AUTOMATIC KEYPHRASE EXTRACTION FROM TEXT: A WALK-THROUGH

Eirini Papagiannopoulou
Aristotle University of Thessaloniki, Greece
epapagia@csd.auth.gr

Ricardo Campos
Polytechnic Institute of Tomar, Portugal
INESC TEC, Portugal
ricardo.campos@ipt.pt

Grigorios Tsoumakas
Aristotle University of Thessaloniki, Greece
greg@csd.auth.gr

ARISTOTLE UNIVERSITY OF THESSALONIKI

ipt Instituto Politécnico de Tomar

LIAAD INESCTEC LABORATÓRIO ASSOCIADO

U.PORTO

ECAI 2020 Santiago de Compostela

August 29, 2020

THE KEYPHRASE EXTRACTION (KE) TASK

Output a set of phrases that together summarize the main topics in a document

Keyphrases:

approximate search

clustering

high-dimensional index

similarity search

792 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 14, NO. 4, JULY/AUGUST 2002

Clustering for Approximate Similarity Search in High-Dimensional Spaces

Chen Li, *Member, IEEE*, Edward Chang, *Member, IEEE*, Hector Garcia-Molina, and Gio Wiederhold, *Fellow, IEEE*

Abstract—In this paper, we present a clustering and indexing paradigm (called Clindex) for high-dimensional search spaces. The scheme is designed for approximate similarity searches, where one would like to find many of the data points near a target point, but where one can tolerate missing a few near points. For such searches, our scheme can find near points with high recall in very few IOs and perform significantly better than other approaches. Our scheme is based on finding clusters and, then, building a simple but efficient index for them. We analyze the trade-offs involved in clustering and building such an index structure, and present extensive experimental results.

Index Terms—Approximate search, clustering, high-dimensional index, similarity search.

1 Introduction

Similarity search has generated a great deal of interest lately because of applications such as similar text/image search and document/image copy detection. These applications characterize objects (e.g., images and text documents) as *feature vectors* in very high-dimensional spaces [13], [23]. A user submits a query object to a search engine and the search engine returns objects that are similar to the query object. The degree of similarity between two objects is measured by some distance function between their feature vectors. The search is performed by returning the objects that are *nearest* to the query object in high-dimensional spaces.

Nearest-neighbor search is inherently expensive, especially when there are a large number of dimensions. First, the search space can grow exponentially with the number of dimensions. Second, there is simply no way to build an index on disk such that *all* nearest neighbors to *any* query point are physically adjacent on disk. (We discuss this "curse of dimensionality" in more detail in Section 2.) Fortunately, in many cases it is sufficient to perform an *approximate search* that returns many but not all nearest neighbors [2], [17], [15], [27], [29], [30]. (A feature vector is often an approximate characterization of an object, so we are already dealing with approximations anyway.) For instance, in content-based image retrieval [11], [19], [40] and document copy detection [9], [13], [20], it is usually acceptable to miss a small fraction of the target objects. Thus, it is not necessary to pay the high price of an exact search.

In this paper, we present a new similarity-search paradigm: a clustering/indexing combined scheme that achieves approximate similarity search with high efficiency. We call this approach *Clindex* (CLustering for INDEXing). Under Clindex, the data set is first partitioned into "similar" clusters. To improve IO efficiency, each cluster is then stored in a sequential file, and a mapping table is built for indexing the clusters. To answer a query, clusters that are near the query point are retrieved into main memory. Clindex then ranks the objects in the retrieved clusters by their distances to the query object, and returns the top, say k, objects as the result.

Both clustering and indexing have been intensively researched (we survey related work in Section 2), but these two subjects have been studied separately with different optimization objectives: clustering optimizes classification accuracy, while indexing maximizes IO efficiency for information retrieval. Because of these different goals, indexing schemes often do not preserve the clusters of data sets, and randomly project objects that are close (hence similar) in high-dimensional spaces onto a 2D plane (the disk geometry). This is analogous to breaking a vase (cluster) apart to fit it into the minimum number of small packing boxes (disk blocks). Although the space required to store the vase may be reduced, finding the boxes in a high-dimensional warehouse to restore the vase requires a great deal of effort.

In this study we show that by; 1) taking advantage of the clustering structures of a data set, and 2) taking advantage of sequential disk IOs by storing each cluster in a sequential file, we can achieve efficient approximate similarity search in high-dimensional spaces with high accuracy. We examine a variety of clustering algorithms on two different data sets to show that Clindex works well when 1) a data set can be grouped into clusters and 2) an algorithm can successfully find these clusters. As a part of our study, we also explore a very natural algorithm called *Cluster Forming* (CF) that achieves a preprocessing cost that is linear in the dimensionality and

- *C. Li, H. Garcia-Molina, and G. Wiederhold are with the Department of Computer Science Stanford University, Stanford, CA 94306. E-mail: {chenli, hector, gio}@db.stanford.edu.*
- *E. Chang is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106. E-mail: echang@ece.ucsb.edu.*

Manuscript received 22 Nov. 1999; revised 1 Aug. 2000; accepted 22 Jan. 2001; posted to Digital Library 7 Sept. 2001. For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 110982.

1041-4347/02/$17.00 © 2002 IEEE

KE is a core NLP task

Document summarization, Information retrieval, Document clustering/classification, Thesaurus building, Information visualization

KE is a difficult task (redundancy/infrequency errors, etc.)

TO SUPERVISE OR NOT TO SUPERVISE?

Disadvantages of unsupervised approaches

- Worse performance compared to supervised approaches

Disadvantages of supervised approaches

- Time and money to obtain annotations
- Annotations are often subjective
- May not generalize successfully to a different corpus

ECAI 2020 Santiago de Compostela

KE SURVEYS

- Papagiannopoulou, E., & Tsoumakas, G. (2020). A review of keyphrase extraction. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(2), e1339.

- Merrouni, Z. A., Frikh, B., & Ouhbi, B. (2020). Automatic Keyphrase Extraction: a Survey and Trends. Journal of Intelligent Information Systems, 54, 391-424.

