

# Phrase Mining

# Unigrams, N-grams and Phrases

## □ Unigrams, n-graphs, and phrases

- **Unigrams** (single words) are *ambiguous (or unclear)*
  - Ex.: “United”: United States? United Airline? United Parcel Service?
- **N-grams**: A contiguous sequence of **n** items from a given sample of text or speech
  - Ex. How many bi-grams or tri-grams in this paper title?
    - “Mining Frequent Patterns without Candidate Generation”
- **Phrase**: A natural, meaningful, *unambiguous* semantic unit
  - Ex.: “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 using database technology

# Why Phrase Mining?



## w/o phrase mining

- What's "United"?

- Applications in NLP, IR, Text Mining
  - Document analysis
  - Indexing in search engine



## w/ phrase mining

-  United Airline!

- Keyphrases for topic modeling
- Summarization

# Definition: Quality Phrase Mining

- Quality phrase mining seeks to extract ***a ranked list of phrases*** with decreasing quality from ***a large collection of documents***
- Examples:

Scientific  
Papers



## Expected Results

data mining  
machine learning  
information retrieval  
...  
support vector  
machine  
...  
the paper  
...

News  
Articles



## Expected Results

US President  
Anderson Cooper  
Barack Obama  
...  
Obama care  
...  
a town  
...

# What Kind of Phrases Are of “High Quality”?

- **Popularity:** Frequency
  - “information retrieval” vs. “cross-language information retrieval”
- **Concordance:** A sequence of words that occur more frequently than expected
  - “successful” vs. “strong” vs. “active learning” vs. “learning classification”

Concordance can be measured using many statistical measures, e.g., significance score, mutual information, t-test, z-test, chi-squared test, likelihood ratio, ...

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

- **Informativeness**
  - “this paper” (frequent but not discriminative, not informative)
- **Completeness**
  - “vector machine” vs. “support vector machine”

# Phrase Mining — Families of Methods

Supervised (linguistic analyzers)

Unsupervised (Exploring statistical signals)

Weakly Supervised (Human provides a small set of labeled data)

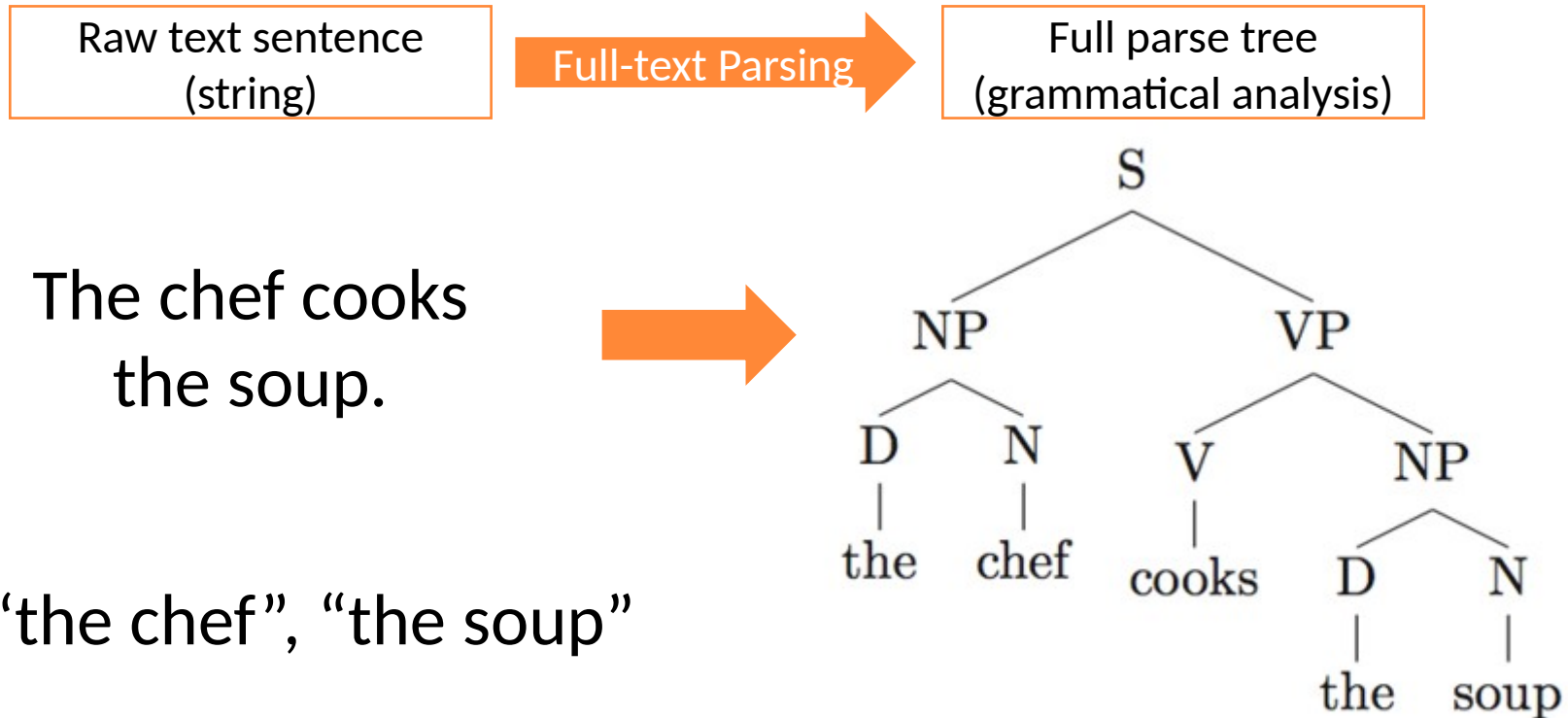
Distantly Supervised (Exploring Knowledge-Bases, e.g., Wikipedia)

# Supervised Phrase Mining

- Phrase mining was originated from the NLP community
- How to use linguistic analyzers to extract phrases?
  - Parsing (e.g., stanford NLP parsers)
  - Noun Phrase (NP) Chunking
- How to rank extracted phrases?
  - C-value [Frantzi et al.'00]
  - TextRank [Mihalcea et al.'04]
  - TF-IDF

