# Learning to Extract Descriptive Keyphrases from Scholarly Documents

Cornelia Caragea

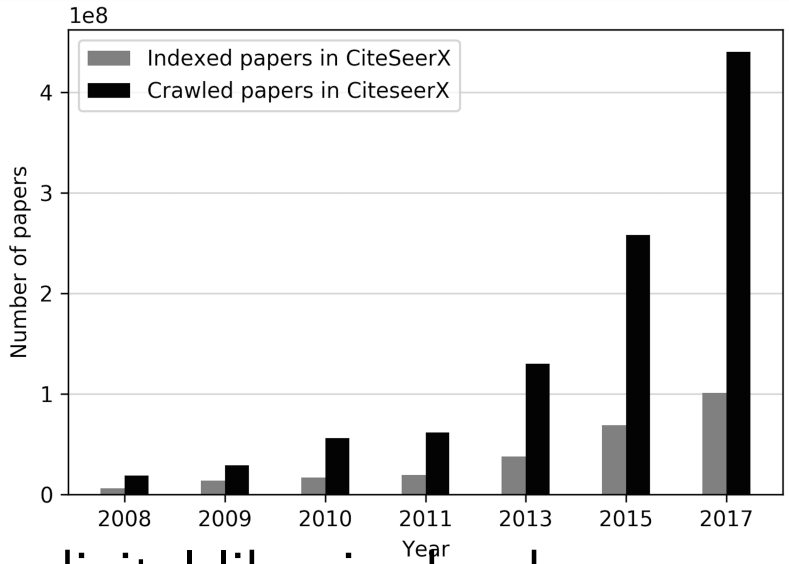
Computer Science
University of Illinois at Chicago

# Scholarly Big Data

## Large number of scholarly documents on the Web

- Microsoft Academic expanded from 83 million records in 2015 to 168 million in 2017 [Hug and Brandle, 2017].
- Google Scholar was estimated to have $\approx 160$ million documents in 2014 [Orduna-Malea et al, 2015].

The growth in the number of papers crawled and indexed by CiteSeerX:




- Navigating in these digital libraries has become very challenging.

# Keyphrases

- **Keyphrases** provide a high-level topic description of a document and can allow for *efficient* data organization, information processing, and document understanding.

**Example: A WWW paper - the author-input keyphrases are shown in red**

*Factorizing Personalized Markov Chains for Next-Basket Recommendation*

*by Rendle, Freudenthaler, and Schmidt-Thieme*

"Recommender systems are an important component of many websites. Two of the most popular approaches are based on matrix factorization (MF) and Markov chains (MC). MF methods learn the general taste of a user by factorizing the matrix over observed user-item preferences. *[...]* In this paper, we present a method bringing both approaches together. Our method is based on personalized transition graphs over underlying Markov chains. *[...]* We show that our factorized personalized MC (FPMC) model subsumes both a common Markov chain and the normal matrix factorization model. For learning the model parameters, we introduce an adaption of the Bayesian Personalized Ranking (BPR) framework for sequential basket data. *[...]*"

# Automatic Keyphrase Extraction

- Keyphrases associated with research papers:
  - Useful in applications such as:
    - topic tracking, document clustering, classification, and summarization, information filtering and search, and query formulation.
- However, manually annotated keyphrases are not always provided with the documents:
  - E.g., documents available from the ACL Anthology and the AAAI DL
- Hence, accurate approaches are required for **keyphrase extraction** from research documents
  - **Keyphrase extraction** is defined as the problem of automatically extracting descriptive phrases or concepts from documents

# Previous Approaches to Keyphrase Extraction

- Many approaches have been studied:
  - Supervised [Frank et al., 1999; Turney, 2000; Hulth, 2003]
    - Binary classification.
  - Unsupervised [Mihalcea and Tarau, 2004; Wan and Xiao, 2008; Liu et al., 2010; Lahiri, Choudhury, and Caragea, 2014]
    - Ranking problem.

- Generally, previous approaches:
  - Use only the textual content of the target document [Mihalcea and Tarau, 2004; Liu et al., 2010].
  - Incorporate a local neighborhood of a document for extracting keyphrases [Wan and Xiao, 2008]
    - However, the neighborhood is limited to textually-similar documents.

