **Innovation:** Sampling-based label generation for robust, positive-only distant training



Robust Positive-Only Distant Training

- In each base classifier, randomly sample K *positive* (e.g., wiki titles, keywords, links) and K *noisy negative labels* from the pools

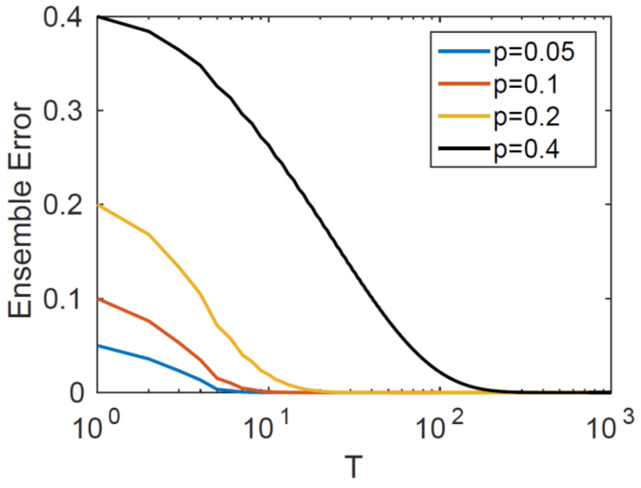


- Noisy negative pool: may still have δ quality phrases among the K negative labels
- They form "perturbed training set": size-2K subset of the full set of all phrases where the labels of some quality phrases are switched from positive to negative
- Each base classifier can be viewed as randomly drawn K phrase candidates with replacement from the positive pool and the negative pool respectively
  - Grow an unpruned decision tree to the point of separating all phrases to meet this requirement
- Use an *ensemble classifier* that averages the results of independently trained base classifiers

Why Is Positive-Only Distant Training Robust?

- Theoretical Analysis
  - T base classifiers

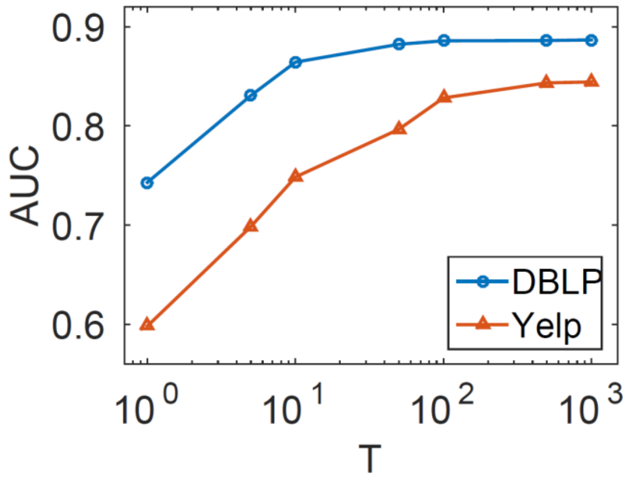
$$\text{ensemble\_error}(T) = \sum_{t=\lfloor 1+T/2 \rfloor}^{T} \binom{T}{t} p^t (1-p)^{T-t}$$



  - Exponentially decreasing

- Empirical Performance
  - AUC to evaluate the ranking

Note: AUC (Area Under Curve), with value range [0,1], is a classification measure to be introduced in the classification module

