# Removing the Clutter of Uninteresting Words in Text

## RAAI to the Rescue?

Cornelia Caragea

Computer Science
Kansas State University

RAAI - June 25, 2018

Machine Learning Group
KSU Computer Science

# The Essence of Text

The narrative of a text contains many details, which are often not interesting or important, and often hide the essence of the text.

As Roman power moved away, tribes from outside the old borders moved in to fill the vacuum. Visigoths set up a new kingdom in Iberia, while Vandals settled eventually in north Africa. In Britain, Germanic tribes arrived and settled on the east coast in 5CE. These settlers eventually formed small Anglo-Saxon kingdoms which filled the vacuum left by the departure of the Romans. Post-Roman Italy itself came under the sway of the Ostrogoths, whose most influential king, Theodoric, was hailed as a new emperor by the Roman Senate, and had good relations with the Christian pope, but kept his seat of power at Ravenna in northern Italy. Southern Italy and Sicily were under the sway of the Byzantine Emperor for several centuries during this period.

[Experiment done with help from Anca Morcovescu, K-12 teacher in DFW area].

# Keyphrases

As Roman power moved away, tribes from outside the old borders moved in to fill the vacuum. **Visigoths** set up a new kingdom in Iberia, while **Vandals** settled eventually in north Africa. In Britain, **Germanic tribes** arrived and settled on the east coast in 5CE. These settlers eventually formed small Anglo-Saxon kingdoms which filled the vacuum left by the departure of the Romans. Post-Roman Italy itself came under the sway of the Ostrogoths, whose most influential king, **Theodoric** was hailed as a new emperor by the Roman Senate, and had good relations with the Christian pope, but kept his seat of power at Ravenna in northern Italy. Southern Italy and Sicily were under the sway of the Byzantine Emperor for several centuries during this period.

- Keyphrases provide a high-level topic description of a document and can allow for *efficient* processing of more information in less time and have a high impact on document understanding.

# Keyphrase Extraction

- Keyphrases
  - Useful in applications such as
    - topic tracking, information filtering and search, query formulation, document clustering, classification, and summarization
- However, manually annotated keyphrases are not always provided with the documents:
  - Need to be gleaned from the content of documents
  - E.g., documents available from the ACL Anthology and the AAAI DL
- Hence, accurate approaches are required for **keyphrase extraction**
  - **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 [Hasan and Ng, 2014]:
  - Supervised approaches [Frank et al., 1999; Turney, 2000; Hulth, 2003; Caragea et al., 2014]
    - Binary classification: candidate phrases classified as keyphrases or non-keyphrases.
  - Unsupervised approaches [Mihalcea and Tarau, 2004; Wan and Xiao, 2008; Liu et al., 2010; Gollapalli and Caragea, 2014]
    - Ranking: candidate phrases are ranked using various measures such as tf, tf-idf, and PageRank scores.

# Candidate Words or Phrases

- Candidate words or phrases are extracted 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.

# Supervised Keyphrase Extraction - Methodology

- Generate Candidate Phrases:
  - We first apply parts-of-speech filters and retain only the nouns and adjectives.
  - Porter Stemmer is applied on every word.
  - Words that have contiguous positions in the document are concatenated into *n*-grams.
  - Finally, we eliminate phrases that end with an adjective and the unigrams that are adjectives.
- Represent each candidate phrase as a **vector of features.**
- Assign a positive or negative class to each phrase based on the human annotated labels.
- Use the data to train machine learning classifiers, which are then used to predict keyphrases for future documents.