- Ygor Gallina, Florian Boudin, and Béatrice Daille. 2020. Large-Scale Evaluation of Keyphrase Extraction Models. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020 (JCDL '20). Association for Computing Machinery, New York, NY, USA, 271–278.

- Çano, E., & Bojar, O. (2019). Keyphrase Generation: A Multi-Aspect Survey. In Proceedings of the 25th Conference of the Open Innovations Association (FRUCT'19). Helsinki, Finland.

OUTLINE

- **Unsupervised methods**
- **Supervised methods**
- **Data/Software/Evaluation**
- **Practical part**

UNSUPERVISED METHODS

Unsupervised

- Statistics-based
- Graph-based
- Embeddings-based

Basic steps

1. Text pre-processing – optional steps (that depend on the solution conceived): stopwords’ removal, part-of-speech (POS) tagging, tokenization, normalization (stemming, lemmatization), etc.
2. Selection of the candidate lexical units based on heuristics
3. Formation of the keyphrases in case the lexical units of step 2 are unigrams
4. Scoring/ranking of the candidate lexical units

STATISTICS-BASED METHODS

Approaches

- ✓ Tf-Idf: the baseline of the task
- ✓ KP-Miner (El-Beltagy and Rafea, 2009): exploits various types of statistical information
- ✓ YAKE (Campos et al., 2018): uses new statistical metrics that capture context information
- ✓ … there are many more methods, which we will not detail here (for schedule reasons)

ECAI 2020 Santiago de Compostela

STATISTICS-BASED METHODS: TF-IDF

THE BASELINE OF THE TASK

$$TfIdf = Tf \times Idf$$

$$Tf: raw\ phrase\ frequency$$

$$Idf = log_2 \frac{N}{1 + |d \in D: phrase \in d|}$$

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.* ***Upper bounds*** *for components of a minimal set of solutions and algorithms of construction of* ***minimal generating sets*** *of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types*

Output:

linear diophantine
linear diophantine equations
diophantine
diophantine equations
nonstrict inequations
minimal generating
minimal generating sets
minimal supporting
minimal supporting set
considered types systems

STATISTICS-BASED METHODS: KP-MINER

MULTIPLE TYPES OF STATISTICAL INFORMATION

Selection of candidate phrases that:

- are not separated by punctuation marks/stopwords
- have specific least allowable seen frequency (lasf) factor, i.e., a phrase has to have appeared at least n times in the document
- have a cutoff constant (CutOff) (number of words after which if a phrase appears for the first time, it is filtered out and ignored)

Ranking of the candidate phrases considering the:

- Tf and Idf scores
- boosting factor for compound terms over the single terms

*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. **Upper bounds** for components of a minimal set of solutions and algorithms of construction of **minimal generating sets** of solutions or all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types*

Output:

linear constraints
natural numbers
linear diophantine
linear diophantine equations
diophantine equations
strict inequations
nonstrict inequations
upper bounds
minimal set
minimal generating

STATISTICS-BASED METHODS: YAKE

CONTEXT INFORMATION

Split text into individual terms

Calculation of 5 features for each term:

- Casing (W_{case}): reflects the casing aspect
- Word Positional ($W_{Position}$): values more those words that appear at the beginning of the document
- Word Frequency (W_{Freq})
- Word Relatedness to Context (W_{Rel}): computes the number of different terms that occur to the left/right side of the candidate word
- Word DifSentence ($W_{DifSentence}$) quantifies how often a candidate word appears within different sentences

$$S(w) = \frac{W_{Rel} \times W_{Position}}{W_{case} + \frac{W_{Freq}}{W_{Rel}} + \frac{W_{DifSentence}}{W_{Rel}}}$$

Form candidate phrases (n-grams) from contiguous sequences with a sliding window of n (best results are achieved when n is set to 3)

Score each phrase:

$$S(p) = \frac{\prod_{w \in p} S(w)}{Tf(p) \times (1 + \sum_{w \in p} S(w))}$$

*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. **Upper bounds** for components of a minimal set of solutions and algorithms of construction of **minimal generating sets** of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types*

Output:

linear diophantine equations
natural numbers
linear constraints
linear diophantine
considered types systems
diophantine equations
systems
minimal supporting set
set
compatibility

DISCUSSION ON STATISTICS-BASED METHODS

Useful types of statistical information:

- Tf, Idf, TfIdf
- heuristics, e.g., lasf and a cutoff constant, casing, position
- context info, e.g., Word Relatedness to Context, Word DifSentence