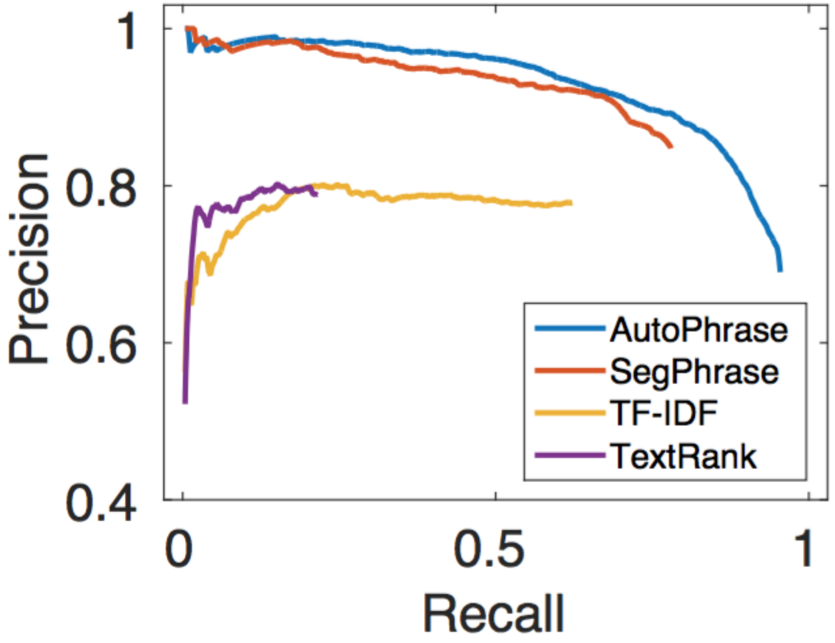


Modeling Single-Word Phrases: Enhancing Recall

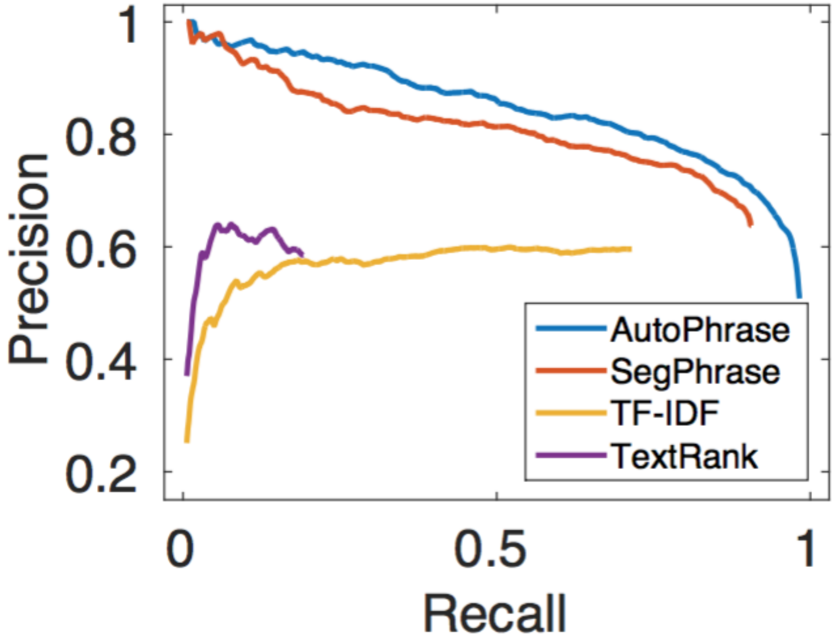
- AutoPhrase simultaneously models single-word and multi-word phrases
  - A phrase can also be a single word, as long as it functions as a constituent in the syntax of a sentence, e.g., "UIUC", "Illinois"
  - Based on our experiments: 10%~30% quality phrases are single-word phrases
- Criteria for modeling single-word phrases
  - **Popularity**: Sufficiently frequent in a given corpus
  - **Informativeness**: Indicative of a specific topic or concept
  - **Independence**: A quality single-word phrase is more likely a complete semantic unit in a given document
- Example: Is the following good single-word phrase?
  - "CMU"? Yes (frequent, informative, independent)
  - "this"? No (not informative)
  - "united"? No (not independent, may be in "United States", "United Airline",…)