# Features for Supervised Keyphrase Extraction

| Feature Name | Description |
|---|---|
| Existing features for keyphrase extraction | |
| *tf-idf* | term frequency * inverse document frequency, computed from a target paper; used in KEA |
| *relativePos* | the position of first occurrence of a phrase divided by the total number of tokens; used in KEA and Hulth's methods |
| POS | the part-of-speech tag of the phrase; used in Hulth's methods |
| Novel features - Citation Network Based | |
| *inCited* | if the phrase occurs in cited contexts |
| *inCiting* | if the phrase occurs in citing contexts |
| *citation tf-idf* | the *tf-idf* value of the phrase, computed from the aggregated citation contexts |
| Novel features - Extensions of Existing Features | |
| *first position* | the distance of the first occurrence of a phrase from the beginning of a paper |
| *tf-idf-Over* | *tf-idf* larger than a threshold $\theta$ |
| *firstPosUnder* | the distance of the first occurrence of a phrase from the beginning of a paper is below some value $\beta$ |

[Caragea et al., 2014; Bulgarov and Caragea, 2015]

# Supervised vs. Unsupervised Models

- Generally, supervised approaches are more accurate.

| | WWW | | | KDD | | |
|---|---|---|---|---|---|---|
| Method | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Supervised | | | | | | |
| **Citation - Enhanced (CeKE)** | **0.227** | **0.386** | **0.284** | **0.213** | **0.413** | **0.280** |
| Hulth - $n$-gram with tags | 0.165 | 0.107 | 0.129 | 0.206 | 0.151 | 0.172 |
| KEA | 0.210 | 0.146 | 0.168 | 0.178 | 0.124 | 0.145 |
| Unsupervised - Top 5 predicted keyphrases | | | | | | |
| TF-IDF | 0.089 | 0.100 | 0.094 | 0.083 | 0.102 | 0.092 |
| TextRank | 0.058 | 0.071 | 0.062 | 0.051 | 0.065 | 0.056 |
| ExpandRank - 1 neigh. | 0.088 | 0.109 | 0.095 | 0.077 | 0.103 | 0.086 |
| ExpandRank - 5 neigh. | 0.093 | 0.113 | 0.100 | 0.080 | 0.108 | 0.090 |
| CiteTextRank | 0.110 | 0.134 | 0.119 | 0.133 | 0.153 | 0.141 |

- However, supervised models require large human-annotated corpora.
  - Led to significant attention towards unsupervised approaches.

# Most Informative Features for Keyphrase Extraction

- Interestingly, features used in supervised approaches influenced the progress of unsupervised approaches, e.g., TF-IDF based ranking.

| Rank | *Feature* | IG Score |
|---|---|---|
| 1 | *abstract tf-idf* | 0.0234 |
| 2 | *first position* | 0.0188 |
| 3 | *citation tf-idf* | 0.0177 |
| 4 | *relativePos* | 0.0154 |
| 5 | *firstPosUnder* | 0.0148 |
| 6 | *inCiting* | 0.0129 |
| 7 | *inCited* | 0.0098 |
| 8 | *POS* | 0.0085 |
| 9 | *tf-idf-Over* | 0.0078 |

Table: Feature ranking by Information Gain on WWW.

- Despite the effectiveness of the relative position in supervised approaches, this has not been used before in unsupervised methods.

# From Data to Knowledge

Intuitively, keyphrases occur frequently and occur very early in a document.

**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.

- We propose:
  - **PositionRank**: an unsupervised, graph-based algorithm that incorporates information from all positions of a word's occurrences into a biased-PageRank to rank keyphrases [Florescu and Caragea, 2017].

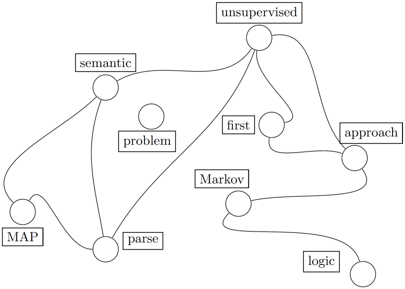
# PositionRank

- Graph construction at word level:

**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.

- window = 3
- $G = (V, E)$



# Biasing PageRank

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

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

where $\alpha$ is a damping factor ($\alpha = 0.85$) and $\widetilde{\mathbf{p}} = [\widetilde{p}_i]_{i=1,\ldots,n} = [\frac{1}{n}, \cdots, \frac{1}{n}]$.