# Our Questions

- In addition to a document's textual content and textually-similar neighbors, are there other informative neighborhoods in research document collections?
- Can these neighborhoods improve keyphrase extraction?

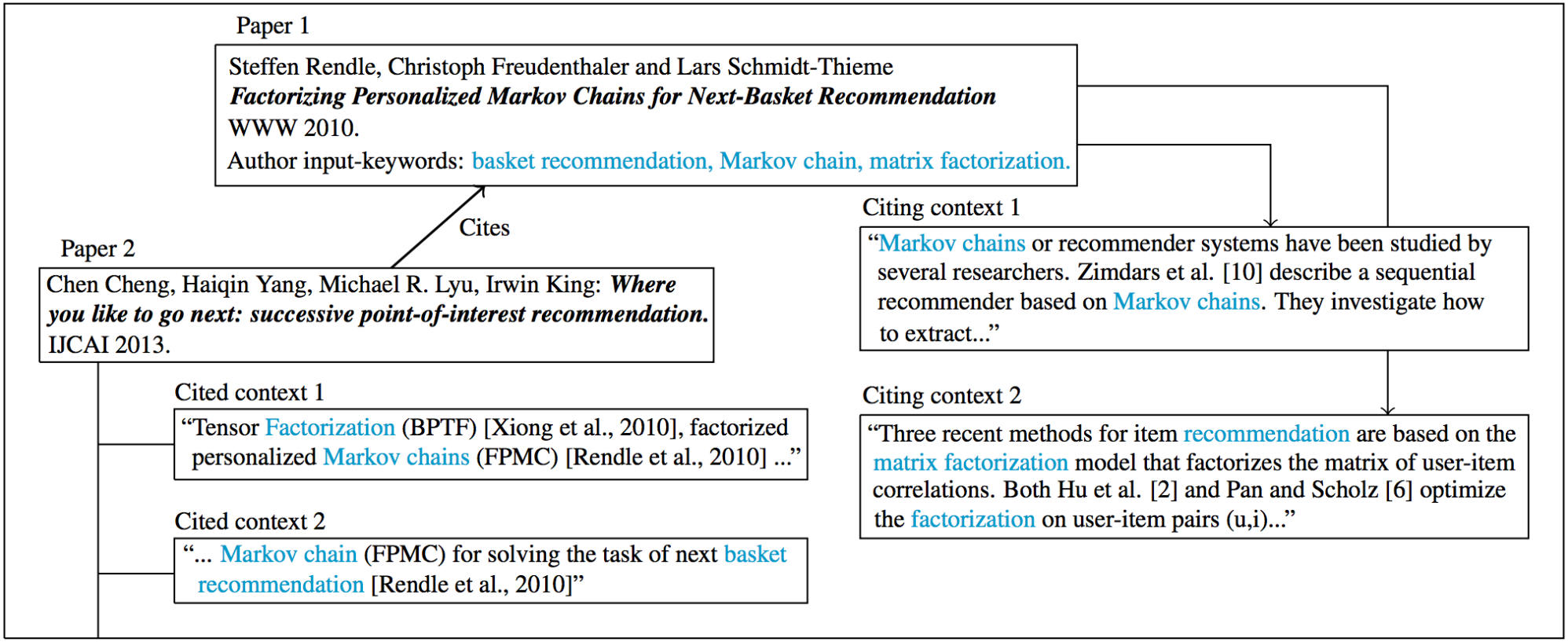
# From Data to Knowledge

- A typical scientific research paper:
  - Proposes new problems or extends the state-of-the-art for existing research problems.
  - Cites relevant, previously-published papers in appropriate *contexts.*
- The citations between research papers give rise to an interlinked document network, commonly referred to as the citation network.

# Citation Networks

- In a citation network, information flows from one paper to another via the citation relation [Shi, Leskovec, and McFarland, 2010]
- Citation contexts capture the influence of one paper on another as well as the flow of information
- Citation contexts or the short text segments surrounding a paper's mention serve as "micro summaries" of a cited paper!

# A Small Citation Network




- Citation contexts are very informative!

[Gollapalli and Caragea, 2014 (**AAAI**); Caragea, 2016 (**AI4DataSci**)]

# Citation Contexts - Previous Usage

- Using terms from citation contexts resembles the analysis of hyperlinks and the graph structure of the Web
  - Web search engines build on the intuition that the anchor text pointing to a page is a good descriptor of its content, and thus use anchor terms as additional index terms for a target webpage.