# Linguistic Analyzer – Parsing

- Minimal Grammatical Segments  Phrases

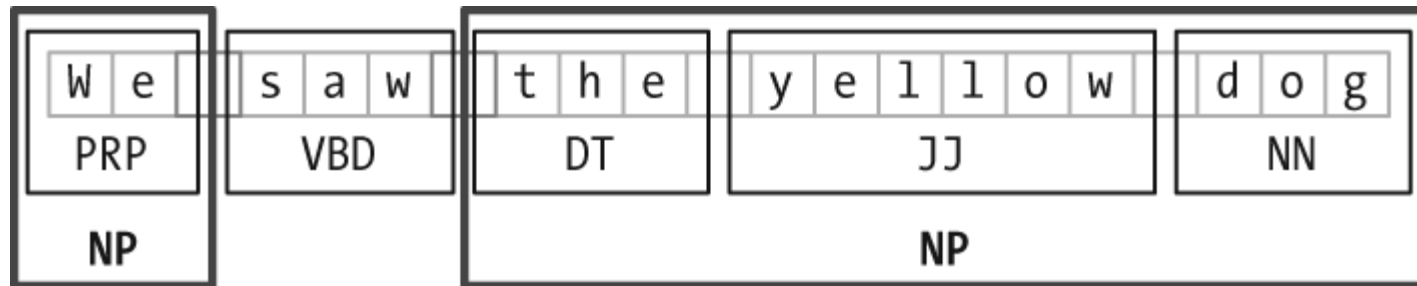


- Phrases: “the chef”, “the soup”



# Linguistic Analyzer – Chunking

- Parsing can be slow
- We need “shallow” phrase mining techniques
- Noun phrase chunking is a light version of parsing
  - Apply tokenization and part-of-speech (POS) tagging to each sentence
  - Search for noun phrase chunks



# A Side Topic - POS

- In traditional grammar, a part of speech is a category of words (or, more generally, of lexical items) which have similar grammatical properties
- In NLP, part-of-speech tagging, also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech based on both its definition and its context, i.e., its relationship with adjacent and related words in a phrase, sentence, or paragraph

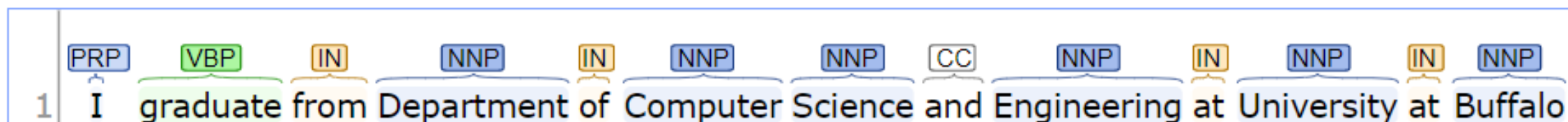
# Tags used in Penn Treebank

- Nine common parts of speech in English: noun, verb, article, adjective, preposition, pronoun, adverb, conjunction, and interjection.
- Most NLP researchers use Penn Treebank tags, which are finer than common English POSes mentioned above.
- [https://www.ling.upenn.edu/courses/Fall\\_2003/ling001/penn\\_treebank\\_pos.html](https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html)

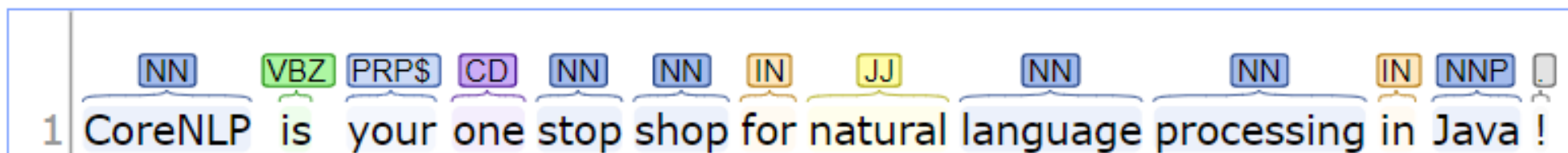
# Example

## Part-of-Speech:

1 I graduate from Department of Computer Science and Engineering at University at Buffalo



1 CoreNLP is your one stop shop for natural language processing in Java !



# Probabilistic Model for Tagging

- Probabilistic approach: there are many sequences of tags, but only one yields (i.e.,  $\text{argmax}$ ) the highest probability.

	Natural	Language	Processing	is	a	field	of	computer	science.
$T_1$	aj.	n.	n.	v.	dt.	n.	cj.	n.	n.
$T_2$	n.	n.	n.	v.	n.	n.	cj.	n.	n.
$T_3$	v.	n.	av.	v.	dt.	n.	cj.	n.	n.
$T_4$	dt.	n.	n.	v.	v.	n.	aj.	v.	n.
$T_5$	cj.	n.	n.	v.	dt.	n.	cj.	n.	n.

$T_2 - T_5$  apparently make no sense and hence their  $P()$ 's are very low.

- The goal is to find the most likely sequence of tags (T), given the sequence of words (W), i.e.,

# A Generative Model for POS tagging

- tag-to-tag transition probabilities:

	Natural	Language	Processing	is	a	field	of	computer	science.
$T_1$	aj.	n.	n.	v.	dt.	n.	cj.	n.	n.

- tag-to-word emission probabilities:

# A Generative Model for POS tagging

- A sequence of words is generated in two phases:
  - Produce a sequence of tags, e.g.,  $\triangle$ , NN, DET, ..., based on probability between each two consecutive tags.
  - For each tag, produce a word, e.g., NN  $\rightarrow$  “language”, DT  $\rightarrow$  “the”.
- Hidden Markov Model (HMM)
  - a tag is a state and a word is an observation
  - In each state/tag, a word/observation emits. After each emission, transit to the next state/tag and emit a word/observation again.
  - The transition and emission probabilities can be obtained by scanning the corpus once
  - In principle, we just need to enumerate all possible tag sequences,  $T_1, T_2, \dots$  and find the one that yields the largest  $P(W|T)P(T)$ .