- Position-Biased PageRank:
  - The idea is to assign higher probabilities to words that occur early in a document and occur frequently - assign different $\widetilde{p}_i$ probabilities.

# Example

Unsupervised Semantic[2] Parsing

We present the first unsupervised approach to the problem of learning a semantic[16] 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[51] 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[90], DIRT and an informed baseline on both precision and recall on this task.

- $p(\text{textrunner}) = \frac{1}{90} = 0.011$
- $p(\text{semantic}) = \frac{1}{2} + \frac{1}{16} + \frac{1}{51} = 0.582$

$\widetilde{p}$ is set to the normalized weights for each candidate word as follows:

$$\widetilde{p} = \left[\frac{p_1}{p_1+p_2+\ldots+p_{|V|}}, \frac{p_2}{p_1+p_2+\ldots+p_{|V|}}, \ldots, \frac{p_{|V|}}{p_1+p_2+\ldots+p_{|V|}}\right]$$

# Example

Unsupervised Semantic[2] Parsing

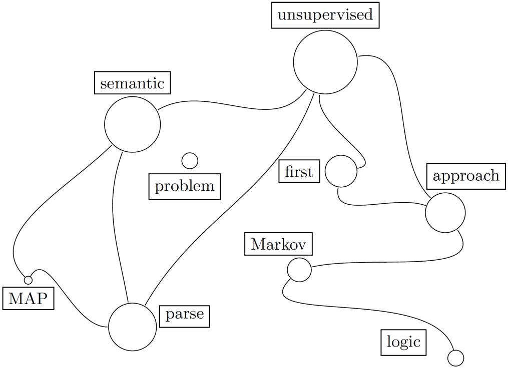
We present the first unsupervised approach to the problem of learning a semantic[16] parser, using Markov logic . Our USP system transforms dependency trees into quasi-logical forms, recursively induces lambda forms from these, and clusters them to abstract away syntactic variations of the same meaning. The MAP semantic[51] 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[90], DIRT and an informed baseline on both precision and recall on this task.

- $p(\text{textrunner}) = \frac{1}{90} = 0.011$
- $p(\text{semantic}) = \frac{1}{2} + \frac{1}{16} + \frac{1}{51} = 0.582$

$\widetilde{p}$ is set to the normalized weights for each candidate word as follows:

$$\widetilde{p} = \left[ \frac{p_1}{p_1+p_2+\ldots+p_{|V|}}, \frac{p_2}{p_1+p_2+\ldots+p_{|V|}}, \ldots, \frac{p_{|V|}}{p_1+p_2+\ldots+p_{|V|}} \right]$$

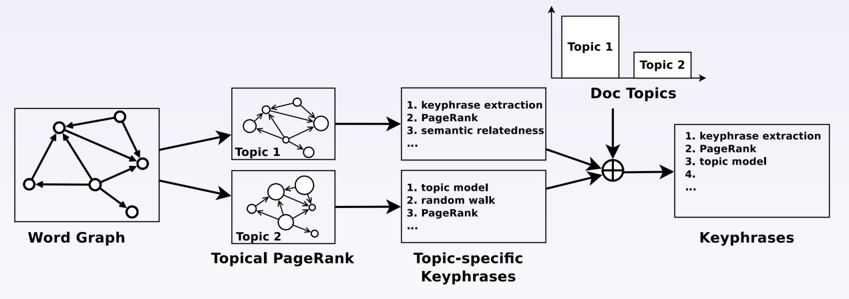
# Scoring Multi-Word Phrases




- Multi-word phrases or $n$-grams are scored by using the sum of scores of individual words that comprise the phrase [Wan and Xiao, 2008].
- The top $k$ ranked phrases are predicted as keyphrases.

# Topic-Decomposed PageRank

- Another Biased PageRank...
    - Topical PageRank for Keyphrase Extraction (TPR)




[Liu et al., 2010].