- Previously used for other tasks:
  - Indexing of cited papers [Ritchie, Teufel, and Robertson, 2006]
  - Author influence in document networks [Kataria, Mitra, Caragea, and Giles, 2011]
  - Scientific paper summarization [Abu-Jbara and Radev, 2011; Qazvinian, Radev, and Özgür, 2010; Qazvinian and Radev, 2008; Mei and Zhai, 2008; Lehnert et al., 1990; Nakov et al., 2004]

# Citation Contexts to Keyphrase Extraction

- How can we use these contexts and how do they help in keyphrase extraction?
- We proposed:
  - **CiteTextRank**: an unsupervised, graph-based algorithm that incorporates evidence from multiple sources (citation contexts as well as document content) in a flexible way to extract keyphrases [Gollapalli and Caragea, 2014 (**AAAI**); Caragea, 2016 (**AI4DataSci**)].

# Unsupervised Keyphrase Extraction I

General steps for unsupervised keyphrase extraction algorithms:

1. Extract candidate words from the content of the target document by applying stopword and parts-of-speech filters.

> **Unsupervised Semantic Parsing**
>
> We present the first unsupervised approach to the problem of learning a semantic parser, using Markov logic . Our USP system transforms dependency trees into quasi-logical forms, recursively induces lambda forms from these, and clusters them to abstract away syntactic variations of the same meaning. The MAP semantic parse of a sentence is obtained by recursively assigning its parts to lambda-form clusters and composing them. We evaluate our approach by using it to extract a knowledge base from biomedical abstracts and answer questions. USP substantially outperforms TextRunner, DIRT and an informed baseline on both precision and recall on this task.

2. Score candidate words based on some criterion.
    - For example, in the TFIDF scoring scheme, a candidate word score is the product of its frequency in the document and its inverse document frequency in the collection.
    - MAP: 0.01; semantic: 0.3; parse: 0.05

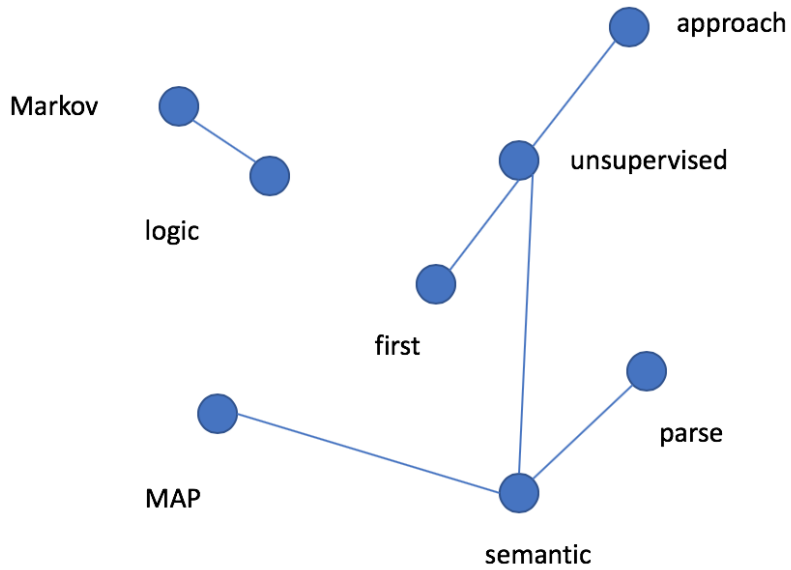
# Unsupervised Keyphrase Extraction II

3. Score consecutive words, phrases or $n$-grams using the sum of scores of individual words that comprise the phrase [Wan and Xiao, 2008].
   - MAP semantic parse: 0.36.