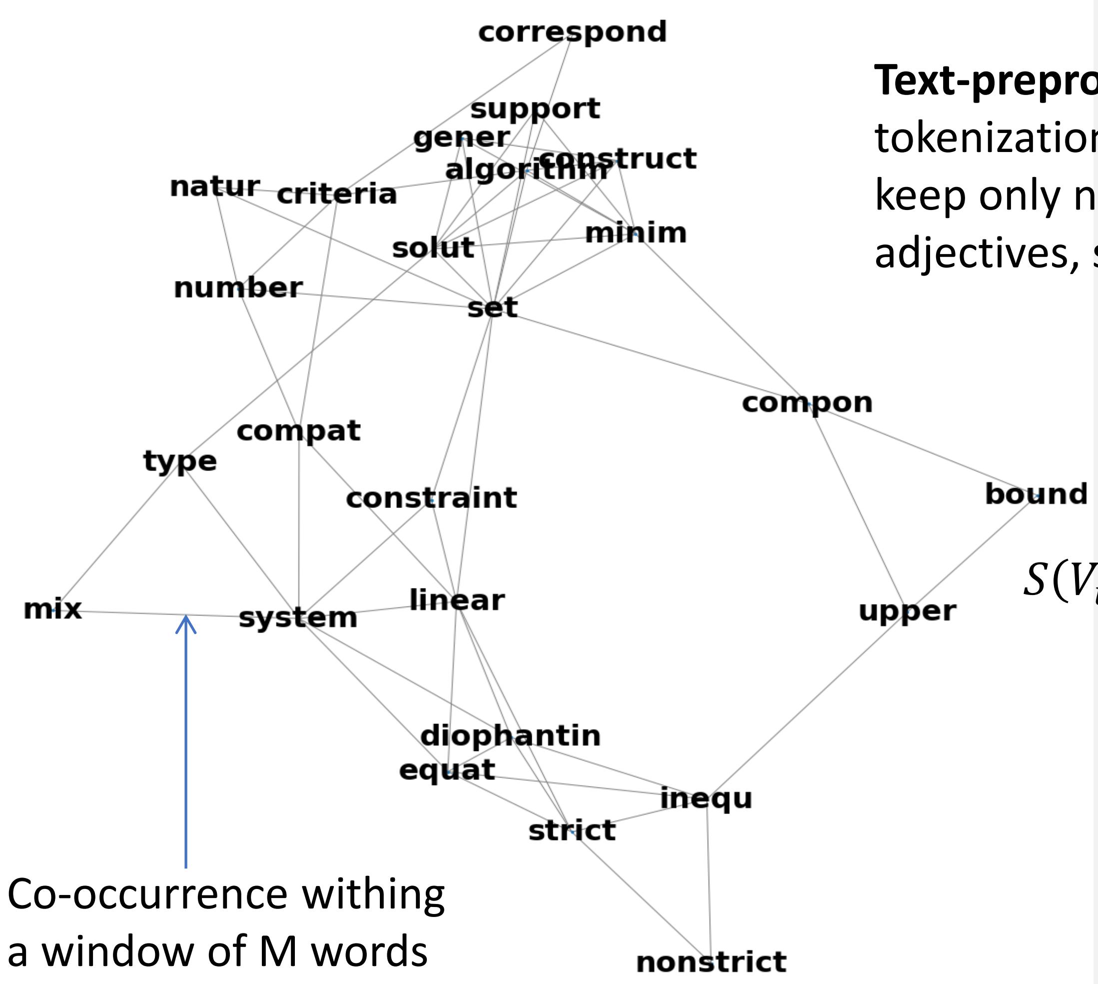
GRAPH-BASED METHODS

Approaches

- ✓ TextRank (Mihalcea and Tarau (2004), SingleRank (Wan and Xiao, 2008), PositionRank (Florescu and Caragea, 2017): classic methods
- ✓ ExpandRank (Wan and Xiao, 2008), CiteTextRank (Gollapalli and Caragea, 2014): incorporating information from similar documents/citation networks
- ✓ TopicRank (Bougouin et al., 2013), Single Topical PageRank (Sterckx et al., 2015a): topic-based methods
- ✓ Wang et al. (2015), Key2Vec (Mahata et al., 2018): use of semantics

GRAPH-BASED METHODS: TEXTRANK

CLASSIC METHODS



Co-occurrence withing a window of M words

Text-preprocessing: tokenization, POS tagging, keep only nouns & adjectives, stemming

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. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types

$$S(V_i) = (1-\lambda) + \lambda * \sum_{j \in N(V_i)} \frac{1}{N(V_j)} S(V_j)$$

$N(V_i)$: *set of* V_i'*s neighbours*

$N(V_j)$: *set of* V_j'*s neighbours*

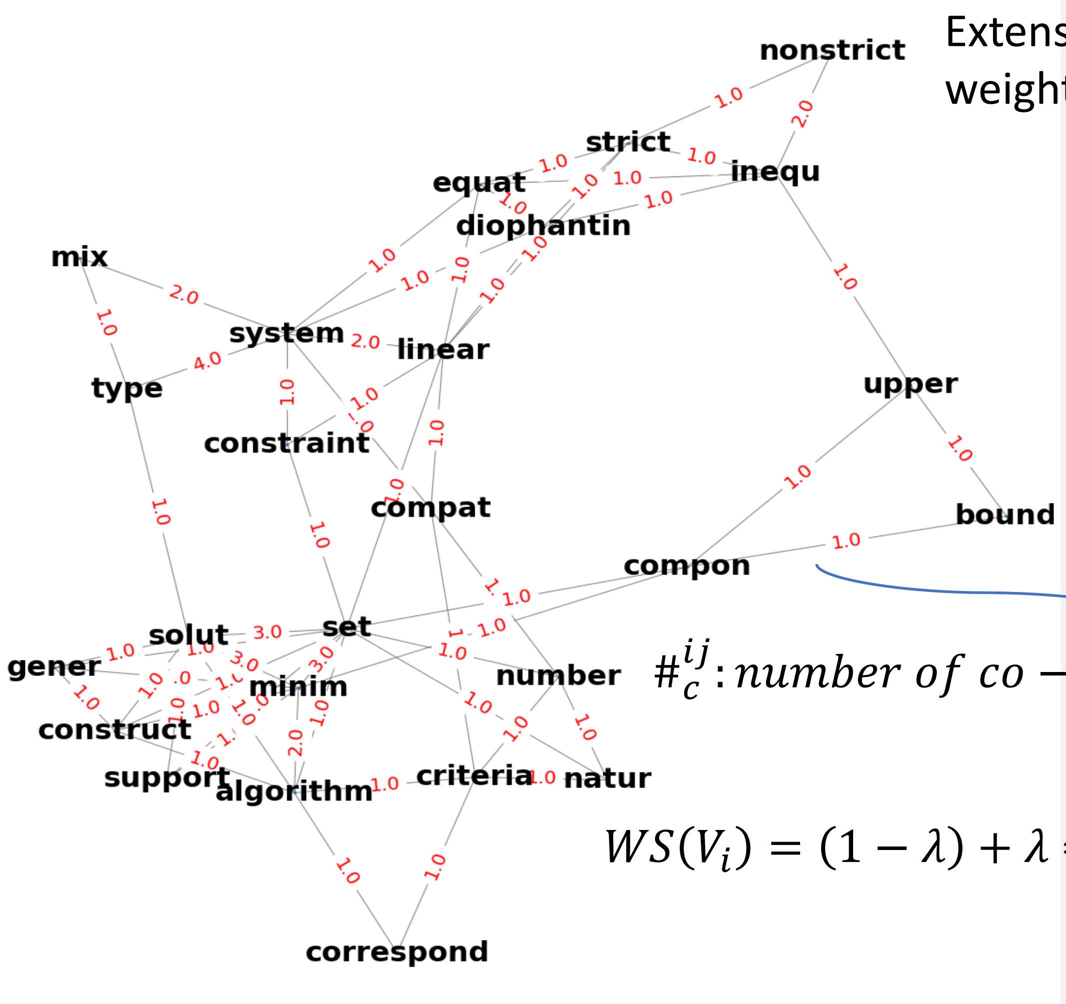
λ: *the probability of jumping from one node to another*