# Experiments and Results

Datasets:

- We evaluated the performance of PositionRank on three datasets:
  - The proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD) and the World Wide Web Conference (WWW) (Gollapalli and Caragea, 2014);
  - Nguyen dataset of research papers on various disciplines (Nguyen and Kan, 2007).
- The author-input keyworks were used as gold-standard for evaluation.

Table: Summary of datasets:

| Dataset | #Docs | Kp | AvgKp | unigrams | bigrams | trigrams | n-grams ($n \geq 4$) |
|---|---|---|---|---|---|---|---|
| KDD | 834 | 3093 | 3.70 | 810 | 1770 | 471 | 42 |
| WWW | 1350 | 6405 | 4.74 | 2254 | 3139 | 931 | 81 |
| Nguyen | 211 | 882 | 4.18 | 260 | 457 | 132 | 33 |

Performance measures for evaluation: Mean Reciprocal Rank, Precision, Recall and F1-score.

# What is the impact of aggregating information from all positions of a word over using first position only?

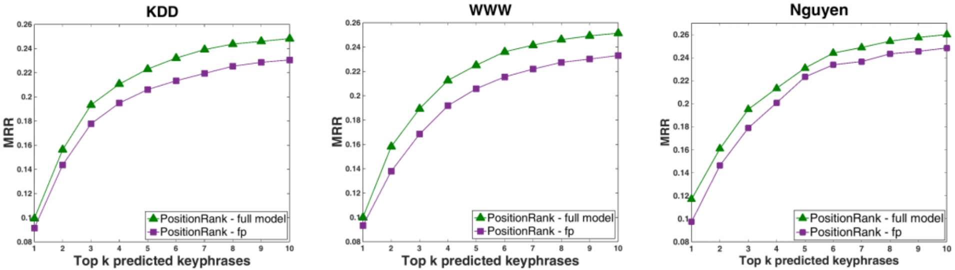



Figure: The comparison of PositionRank that aggregates information from all positions of a word's occurrences (full model) with the PositionRank that uses only the first position of a word (fp).

# How well does position information aid in unsupervised keyphrase extraction from research papers?

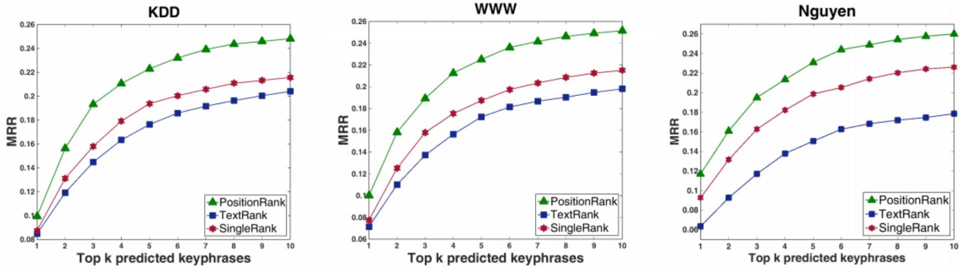



Figure: : MRR curves for PositionRank and two unbiased PageRank-based models that do not consider position information.

# How does PositionRank compare with other previous methods?

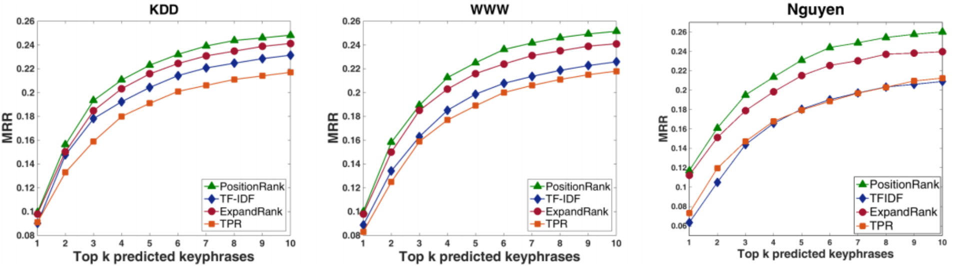



Figure: MRR curves for PositionRank and previous methods on the three datasets.