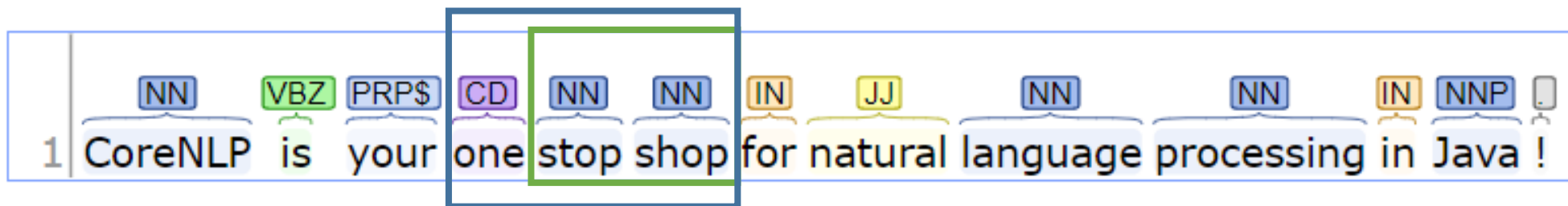
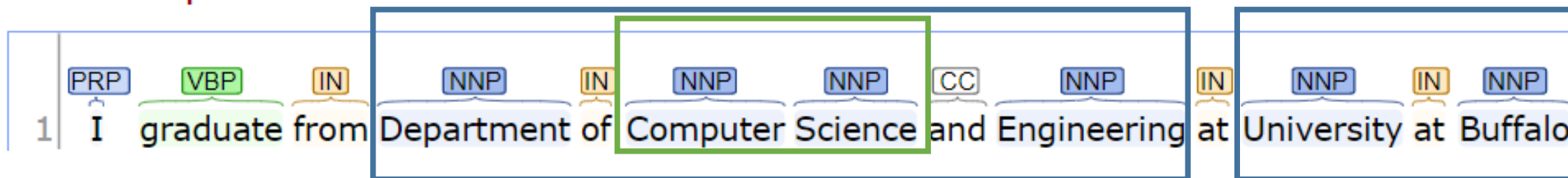
# Viterbi Algorithm

- Dynamic programming
- The Viterbi algorithm operates by iteratively computing the highest probability paths to each state at each time step, storing these probabilities, and backtracking to determine the most probable sequence of hidden states.
  - speech recognition
  - align DNA sequences
  - named entity recognition
- <https://www.geeksforgeeks.org/viterbi-algorithm-for-hidden-markov-models-hmms/>



# Back to Phrase Chunking

Part-of-Speech:



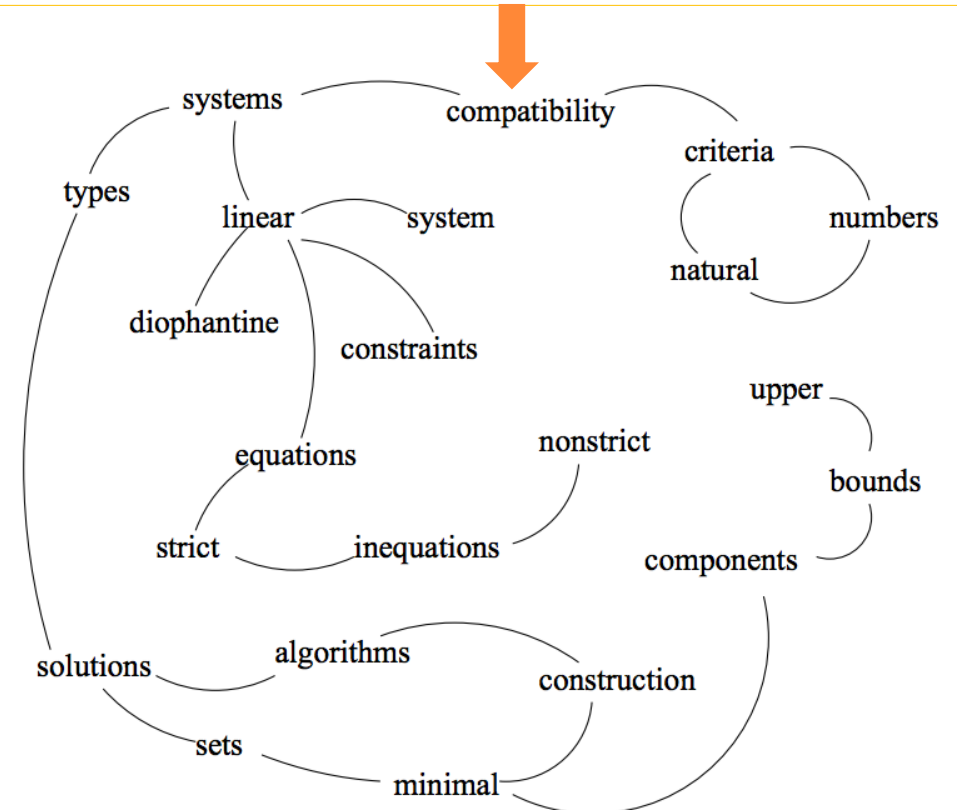
# Inefficiencies of Linguistic Analyzer

- Difficult to directly apply pre-trained to new domains (e.g. twitter, biomedical, yelp)
  - Unless sophisticated, manually curated, domain-specific training data are provided
- Computationally slow
  - Cannot be applied on web-scale data to support emerging applications
- Lack of the usage of corpora-level information
  - NP sometimes cannot meet the requirements of quality phrases

# Ranking

- C-Value
  - Prefers “maximal” phrases
  - Popularity & Completeness
- TextRank
  - Similar to PageRank
  - Popularity & Informativeness
- TF-IDF
  - Term Frequency
  - Inverse Document Frequency
  - Popularity & Informativeness

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. ....