Output:
- minimal set
- **strict inequations**
- set
- systems
- linear
- solutions
- minimal
- algorithms
- inequations

Phrase formation/scoring: adjacent words in the text that belong to the top N scored words

GRAPH-BASED METHODS: SINGLERANK

CLASSIC METHODS



Extension to TextRank: weights to edges

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.* ***Upper bounds*** *for components of a minimal set of solutions and algorithms of construction of* ***minimal generating sets*** *of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types*

$\#_c^{ij}: number\ of\ co-occurrences\ of\ words\ i\ \&\ j$

$$WS(V_i) = (1-\lambda) + \lambda * \sum_{j \in N(V_i)} \frac{\#_c^{ij}}{\sum_{v_k \in N(V_j)} \#_c^{jk}} WS(V_j)$$

Output:

minimal generating sets
minimal supporting set
minimal set
linear diophantine equations
types systems
set
strict inequations
systems
linear constraints
nonstrict inequations

Phrase formation/scoring: for each continuous sequence of nouns and adjectives in the text the scores of the constituent words are summed up. The top-ranked candidates are returned as keyphrases.

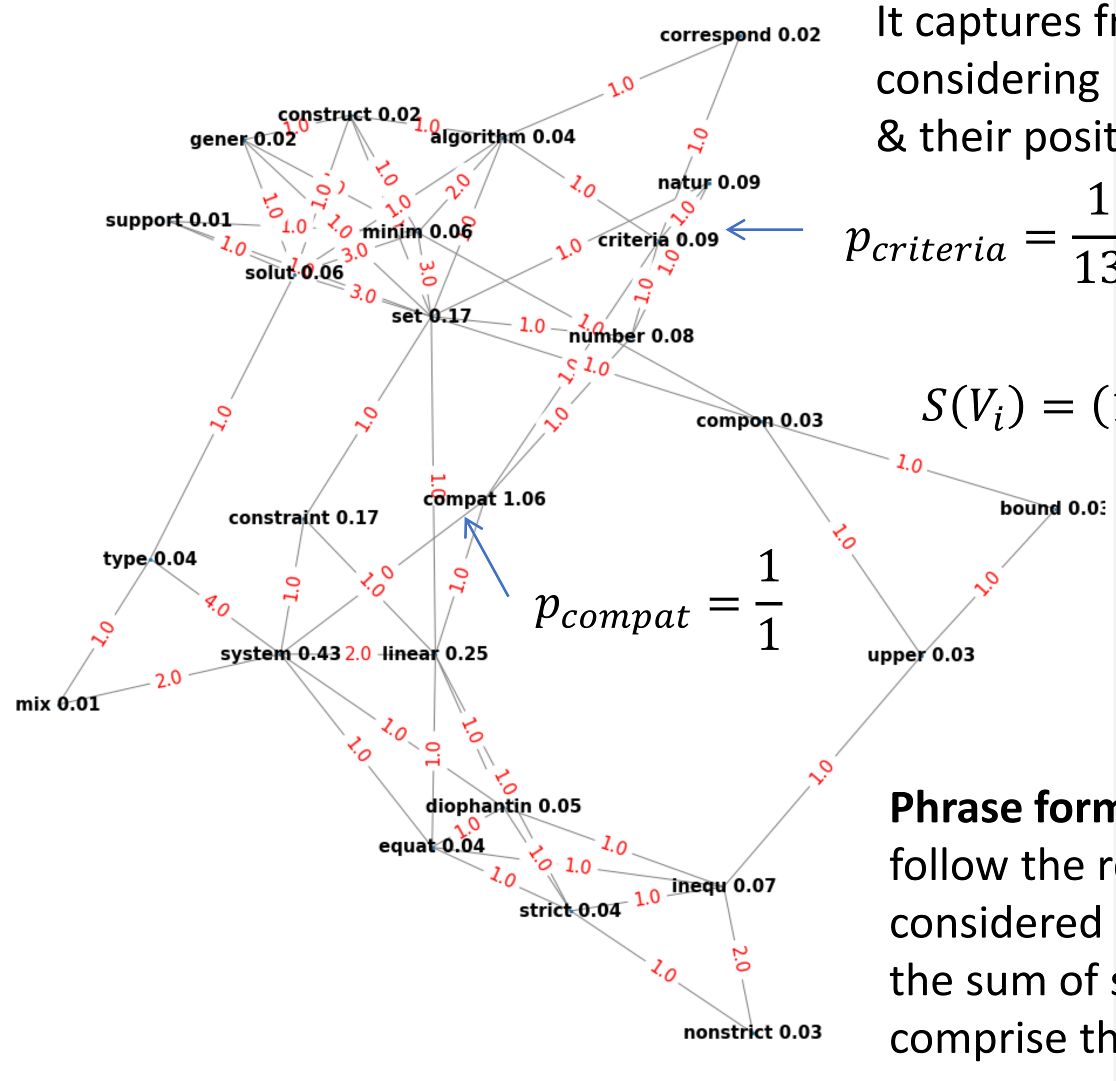
ECAI 2020 Santiago de Compostela

GRAPH-BASED METHODS: POSITIONRANK

CLASSIC METHODS

$$p = \left[\frac{p_{compat}}{p_{compat} + \cdots + p_{|V|}}, \frac{p_{criteria}}{p_{compat} + \cdots + p_{|V|}}, \ldots, \frac{p_{|V|}}{p_{compat} + \cdots + p_{|V|}}\right]$$

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. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types



It captures frequent phrases considering words' co-occurrences & their positions in the text.

$$p_{criteria} = \frac{1}{13} + \frac{1}{58}$$

$$p_{compat} = \frac{1}{1}$$

$$S(V_i) = (1 - \lambda) * p_i + \lambda * \sum_{j \in N(V_i)} \frac{w_{ji}}{N(V_j)} S(V_j)$$

Phrase formation/scoring: Noun phrases that follow the regex (adjective)*(noun)+ are considered as candidates. They are scored using the sum of scores of the individual words that comprise the phrase.