# Overall Performance Summary of PositionRank

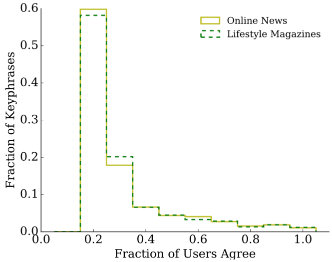
| Dataset | Unsupervised method | Top2 | | | Top4 | | | Top6 | | | Top8 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P% | R% | F1% | P% | R% | F1% | P% | R% | F1% | P% | R% | F1% |
| KDD | PositionRank | **11.1** | **5.6** | **7.3** | **10.8** | **11.1** | **10.6** | **9.8** | **15.3** | **11.6** | **9.2** | **18.9** | **12.1** |
| | PositionRank-fp | 10.3 | 5.3 | 6.8 | 10.2 | 10.4 | 10.0 | 9.1 | 13.8 | 10.9 | 8.6 | 17.2 | 11.3 |
| | TF-IDF | 10.5 | 5.2 | 6.8 | 9.6 | 9.7 | 9.4 | 9.2 | 13.8 | 10.7 | 8.7 | 17.4 | 11.3 |
| | TextRank | 8.1 | 4.0 | 5.3 | 8.3 | 8.5 | 8.1 | 8.1 | 12.3 | 9.4 | 7.6 | 15.3 | 9.8 |
| | SingleRank | 9.1 | 4.6 | 6.0 | 9.3 | 9.4 | 9.0 | 8.7 | 13.1 | 10.1 | 8.1 | 16.4 | 10.6 |
| | ExpandRank | 10.3 | 5.5 | 6.9 | 10.4 | 10.7 | 10.1 | 9.2 | 14.5 | 10.9 | 8.4 | 17.5 | 11.0 |
| | TPR | 9.3 | 4.8 | 6.2 | 9.1 | 9.3 | 8.9 | 8.8 | 13.4 | 10.3 | 8.0 | 16.2 | 10.4 |
| WWW | PositionRank | **11.3** | **5.3** | **7.0** | **11.3** | **10.5** | **10.5** | **10.8** | **14.9** | **12.1** | **9.9** | **18.1** | **12.3** |
| | PositionRank-fp | 9.6 | 4.5 | 6.0 | 10.3 | 9.6 | 9.6 | 10.1 | 13.8 | 11.2 | 9.4 | 17.2 | 11.7 |
| | TF-IDF | 9.5 | 4.5 | 5.9 | 10.0 | 9.3 | 9.3 | 9.6 | 13.3 | 10.7 | 9.1 | 16.8 | 11.4 |
| | TextRank | 7.7 | 3.7 | 4.8 | 8.6 | 7.9 | 8.0 | 8.1 | 12.3 | 9.8 | 8.2 | 15.2 | 10.2 |
| | SingleRank | 9.1 | 4.2 | 5.6 | 9.6 | 8.9 | 8.9 | 9.3 | 13.0 | 10.5 | 8.8 | 16.3 | 11.0 |
| | ExpandRank | 10.4 | 5.3 | 6.7 | 10.4 | 10.6 | 10.1 | 9.5 | 14.7 | 11.2 | 8.6 | 17.7 | 11.2 |
| | TPR | 8.8 | 4.2 | 5.5 | 9.6 | 8.9 | 8.9 | 9.5 | 13.2 | 10.7 | 9.0 | 16.5 | 11.2 |
| Nguyen | PositionRank | **10.5** | **5.8** | **7.3** | **10.6** | **11.4** | **10.7** | **11.0** | **17.2** | **13.0** | **10.2** | **21.1** | **13.5** |
| | PositionRank-fp | 10.0 | 5.4 | 6.8 | 10.4 | 11.1 | 10.5 | 11.2 | 17.4 | 13.2 | 10.1 | 21.2 | 13.3 |
| | TF-IDF | 7.3 | 4.0 | 5.0 | 9.5 | 10.3 | 9.6 | 9.1 | 14.4 | 10.9 | 8.9 | 18.9 | 11.8 |
| | TextRank | 6.3 | 3.6 | 4.5 | 7.4 | 7.4 | 7.2 | 7.8 | 11.9 | 9.1 | 7.2 | 14.8 | 9.4 |
| | SingleRank | 9.0 | 5.2 | 6.4 | 9.5 | 9.9 | 9.4 | 9.2 | 14.5 | 11.0 | 8.9 | 18.3 | 11.6 |
| | ExpandRank | 9.5 | 5.3 | 6.6 | 9.5 | 10.2 | 9.5 | 9.1 | 14.4 | 10.8 | 8.7 | 18.3 | 11.4 |
| | TPR | 8.7 | 4.9 | 6.1 | 9.1 | 9.5 | 9.0 | 8.8 | 13.8 | 10.5 | 8.8 | 18.0 | 11.5 |