# Phrase Mining — Families of Methods

Supervised (linguistic analyzers)

Unsupervised (Exploring statistical signals)

Weakly Supervised (Human provides a small set of labeled data)

Distantly Supervised (Exploring Knowledge-Bases, e.g., Wikipedia)

# Unsupervised Phrase Mining

- Statistics based on massive text corpora
  - Popularity
    - Raw frequency, or frequency distribution based on Zipfian ranks [Deane'05]
  - Concordance
    - Significance score [Church et al.'91][El-Kishky et al.'14]
  - Completeness
    - Comparison to super/sub-sequences [Parameswaran et al.'10]

# Mining Phrases: Why Not Use Raw Frequency Based Methods?

- Traditional data-driven approaches
  - Frequent pattern mining
    - If AB is frequent, likely AB could be a phrase
- Raw frequency could NOT reflect the quality of phrases
  - E.g.,  $\text{frequency}(\text{vector machine}) \geq \text{frequency}(\text{support vector machine})$
  - Need to rectify the frequency based on segmentation results
- Phrasal segmentation will tell
  - Some words should be treated as a whole phrase whereas others are still unigrams

# Pointwise Mutual Information (PMI)

$$PMI(x; y) = \log \frac{p(x, y)}{p(x)p(y)}$$

PMI of concrete occurrences of x and y.

- Question: if two words x and y appear together all the time, the PMI is
  - High? Low?
- Normalized PMI (NPMI)

$$NPMI(x; y) = \frac{\log \frac{p(x, y)}{p(x)p(y)}}{-\log p(x, y)}$$

Normalized Pointwise Mutual Information of x and y.


# PMI

- PMI maximizes when  $X$  and  $Y$  are perfectly associated
- NPMI: normalized PMI
  - -1 (in the limit) for never occurring together
  - 0 for independence
  - +1 for complete co-occurrence



# Word2vec phrase score

discounting coefficient for frequency threshold

$$score(x; y) = \frac{count(x, y) - \delta}{count(x) * count(y)}$$


# Difficulties: Comparison to Super/Sub-sequences

- Frequency ratio between an n-gram phrase and its two (n-1)-gram phrases

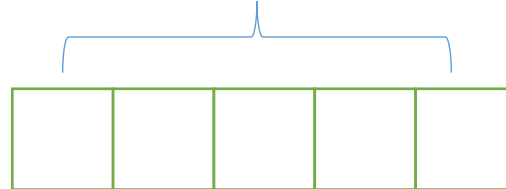
- Example

- **Pre-confidence** of *San Antonio*: 2385 / 14585
  - **Post-confidence** of *San Antonio*: 2385 / 2855
  - Expand/terminate based on thresholds

Phrase	Raw frequency
San	14585
Antonio	2855
San Antonio	2385

- Assumption

An n-gram quality phrase



Two (n-1)-gram sub-phrases



At least one of them is not a quality phrase

- Counter-example

- “*relational database system*” is a quality phrase
  - Both “*relational database*” and “*database system*” can be quality phrases

# Limitations of Statistical Signals

- The thresholds should be carefully chosen
- Only consider a subset of quality phrase requirements
- Combining different signals in an unsupervised manner is difficult
  - **Introducing some supervision may help!**

# Phrase Mining — Families of Methods

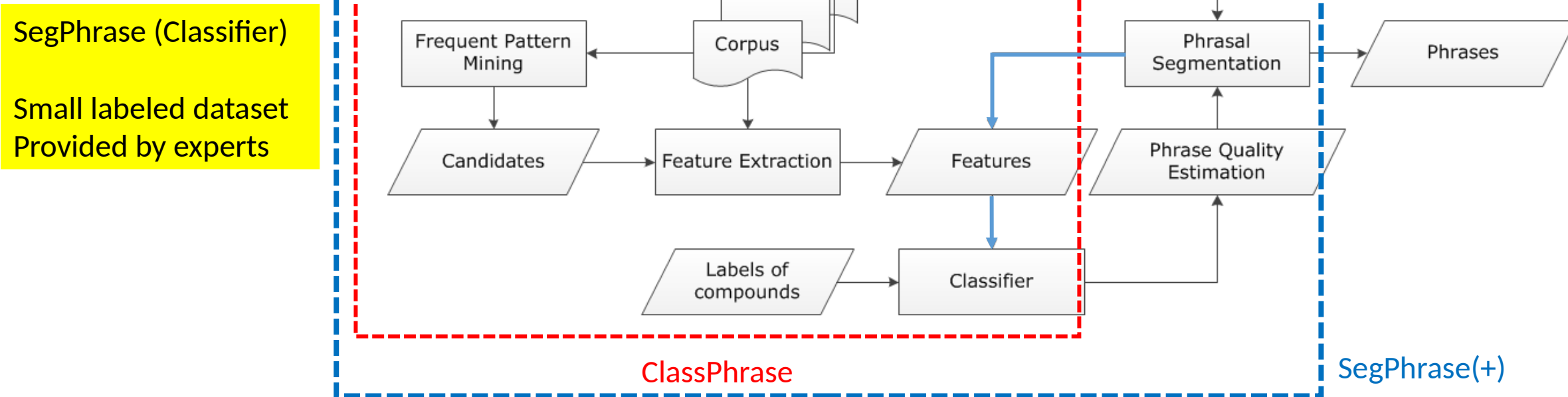
Supervised (linguistic analyzers)

Unsupervised (Exploring statistical signals)

Weakly Supervised (Human provides a small set of labeled data)

Distantly Supervised (Exploring Knowledge-Bases, e.g., Wikipedia)

# SegPhrase+: Exploring Training Data to Enhance Quality



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

# ClassPhrase I: 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 experiments)
- **Popularity** measured by *raw* frequent words and phrases mined from the corpus

# ClassPhrase II: Feature Extraction: Concordance

- Partition a phrase into two parts to check whether the co-occurrence is significantly higher than pure random

- support vector machine  
 $u_l \quad u_r$

this paper demonstrates  
 $u_l \quad u_r$

$$\langle u_l, u_r \rangle = \arg \min_{u_l \oplus u_r = v} \log \frac{p(v)}{p(u_l)p(u_r)}$$