4. Output the top-scoring phrases as the predicted keyphrases.

# TextRank for Keyphrase Extraction

- An application of PageRank to word graphs for an NLP application (window size = 2)

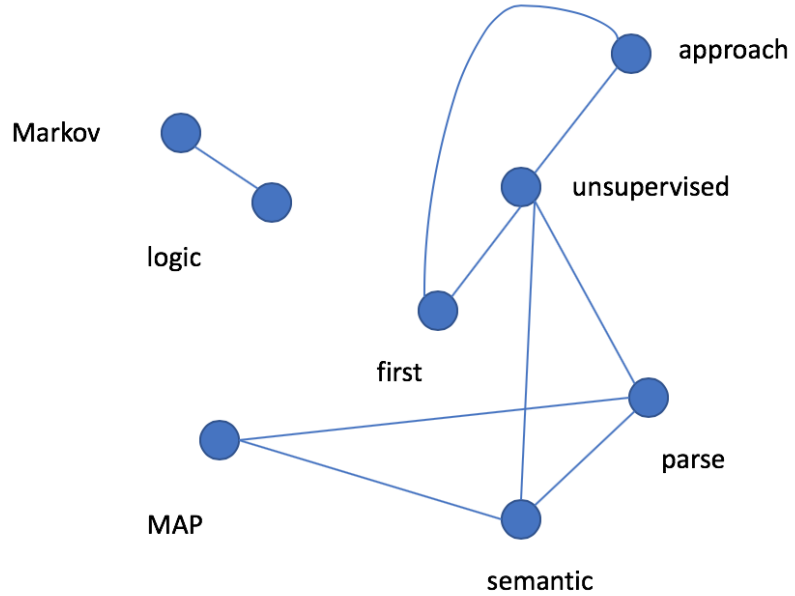



$$s(v_i) = \alpha \sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{\sum_{v_k \in Adj(v_j)} w_{jk}} s(v_j) + (1-\alpha) p_i,$$

where $\alpha$ is a damping factor ($\alpha = 0.85$) and $\mathbf{p} = [\frac{1}{n}, \cdots, \frac{1}{n}]$.
[Mihalcea and Tarau, 2004]

# SingleRank for Keyphrase Extraction

- An application of PageRank to word graphs for an NLP application (window size >= 2)




$$s(v_i) = \alpha \sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{\sum_{v_k \in Adj(v_j)} w_{jk}} s(v_j) + (1-\alpha) p_i,$$

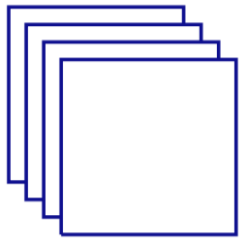
where $\alpha$ is a damping factor ($\alpha = 0.85$) and $\mathbf{p} = [\frac{1}{n}, \cdots, \frac{1}{n}]$.
[Wan and Xiao, 2008]

# ExpandRank for Keyphrase Extraction

- Extension of SingleRank to include textually similar documents.

Target document $d$:


$Sim$: textually similar neighbors:



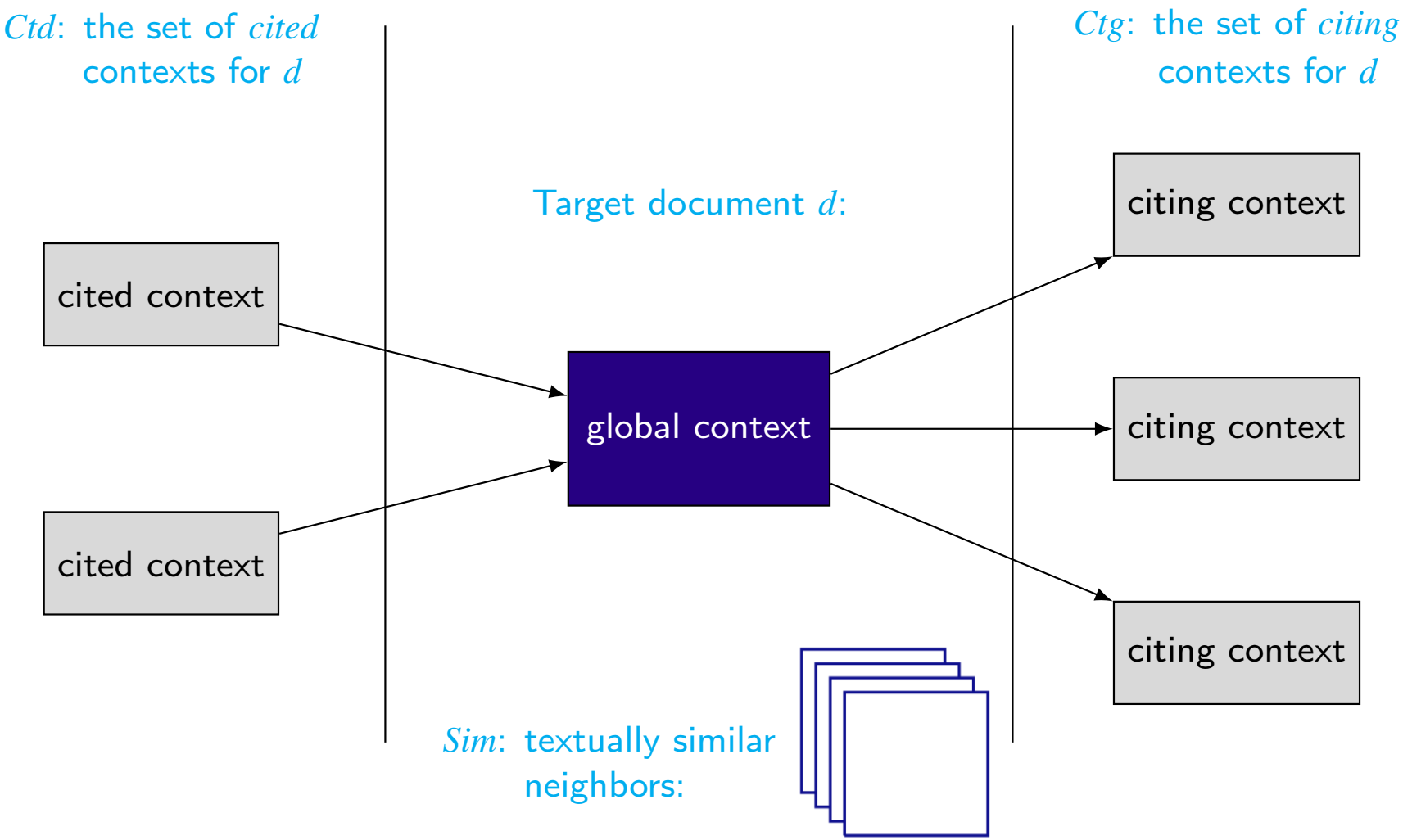
$$s(v_i) = \alpha \sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{\sum_{v_k \in Adj(v_j)} w_{jk}} s(v_j) + (1-\alpha) p_i,$$

[Wan and Xiao, 2008]

# CiteTextRank

**CiteTextRank** incorporates information from *citation contexts* while scoring candidate words in step 2, through an extension of PageRank.

# CiteTextRank: Definitions and Notation




- $T = \{Ctd, Ctg, Sim, g\}$ represents the types of available contexts for $d$.

# Graph Construction in CiteTextRank I

We construct an undirected graph, $G = (V, E)$ for $d$ as follows:

1. For each unique candidate word from all available contexts of $d$, add a vertex in $G$.
2. Add an undirected edge between two vertices $v_i$ and $v_j$ if the words corresponding to these vertices occur within a window of $w$ contiguous tokens in any of the contexts.