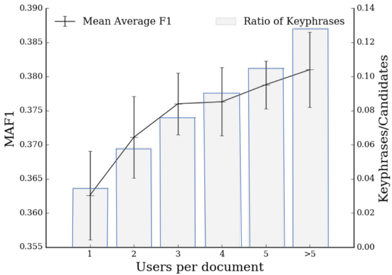
# Summary

- Developments in keyphrase extraction are central to ***document understanding, knowledge discovery and organization*** and have a direct impact on the development of digital libraries.
- We proposed a novel unsupervised graph-based model, called PositionRank, which incorporates both the position of words and their frequency
  - ***Our model outperforms strong baselines in terms of all performance measures on scholarly documents***

# Limitations and Potential Extensions

- Keyphrase extraction is very subjective




[Sterckx, Caragea, Demeester, Develder, 2016 (**EMNLP**)]

- Crowdsourcing for keyphrase extraction.
- Extentions to other CS areas and other scientific domains, e.g., ACL Anthology, PubMed, Social Science, Political Science, Ecology.

# Limitations and Potential Extensions II

- Predict terms not found in a target paper to be keyphrases (through semantic and syntactic features).

**Title: A Unified Approach for Schema Matching, Coreference and Canonicalization by Wick et al.**

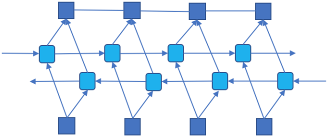
**ABSTRACT**

The automatic consolidation of database records from many heterogeneous sources into a single repository requires solving several information integration tasks. Although tasks such as coreference, schema matching, and canonicalization are closely related, they are most commonly studied in isolation. Systems that do tackle multiple integration problems traditionally solve each independently, allowing errors to propagate from one task to another. In this paper, we describe a discriminatively-trained model that reasons about schema matching, coreference, and canonicalization jointly. We evaluate our model on a real-world data set of people and demonstrate that simultaneously solving these tasks reduces errors over a cascaded or isolated approach. Our experiments show that a joint model is able to improve substantially over systems that either solve each task in isolation or with the conventional cascade. We demonstrate nearly a 50% error reduction for coreference and a 40% error reduction for schema matching.

**Keywords**

Data Integration, Coreference, Schema Matching, Canonicalization, Conditional Random Field, Weighted Logic

- ... and consider dependencies between the labels and between the words in the text.



Bi-LSTM-CRF

# References


- C. Florescu and C. Caragea (2017). PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (**ACL '17**).*
- 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**).*
- 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, F. Bulgarov, A. Godea, and S. Das Gollapalli. (2014). Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (**EMNLP '14**).*

# Thank you!

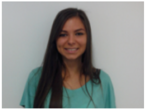

- **Acknowledgements:**

National Science Foundation
WHERE DISCOVERIES BEGIN

CiteSeer$^x$


Florin Bulgarov


Sujatha Das Gollapalli


Andreea Godea


Ashwini Tonge


Lucas Sterckx


Corina Florescu


Alina Ciobanu


Kishore Neppalli


MLg Machine Learning Group
KSU Computer Science