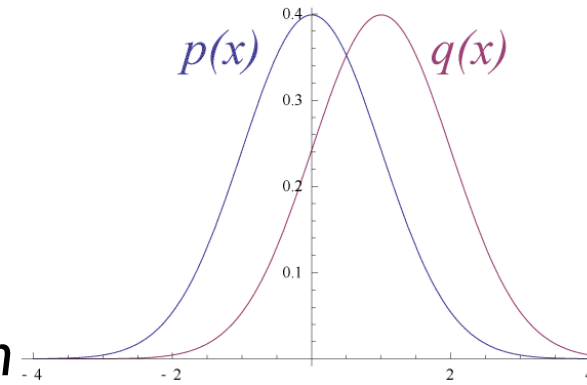
- Pointwise mutual information:  $PMI(u_l, u_r) = \log \frac{p(v)}{p(u_l)p(u_r)}$

- Pointwise KL divergence:  $PKL(v \| \langle u_l, u_r \rangle) = p(v) \log \frac{p(v)}{p(u_l)p(u_r)}$

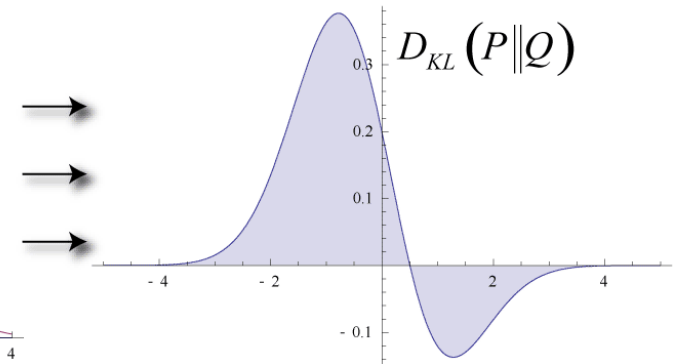
- The additional  $p(v)$  is multiplied with pointwise mutual information, leading to less bias towards rare-occurred phrases

# KL Divergence: Comparing Two Probability Distributions

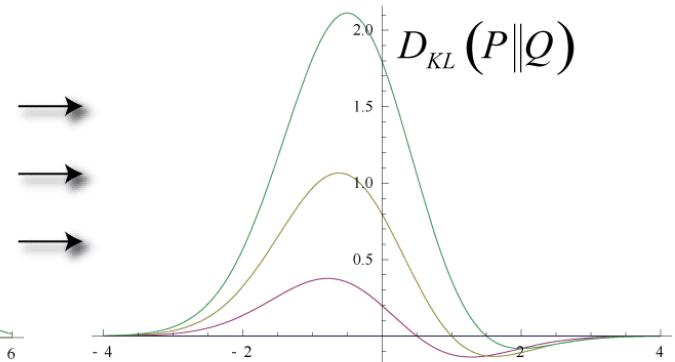
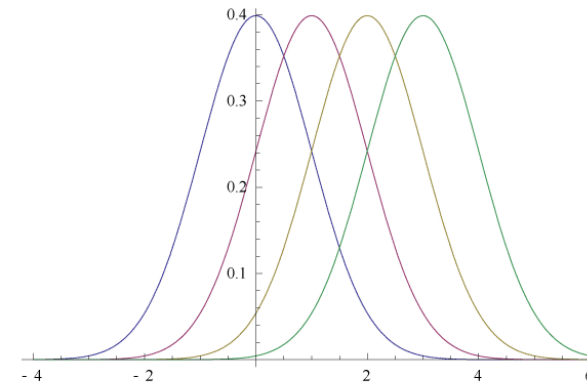
- The Kullback-Leibler (KL) divergence: Measure the difference between two probability distributions over the same variable  $x$ 
  - From information theory, closely related to *relative entropy*, *information divergence*, and *information for discrimination*
- $D_{KL}(p(x) || q(x))$ : divergence of  $q(x)$  from  $p(x)$ , measuring the information lost when  $q(x)$  is used to approximate  $p(x)$



Original Gaussian PDF's



KL Area to be Integrated



$$D_{KL}(p(x) || q(x)) = \sum_{x \in X} p(x) \ln \frac{p(x)}{q(x)}$$

Discrete form



$$D_{KL}(p(x) || q(x)) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx$$

Continuous form

Ack.: Wikipedia entry: *The Kullback-Leibler (KL) divergence*



# ClassPhrase II: Feature Extraction: Informativeness

- Deriving 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 dash should be higher (punctuations information)
    - “state-of-the-art”
- We can also incorporate features using some NLP techniques, such as POS tagging, chunking, and semantic parsing

# ClassPhrase III: Classifier

- Weakly Supervised
  - Labels: Whether a phrase is a quality one or not
    - “support vector machine”: 1
    - “the experiment shows”: 0
  - For ~1GB corpus, only 300 labels
- Pros
  - Binary annotations are easy
- Cons
  - The selection of hundreds of varying-quality phrases from millions of candidates should be careful
- Random Forest as classifier
  - Predicted phrase quality scores lie in  $[0, 1]$
  - Bootstrap many different datasets from limited labels

# SegPhrase

- 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

- Rectified phrase frequency (expected influence)
  - Example:

sequence	frequency	phrase?	rectified
support vector machine	100	yes	80
support vector	160	yes	50
vector machine	150	no	6
support	500	N/A	150
vector	1000	N/A	200
machine	1000	N/A	150

# SegPhrase: Segmentation of Phrases

- Partition a sequence of words by maximizing the likelihood
  - Considering
    - Phrase quality score
      - ClassPhrase assigns a **quality score** for each phrase
    - Probability in corpus
    - Length penalty
      - **length penalty**  $\ln$  , it favors shorter phrases
- Filter out phrases with low rectified frequency
  - Bad phrases are expected to rarely occur in the segmentation results

# SegPhrase+: Enhancing Phrasal Segmentation

- SegPhrase+: One more round for enhanced phrasal segmentation
- **Feedback**
  - Using rectified frequency, re-compute those features previously computing based on raw frequency
- Process
  - Classification ☾ Phrasal segmentation // SegPhrase  
→ Classification ☾ Phrasal segmentation // SegPhrase+
- **Effects** on computing quality scores
  - np hard in the strong sense
  - ~~np hard in the strong~~
  - data base management system



# Phrase Mining — Families of Methods

Supervised (linguistic analyzers)