AutoPhrase: Cross-Domain Evaluation Results

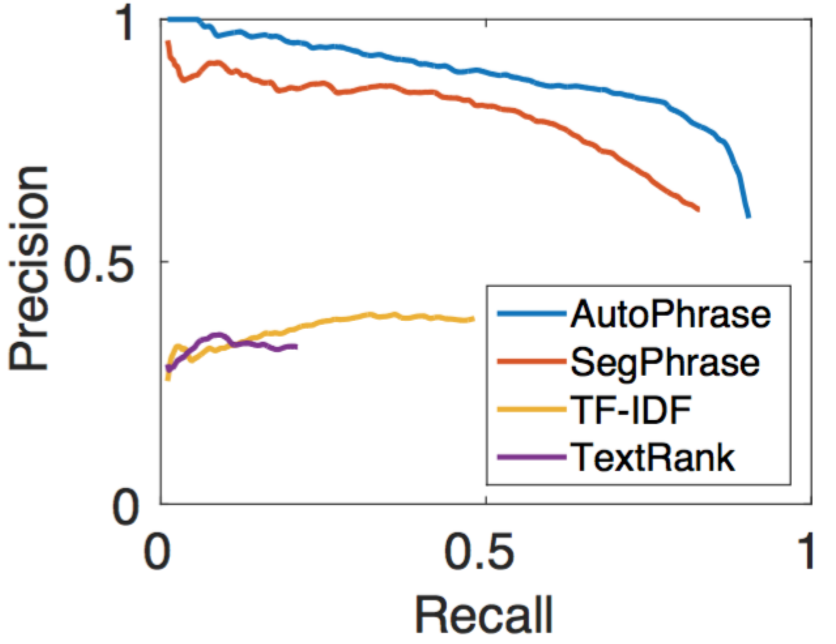
Computer Science Papers



Yelp Business Reviews



Wikipedia Articles



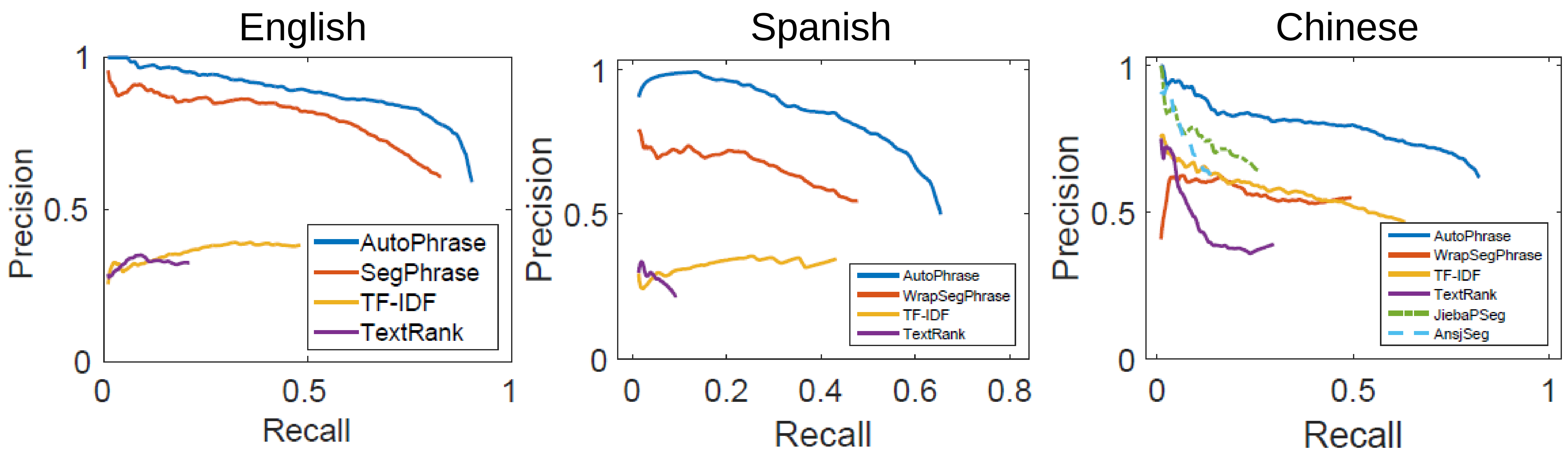
AutoPhrase (TKDE'18): Best performing and generating multi-word and single word phrases

SegPhrase (SIGMOD'15): Outperformed TopMine (VLDB'15) and many other methods

TF-IDF: Stanford NLP Parser (LREC'16) + Ranked by TF-IDF

TextRank (ACL'04): Stanford NLP Parser (LREC'16) + Ranked by TextRank

AutoPhrase: Cross-Language Evaluation Results



AutoPhrase (TKDE'18): Best performing and generating multi-word and single word phrases

WrapSegPhrase: non-English characters → English letters & SegPhrase

JiebaSeg: Specifically for Chinese; Dictionaries & Hidden Markov Models

AnsjSeg: Specifically for Chinese; Dictionaries & Conditional Random Fields

AutoPhrase: An Example Run From Chinese Wikipedia

Phrase's Rank	Phrase	Translation (Explanation)
1	江苏_舜_天	(the name of a soccer team)
2	苦_艾_酒	Absinthe
3	白发_魔_女	(the name of a novel/TV-series)
4	笔记_型_电脑	notebook computer, laptop
99,994	计算机_科学技术	Computer Science and Technology
99,995	恒_天然	Fonterra (a company)
99,996	中国_作家_协会_副_主席	The Vice President of Writers Association of China
99,997	维他命_b	Vitamin B
99,998	舆论_导向	controlled guidance of the media
…	…	…

- The size of positive pool is about 29,000
- AutoPhrase finds more than 116,000 quality phrases (quality score > 0.5)

Summary

Summary: Pattern Mining Applications: Mining Quality Phrases from Text Data

- From Frequent Pattern Mining to Phrase Mining
- Previous Phrase Mining Methods
- New Methods that Integrate Pattern Mining with Phrase Mining
  - ToPMine: Phrase Mining without Training Data
- SegPhrase: Phrase Mining with Tiny Training Sets
- AutoPhrase: Phrase Mining with Distant Supervision

Recommended Readings

- S. Bergsma, E. Pitler, D. Lin, Creating Robust Supervised Classifiers via Web-scale N-gram Data, ACL'2010
- D. M. Blei and J. D. Lafferty. Visualizing Topics with Multi-word Expressions. arXiv:0907.1013, 2009
- D.M. Blei, A. Y. Ng, M. I. Jordan, J. D. Lafferty, Latent Dirichlet Allocation. JMLR 2003
- M. Danilevsky, C. Wang, N. Desai, X. Ren, J. Guo, J. Han. Automatic Construction and Ranking of Topical Keyphrases on Collections of Short Documents. SDM'14
- A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han. Scalable Topical Phrase Mining from Text Corpora. VLDB'15
- R. V. Lindsey, W. P. Headden, III, M. J. Stipicevic. A Phrase-Discovering Topic Model Using Hierarchical Pitman-Yor Processes. EMNLP-CoNLL'12.
- J. Liu, J. Shang, C. Wang, X. Ren, J. Han, Mining Quality Phrases from Massive Text Corpora. SIGMOD'15
- A. Parameswaran, H. Garcia-Molina, and A. Rajaraman. Towards the Web of Concepts: Extracting Concepts from Large Datasets. VLDB'10
- X. Wang, A. McCallum, X. Wei. Topical N-grams: Phrase and Topic Discovery, With and Application to Information Retrieval. ICDM'07
- J. Shang, J. Liu, M. Jiang, X. Ren, C. R Voss, J. Han, "Automated Phrase Mining from Massive Text Corpora", IEEE Transactions on Knowledge and Data Engineering, 30(10):1825-1837 (2018)