**Unsupervised Semantic Parsing**

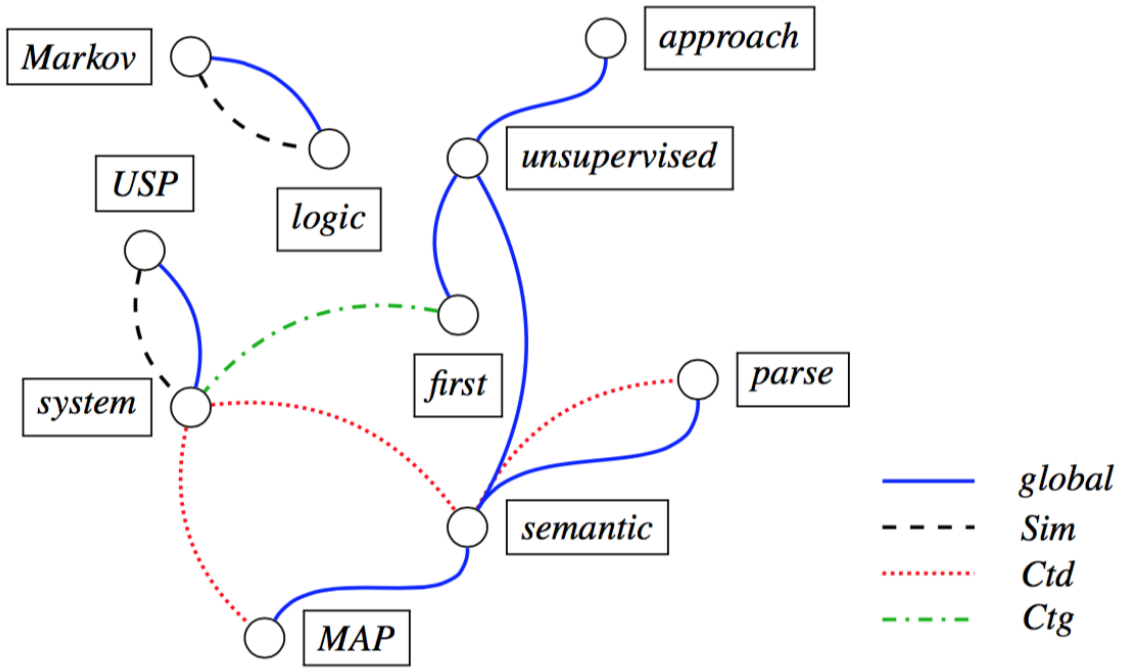
We present the first unsupervised approach to the problem of learning a semantic parser, using Markov logic . Our USP system transforms dependency trees into quasi-logical forms, recursively induces lambda forms from these, and clusters them to abstract away syntactic variations of the same meaning. The MAP semantic parse of a sentence is obtained by recursively assigning its parts to lambda-form clusters and composing them. We evaluate our approach by using it to extract a knowledge base from biomedical abstracts and answer questions. USP substantially outperforms TextRunner, DIRT and an informed baseline on both precision and recall on this task.

# Graph Construction in CiteTextRank II

**Unsupervised Semantic Parsing**

We present the first unsupervised approach to the problem of learning a semantic parser, using Markov logic . Our USP system transforms dependency trees into quasi-logical forms, recursively induces lambda forms from these, and clusters them to abstract away syntactic variations of the same meaning. The MAP semantic parse of a sentence is obtained by recursively assigning its parts to lambda-form clusters and composing them. We evaluate our approach by using it to extract a knowledge base from biomedical abstracts and answer questions. USP substantially outperforms TextRunner, DIRT and an informed baseline on both precision and recall on this task.

$w = 2$:



# Parameterized Edge Weights in CiteTextRank

- The weight $w_{ij}$ of an edge $(v_i, v_j) \in E$ is given as

$$w_{ij} = w_{ji} = \sum_{t \in T} \sum_{c \in C_t} \lambda_t \cdot \text{cossim}(c, d) \cdot \#_c(v_i, v_j)$$

  where $\lambda_t$ is the weight for contexts of type $t$ and $C_t$ is the set of contexts of type $t \in T$.
- Unlike simple graph edges with fixed weights, our equation corresponds to *parameterized edge weights*.
- We incorporate the notion of "importance" of contexts of a certain type using the $\lambda_t$ parameters.

# Vertex Scoring in CiteTextRank

- Initialization: $\mathbf{s} = [s(v_1), \cdots, s(v_n)] = [\frac{1}{n}, \cdots, \frac{1}{n}]$, where $n = |V|$.
- We score vertices in $G$ using their PageRank obtained by recursively computing the equation:

$$s(v_i) = \alpha \sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{\sum_{v_k \in Adj(v_j)} w_{jk}} s(v_j) + (1 - \alpha) p_i,$$

  where $\alpha$ is a damping factor ($\alpha = 0.85$) and $\mathbf{p} = [\frac{1}{n}, \cdots, \frac{1}{n}]$.
  [Page et al., 1999; Haveliwala et al., 2003]

- The PageRank score for a vertex provides a measure of its importance in the graph by taking into account global information computed recursively from the entire graph.