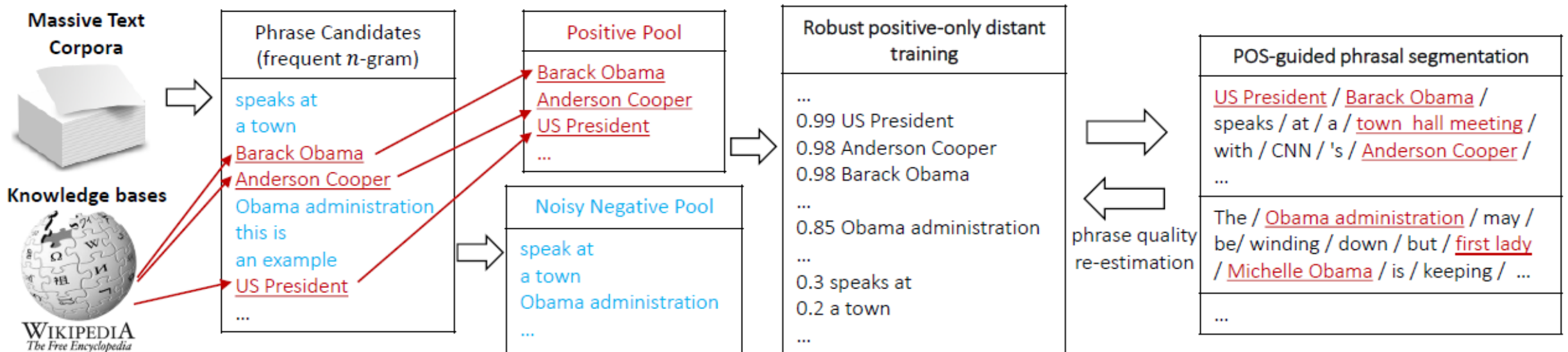
Unsupervised (Exploring statistical signals)

Weakly Supervised (Human provides a small set of labeled data)

Distantly Supervised (Exploring Knowledge-Bases, e.g., Wikipedia)

# AutoPhrase: Automated Phrase Mining

- Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, Jiawei Han, “**AutoPhrase**: Automated Phrase Mining from Massive Text Corpora”, 2017
- Automatic extraction of high-quality phrases (e.g., scientific terms and general entity names) in a given corpus (e.g., research papers and news)
  - No human efforts
  - Multiple languages
  - High performance—precision, recall, efficiency



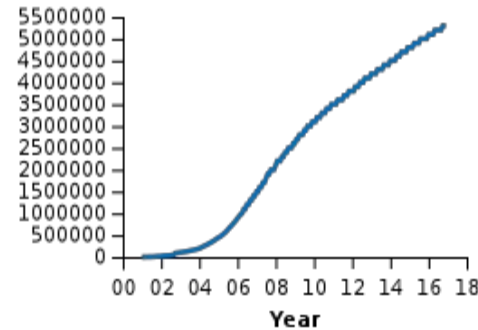
# Definition: “Automatic”

- Automatic 🐦 Minimal Human Effort
- Using only existing general knowledge bases without any other human effort

Entity names, High-quality Phrases



Number of Wikipedia articles



Rapidly growing!  
Freely available!

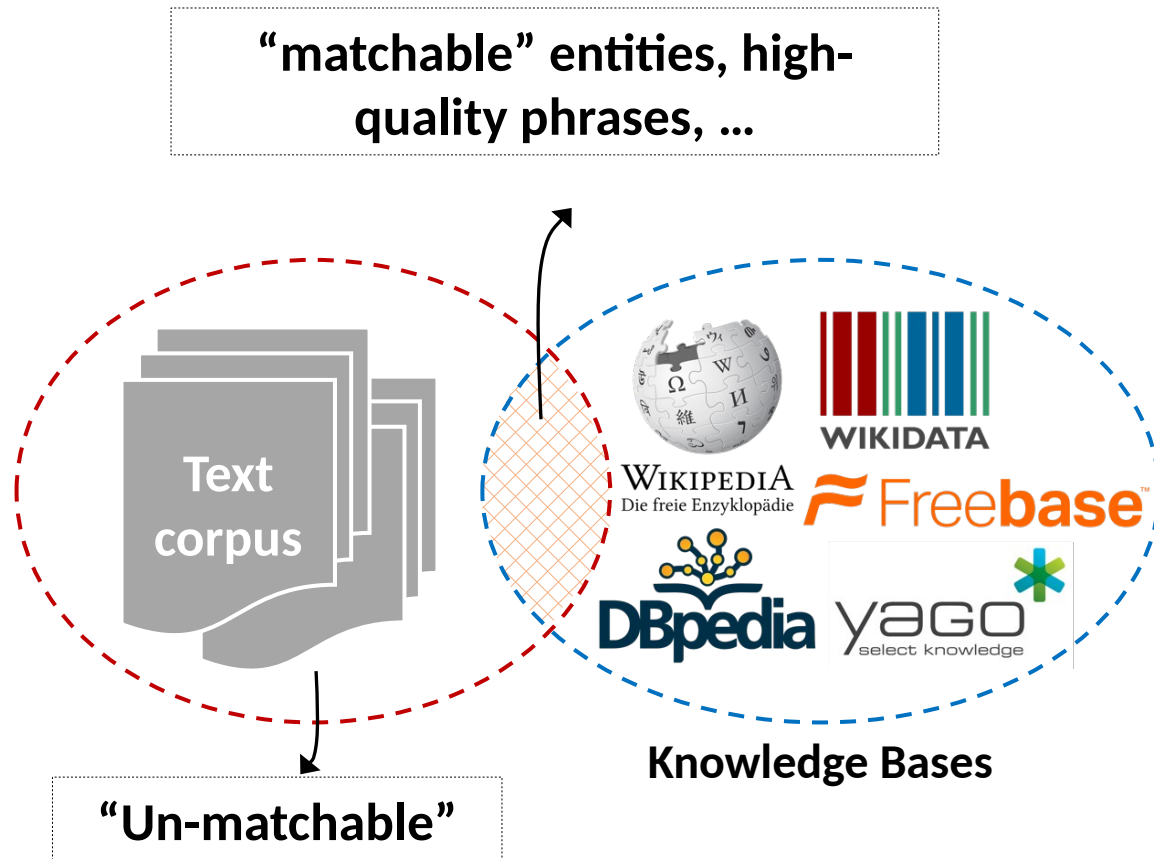
- Common knowledge
- Life sciences
- Art ...