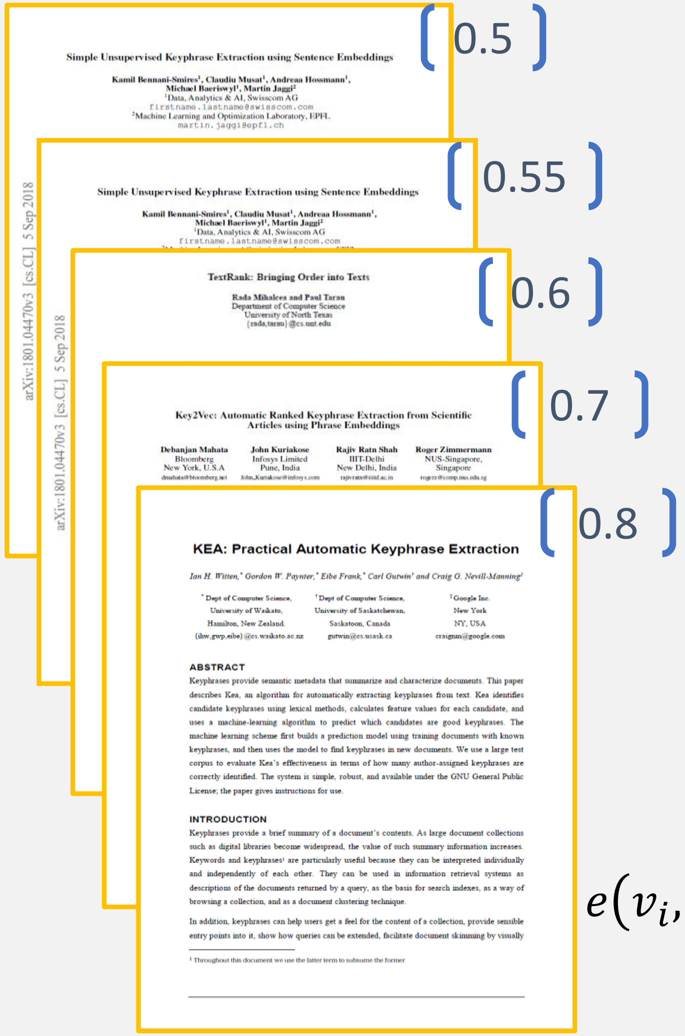
Output:

- types systems
- **minimal generating sets**
- **linear diophantine equations**
- minimal supporting set
- minimal set
- systems
- **linear constraints**
- compatibility
- set
- mixed types

GRAPH-BASED METHODS: EXPANDRANK

INFORMATION FROM SIMILAR DOCUMENTS

Extension of SR that constructs an appropriate knowledge context considering neighboring docs.



Simple Unsupervised Keyphrase Extraction using Sentence Embeddings

Kamil Bennani-Smires[1], Claudiu Musat[1], Andreea Hossmann[1], Michael Baeriswyl[1], Martin Jaggi[2]
[1]Data, Analytics & AI, Swisscom AG
firstname.lastname@swisscom.com
[2]Machine Learning and Optimization Laboratory, EPFL
martin.jaggi@epfl.ch

arXiv:1801.04470v3 [cs.CL] 5 Sep 2018

TextRank: Bringing Order into Texts

Rada Mihalcea and Paul Tarau
Department of Computer Science
University of North Texas
{rada,tarau}@cs.unt.edu

Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings

Debanjan Mahata, Bloomberg, New York, U.S.A, dmahata@bloomberg.net
John Kuriakose, Infosys Limited, Pune, India, John_Kuriakose@infosys.com
Rajiv Ratn Shah, IIIT-Delhi, New Delhi, India, rajivratn@iiitd.ac.in
Roger Zimmermann, NUS-Singapore, Singapore, rogerz@comp.nus.edu.sg

KEA: Practical Automatic Keyphrase Extraction

Ian H. Witten,* Gordon W. Paynter,* Eibe Frank,* Carl Gutwin† and Craig G. Nevill-Manning‡

* Dept of Computer Science, University of Waikato, Hamilton, New Zealand. {ihw,gwp,eibe}@cs.waikato.ac.nz
† Dept of Computer Science, University of Saskatchewan, Saskatoon, Canada. gutwin@cs.usask.ca
‡ Google Inc. New York NY, USA craignm@google.com

ABSTRACT

Keyphrases provide semantic metadata that summarize and characterize documents. This paper describes Kea, an algorithm for automatically extracting keyphrases from text. Kea identifies candidate keyphrases using lexical methods, calculates feature values for each candidate, and uses a machine-learning algorithm to predict which candidates are good keyphrases. The machine learning scheme first builds a prediction model using training documents with known keyphrases, and then uses the model to find keyphrases in new documents. We use a large test corpus to evaluate Kea's effectiveness in terms of how many author-assigned keyphrases are correctly identified. The system is simple, robust, and available under the GNU General Public License; the paper gives instructions for use.

INTRODUCTION

Keyphrases provide a brief summary of a document's contents. As large document collections such as digital libraries become widespread, the value of such summary information increases. Keywords and keyphrases[1] are particularly useful because they can be interpreted individually and independently of each other. They can be used in information retrieval systems as descriptions of the documents returned by a query, as the basis for search indexes, as a way of browsing a collection, and as a document clustering technique.