# Experiments and Results for CiteTextRank

Datasets:

- We constructed several datasets of research papers and their citation networks using the CiteSeerX digital library.
  - These datasets consist of the proceedings of the ACM KDD and WWW conferences.
- The author-input keyworks were used as gold-standard for evaluation.

| Conference | #Docs(CiteSeerX) | #DocsUsed | AvgKeywords | AvgCtg | AvgCtd |
|---|---|---|---|---|---|
| WWW | 1350 | 406 | 4.81 | 15.91 | 17.39 |
| KDD | 834 | 335 | 4.09 | 18.85 | 16.82 |

Table: Summary of datasets: #DocsUsed represent the number of documents for which both citing and cited contexts were extracted from CiteSeerX and for which author-input keyphrases were available.

All datasets and code are available online.

# Experimental Setting for CiteTextRank

Our experiments are organized around the following questions:

- Does citation network information aid keyphrase extraction from research papers?
- How does CiteTextRank perform in the absence of either citing and cited contexts?
- How does CiteTextRank compare with previous methods?

Evaluation measures:

- Mean reciprocal rank, MRR

$$MRR = \frac{1}{|D|} \sum_{d=1}^{|D|} \frac{1}{r_d}$$

$r_d$ is the rank at which the first correct prediction was found for $d \in D$.

# Citation Network Information in Keyphrase Extraction

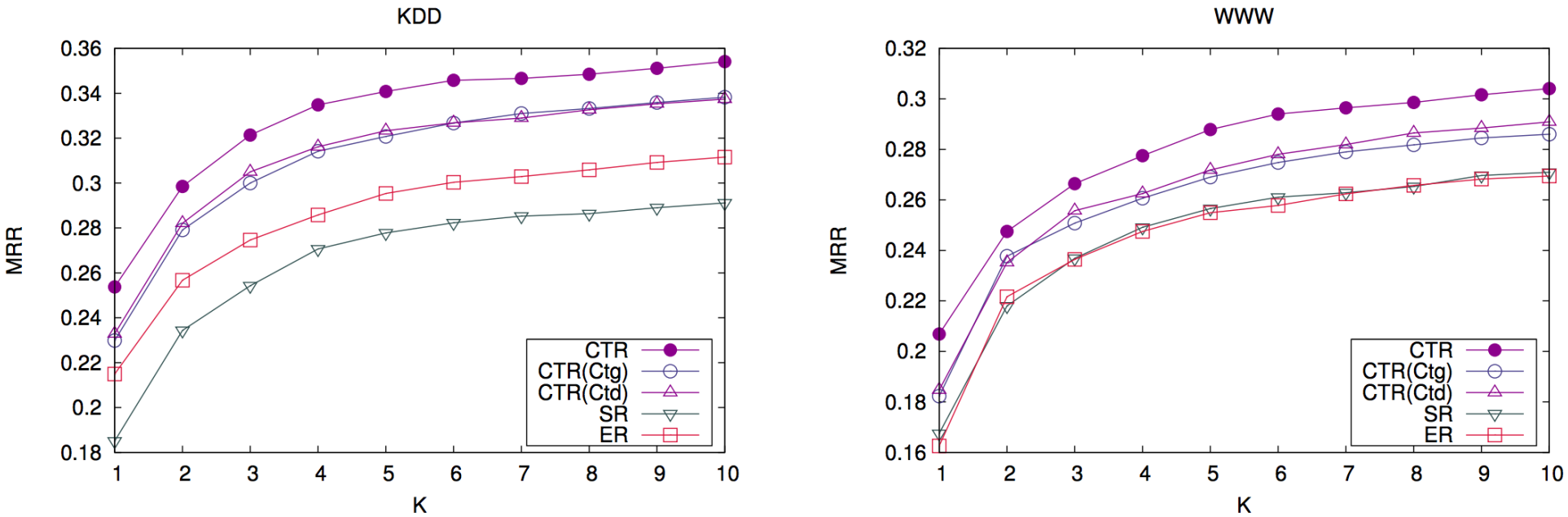



Figure: CiteTextRank (CTR) that uses citation network neighbors is compared with ExpandRank (ER) that uses textually-similar neighbors and SingleRank (SR) that only uses the target document content [Wan and Xiao, 2008].

**CiteTextRank** substantially outperforms models that take into account only textually-similar documents. Cited and citing contexts contain significant hints that aid keyphrase extraction.