That's it?  
Problem solved?  
Everything can be  
found in KBs?

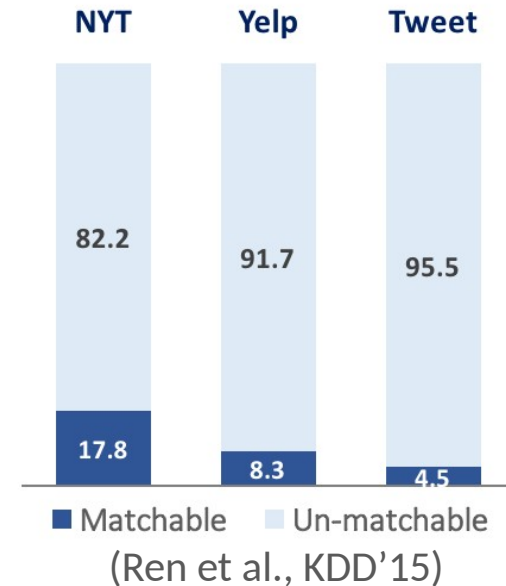




# Challenge: “Un-matchable” is Dominating!



Example: entity types



## Reasons

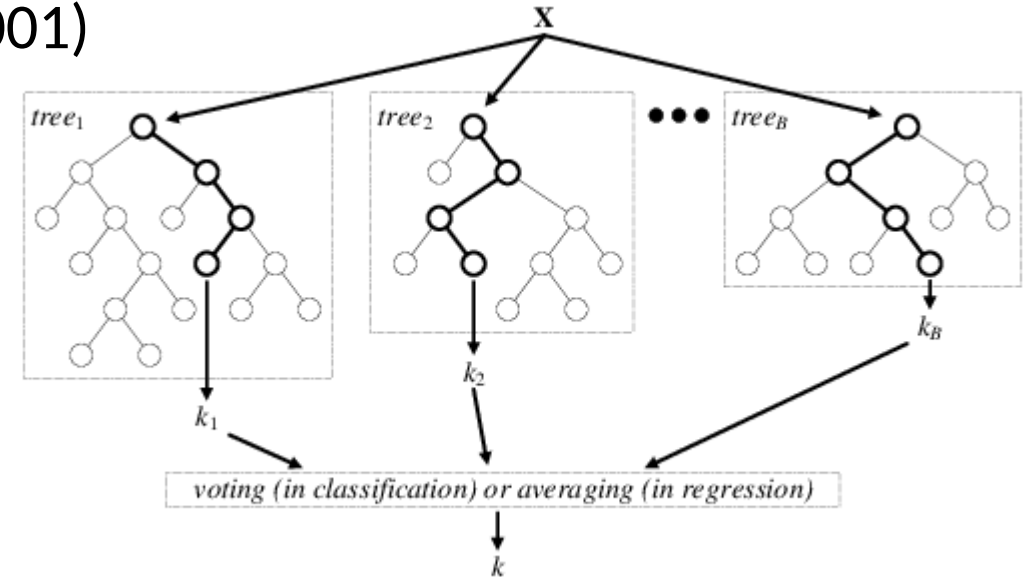
- Incomplete knowledge bases
- Low-confidence matching

# AutoPhrase: Label Generation by Distant Supervision

- Completely remove the human effort for labeling phrases
- **Distant training:** Utilize high-quality phrases in KBs (e.g., Wiki) as positive phrase labels
- **Method: Sampling-based label generation**
  - **Positive Labels**
    - Wikipedia contains many high-quality phrases in titles, keywords, and internal links
      - E.g., in Chinese, more than 20,000
    - Uniformly draw 100 samples as positive labels for single-word and multi-word phrases respectively
  - **Negative Labels**
    - Phrase candidates meeting the popularity requirement is huge in size and the majority of them are actually poor in quality (e.g., “francisco opera and”).
      - Ex. A small corpus in Chinese has about **4 million frequent phrase candidates**, while **more than 90%** are not in good quality

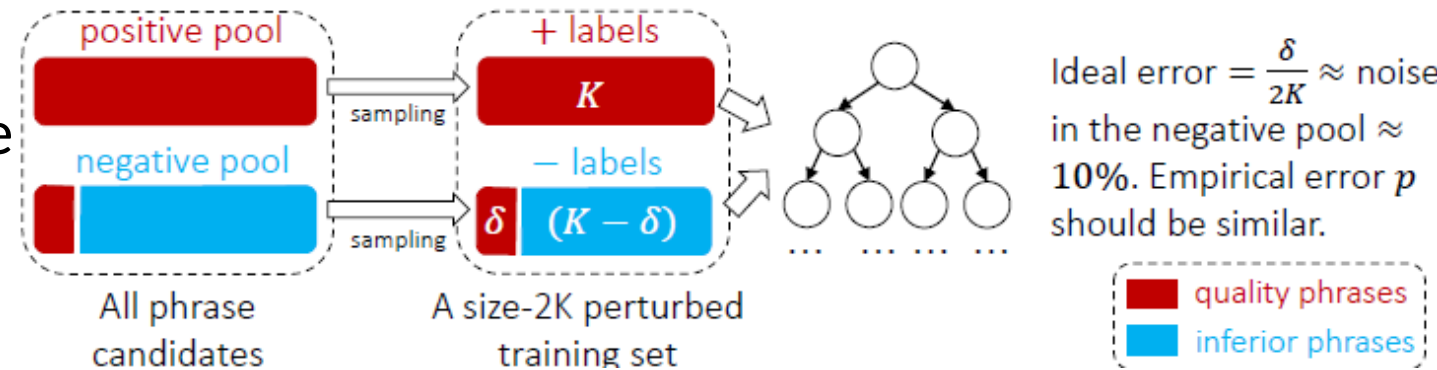
# Random Forest: Basic Concept

- Random Forest (first proposed by L. Breiman in 2001)
  - A variation of bagging for *decision trees*
  - *Data bagging*
    - Use a subset of training data by sampling with replacement for each tree
  - *Feature bagging*
    - At each node use a random selection of attributes as candidates and split by the best attribute among them
- Comparing with original bagging, the *random forests* method increases the diversity among generated trees
- During classification, each tree votes and the most popular class is returned