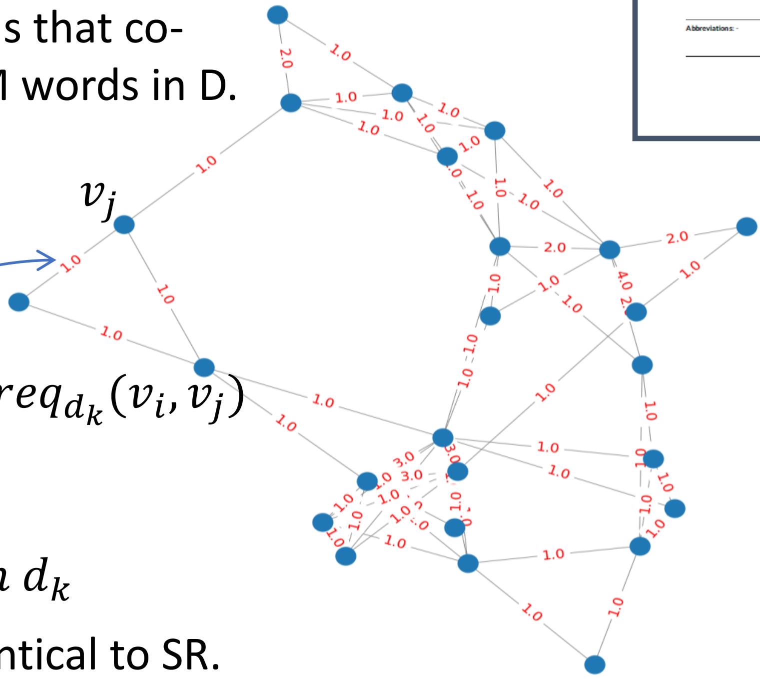
In addition, keyphrases can help users get a feel for the content of a collection, provide sensible entry points into it, show how queries can be extended, facilitate document skimming by visually

[1] Throughout this document we use the latter term to subsume the former

1. Each doc is represented by a TfIdf vector. From a set of docs, find the k nearest neighbors, ending up to a target set of k+1 docs (D).

$sim(d_0, d_k)$: $cosine\ similarity\ between\ d_0\ \&\ d_k$

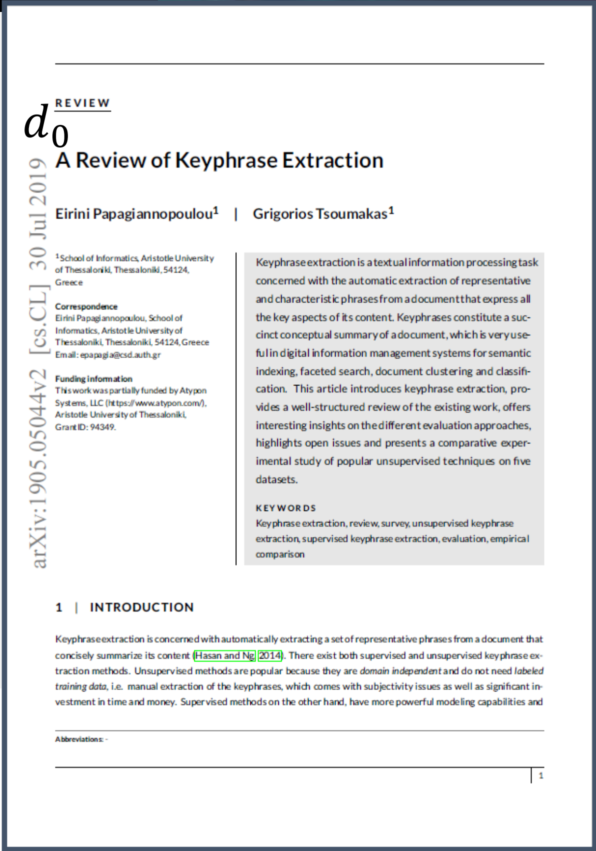
2. Graph construction based on the set D: edges between the words that co-occur within a window of M words in D.



$$e(v_i, v_j) = \sum_{d_k \in D} sim(d_0, d_k) \times freq_{d_k}(v_i, v_j)$$

$freq_{d_k}$: $co-occurrence\ frequency\ of\ v_i,\ v_j\ in\ d_k$

3. Once the graph is constructed, the rest procedure is identical to SR.



REVIEW

A Review of Keyphrase Extraction

Eirini Papagiannopoulou[1] | Grigorios Tsoumakas[1]

[1]School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece

Correspondence
Eirini Papagiannopoulou, School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece
Email: epapagia@csd.auth.gr

Funding information
This work was partially funded by Atypon Systems, LLC (https://www.atypon.com/), Aristotle University of Thessaloniki, Grant ID: 94349.

Keyphrase extraction is a textual information processing task concerned with the automatic extraction of representative and characteristic phrases from a document that express all the key aspects of its content. Keyphrases constitute a succinct conceptual summary of a document, which is very useful in digital information management systems for semantic indexing, faceted search, document clustering and classification. This article introduces keyphrase extraction, provides a well-structured review of the existing work, offers interesting insights on the different evaluation approaches, highlights open issues and presents a comparative experimental study of popular unsupervised techniques on five datasets.

KEYWORDS
Keyphrase extraction, review, survey, unsupervised keyphrase extraction, supervised keyphrase extraction, evaluation, empirical comparison

arXiv:1905.05044v2 [cs.CL] 30 Jul 2019

1 | INTRODUCTION

Keyphrase extraction is concerned with automatically extracting a set of representative phrases from a document that concisely summarize its content (Hasan and Ng 2014). There exist both supervised and unsupervised keyphrase extraction methods. Unsupervised methods are popular because they are *domain independent* and do not need *labeled training data*, i.e. manual extraction of the keyphrases, which comes with subjectivity issues as well as significant investment in time and money. Supervised methods on the other hand, have more powerful modeling capabilities and