# CiteTextRank vs. Previous Methods

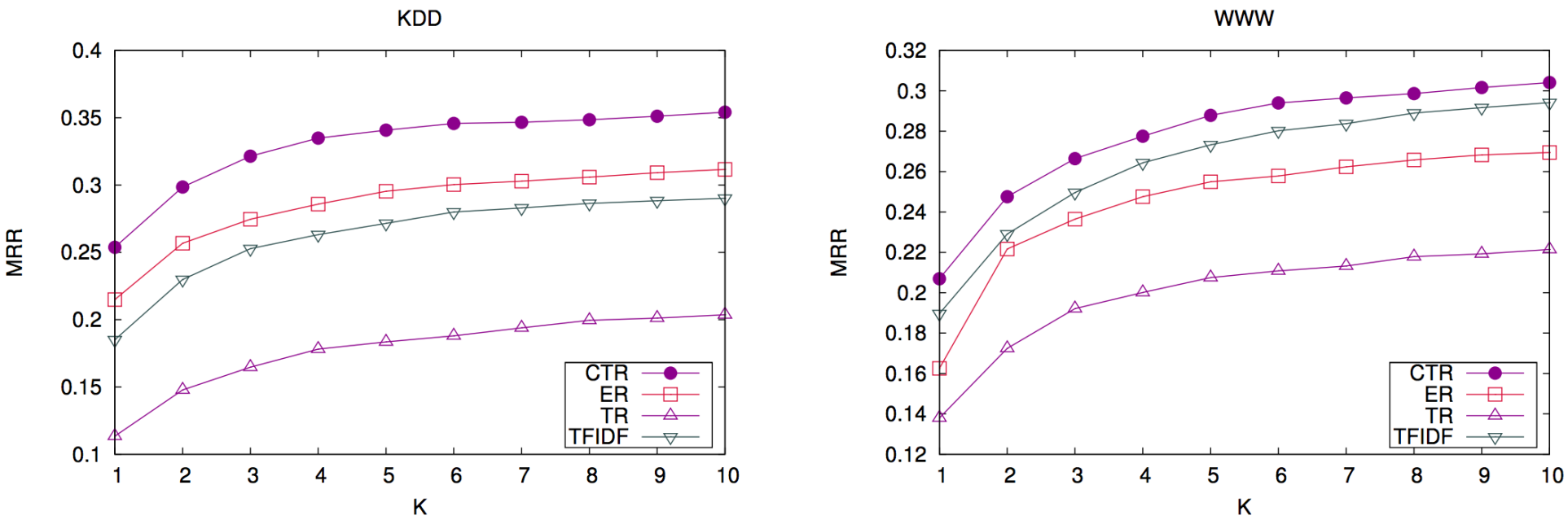



Figure: MRR curves for different keyphrase extraction methods. CTR is compared with the baselines: TFIDF, TextRank (TR) [Mihalcea and Tarau, 2004], and ExpandRank (ER) [Wan and Xiao, 2008].

**CiteTextRank** outperforms previous models for keyphrase extraction.

# Summary, Limitations and Potential Extensions

- Developments in keyphrase extraction are central to ***knowledge discovery and organization*** and have a direct impact on the development of digital libraries.

- Citation context lengths: Incorporate more sophisticated approaches to identifying the text that is relevant to a target citation [Abu-Jbara and Radev, 2012; Teufel et al., 2006] and study the influence of context lengths on the quality of extracted keyphrase.
- Keyphrase extraction is very subjective
- Extend our models to other CS areas and other scientific domains, e.g., ACL Anthology, PubMed, Social Science, Political Science, Ecology.

# References


- S. Das Gollapalli and C. Caragea (2014). Extracting Keyphrases from Research Papers using Citation Networks. In: *Proceedings of the 28th American Association for Artificial Intelligence (**AAAI '14**).*
- C. Caragea (2016). Identifying Descriptive Keyphrases from Scholarly Big Data. In: *Artificial Intelligence for Data Science (**AI4DataSci '16**).*
- L. Sterckx, C. Caragea, T. Demeester, and C. Develder. (2016). Supervised Keyphrase Extraction as Positive Unlabeled Learning. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (**EMNLP '16**).*
- R. Mihalcea and P. Tarau. (2004). TextRank: Bringing order into text. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (**EMNLP '04**).*
- X. Wan and J. Xiao (2008). Single document keyphrase extraction using neighborhood knowledge. In: *Proceedings of the Association for the Advancement of Artificial Intelligence (**AAAI '08**).*