# 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:  $\delta$  quality phrases among the  $K$  negative labels



- Perturbed training set: size-2K subset of the full set of all phrase where the labels of some quality phrases are switched from positive to negative
- For each base classifier, we randomly draw  $K$  phrase candidates with replacement from the positive pool and the negative pool respectively
- We 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  $T$  independently trained base classifiers

# Generating High-Quality Phrases in Multi-Languages

- Complicated pre-processing models, such as dependency parsing, heavily rely on human efforts and thus cannot be smoothly applied to multiple languages
- Minimum Language Dependency = **Tokenization + POS tagging**
- Drawbacks of Frequency-based signals only: Over-decomposition & Under-decomposition
  - “Sophia Smith” vs. “Sophia” and “Smith”
  - “Great Firewall” vs. “firewall software”
- Drawbacks of POS only:
  - “classifier SVM” vs. “discriminative classifier” and “SVM”
- Context-aware phrasal segmentation

#1:	[Sophia	Smith]	was	born	in	England	.
	NNP	NNP	VBD	VBN	IN	NNP	.
#2:	...	the	[Great	Firewall]	is	...	.
	...	DT	NNP	NNP	VBZ	...	.
#3:	This	is	a	great	[firewall	software]	.
	DT	VBZ	DT	JJ	NN	NN	.

#4:	The	[discriminative	classifier]	[SVM]	is	...
	DT	JJ	NN	NN	VBZ	...

# Single-Word Modeling: Enhancing Recall

- AutoPhrase: Simultaneously model single-word and multi-word phrases
- A phrase is not only a group of multiple words, but also possibly a single word, as long as it functions as a constituent in the syntax of a sentence, e.g., “ISU”, “Illinois”
  - Based on our experiments: 10%~30% quality phrases are single-word phrases
- Modeling single-word phrases: Examining requirements of quality multi-word phrase
  - Popularity: sufficient frequent in the given corpus
  - Concordance: the collocation of tokens in such frequency that is significantly higher than random
  - Informativeness: indicative of a specific topic or concept
  - Completeness: Complete semantic unit
- Only **concordance** cannot be defined in single-word phrases
- **Independence**: A quality single-word phrase is more likely a complete semantic unit in the given documents

# References

- Blei, D.M. and Lafferty, J.D., 2009. Topic models. Text mining: classification, clustering, and applications, 10(71), p.34
- D. M. Blei and J. D. Lafferty. Visualizing Topics with Multi-Word Expressions, arXiv:0907.1013, 2009
- Church, K., Gale, W., Hanks, P. and Hindle, D., 1991. Using statistics in lexical analysis. Lexical acquisition: exploiting on-line resources to build a lexicon, 115, p.164.
- 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
- Deane, P., 2005, A nonparametric method for extraction of candidate phrasal terms. In Proc. ACL, pp. 605-613
- Ahmed El-Kishky, Yanglei Song, Chi Wang, Clare R. Voss, and Jiawei Han, "[Scalable Topical Phrase Mining from Text Corpora](#)", PVLDB 8(3): 305 - 316, 2015
- Evans, D.A. and Zhai, C., 1996. Noun-phrase analysis in unrestricted text for information retrieval. In Proc. ACL, pp. 17-24
- K. Frantzi, S. Ananiadou, and H. Mima, Automatic Recognition of Multi-Word Terms: the c-value/nc-value Method. Int. Journal on Digital Libraries, 3(2), 2000
- Koo, T., Carreras Pérez, X. and Collins, M., 2008. Simple semi-supervised dependency parsing. In Proc. ACL, pp. 595-603
- R. V. Lindsey, W. P. Headden, III, M. J. Stipicevic. A phrase-discovering topic model using hierarchical pitman-yor processes, EMNLP-CoNLL'12.
- Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, Jiawei Han, "[Mining Quality Phrases from Massive Text Corpora](#)", in Proc. of 2015 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'15), Melbourne, Australia, May 2015
- Jialu Liu, Jingbo Shang, and Jiawei Han, [Phrase Mining from Massive Text and Its Applications](#), Morgan & Claypool Publishers, 2017.

# References (Continued)

- Liu, Z., Chen, X., Zheng, Y. and Sun, M., 2011, June. Automatic keyphrase extraction by bridging vocabulary gap. In Proceedings of the Fifteenth Conference on Computational Natural Language Learning (pp. 135-144)
- O. Medelyan and I. H. Witten, Thesaurus Based Automatic Keyphrase Indexing. IJCDL'06
- Q. Mei, X. Shen, C. Zhai. Automatic Labeling of Multinomial Topic Models, KDD'07
- Mihalcea, R. and Tarau, P., 2004. TextRank: Bringing order into texts. Association for Computational Linguistics.
- A. Parameswaran, H. Garcia-Molina, and A. Rajaraman. Towards the Web of Concepts: Extracting Concepts from Large Datasets. VLDB'2010
- Park, Y., Byrd, R.J. and Boguraev, B.K., 2002. Automatic glossary extraction: beyond terminology identification. In Proceedings of the 19th international conference on Computational linguistics-Volume 1 (pp. 1-7)
- Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, Jiawei Han, "[Automated Phrase Mining from Massive Text Corpora](#)", IEEE Transactions on Knowledge and Data Engineering, 30(10):[1825-1837](#) (2018)
- Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C. and Nevill-Manning, C.G., 1999. KEA: Practical automatic keyphrase extraction. In Proceedings of the fourth ACM conference on Digital libraries (pp. 254-255). ACM.
- Xun, E., Huang, C. and Zhou, M., 2000. A unified statistical model for the identification of English baseNP. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (pp. 109-116).
- X. Wang, A. McCallum, X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval, ICDM'07
- Zhang, Z., Iria, J., Brewster, C. and Ciravegna, F., 2008. A comparative evaluation of term recognition algorithms. In LREC.