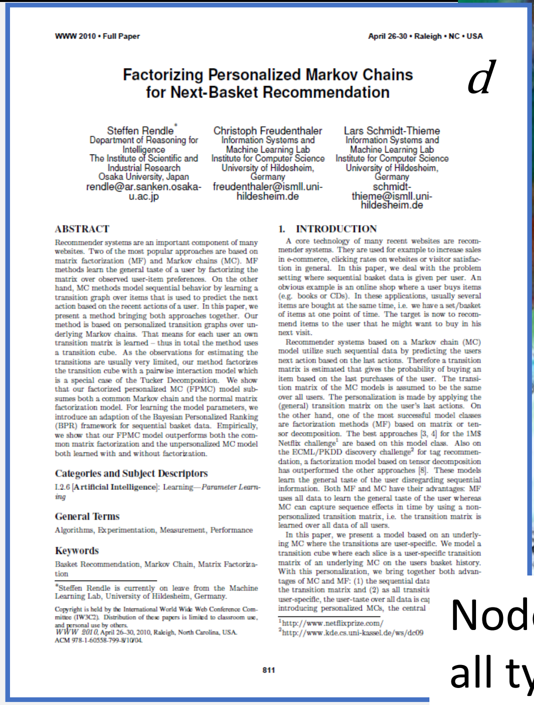
Abbreviations: -

1

GRAPH-BASED METHODS: CITETEXTRANK

INFORMATION FROM CITATION NETWORKS

Global context: doc's content


WWW 2010 • Full Paper April 26-30 • Raleigh • NC • USA

Factorizing Personalized Markov Chains for Next-Basket Recommendation

Steffen Rendle* Department of Reasoning for Intelligence The Institute of Scientific and Industrial Research Osaka University, Japan rendle@ar.sanken.osaka-u.ac.jp

Christoph Freudenthaler Information Systems and Machine Learning Lab Institute for Computer Science University of Hildesheim, Germany freudenthaler@ismll.uni-hildesheim.de

Lars Schmidt-Thieme Information Systems and Machine Learning Lab Institute for Computer Science University of Hildesheim, Germany schmidt-thieme@ismll.uni-hildesheim.de

ABSTRACT

1. INTRODUCTION

Categories and Subject Descriptors

General Terms

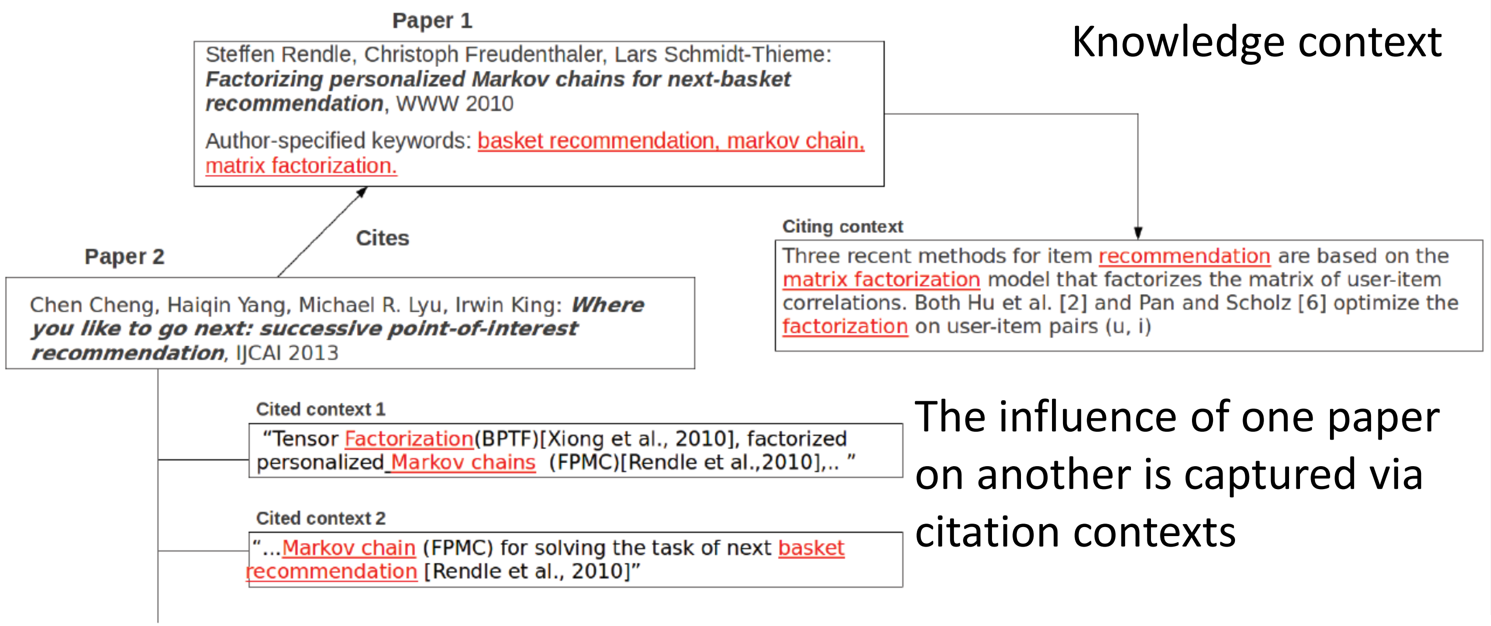
Algorithms, Experimentation, Measurement, Performance

Keywords

Basket Recommendation, Markov Chain, Matrix Factorization

811

Knowledge context



The influence of one paper on another is captured via citation contexts

Gollapalli, S. D. and Caragea, C. (2014) Extracting keyphrases from research papers using citation networks. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence, Québec City, Québec, Canada, July 27 -31, 2014*, 1629–1635.

Nodes: words from all types of contexts, edges based on word co-occurrences

$$e(v_i, v_j) = \sum_{t \in TC} \sum_{c \in C_t} \lambda_t \cdot sim(c,d) \cdot \#_c(v_i, v_j)$$

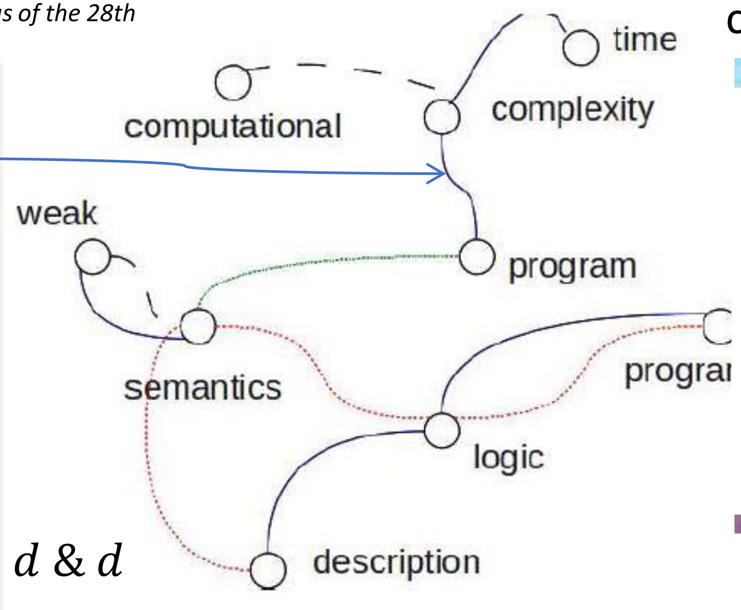
$TC: available\ context\ types\ in\ d$

$C_t: the\ set\ of\ contexts\ of\ type\ t \in TC$

$\lambda_t: weight\ for\ contexts\ of\ type\ t$

$\#_c: co-occurrence\ of\ v_i,\ v_j\ in\ context\ c$

$sim(c,d): cos.sim.between\ TfIdf\ vectors\ of\ any\ context\ c\ of\ d\ \&\ d$

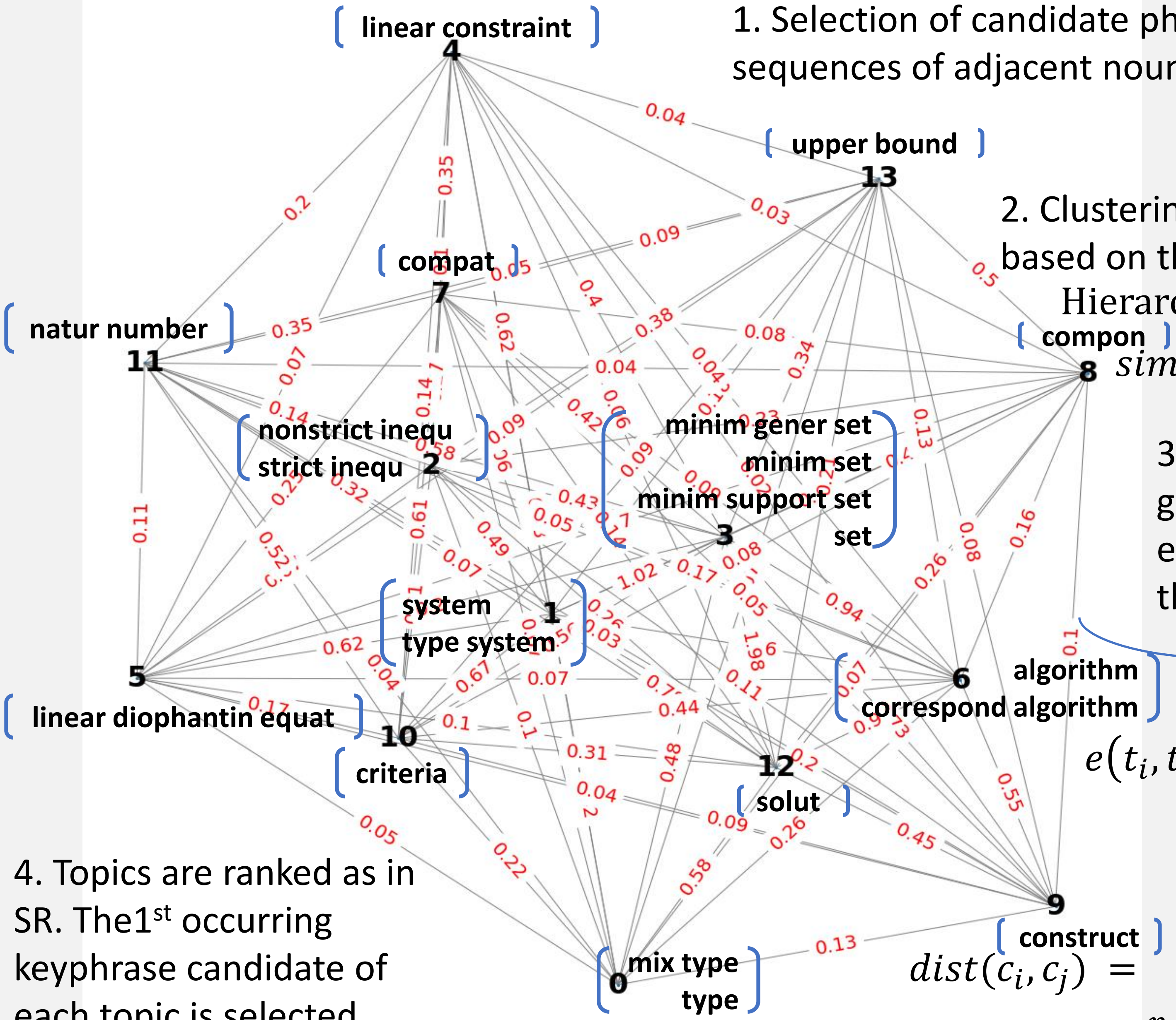


Consecutive words form phrases.
Words' scoring using PageRank.
Phrase scores by summing the words' scores.

GRAPH-BASED METHODS: TOPICRANK

TOPIC-BASED METHODS: CLUSTERING

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. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types



1. Selection of candidate phrases from sequences of adjacent nouns & adjectives

2. Clustering of similar candidates into topics based on the words they share

Hierarchical Agglomerative Clustering

$$sim(c_i, c_j) = \frac{|stems(c_i) \cap stems(c_j)|}{|stems(c_i) \cup stems(c_j)|}$$

3. Build of a complete graph: topics as nodes & edge weights based on the semantic strength

$$e(t_i, t_j) = \sum_{c_i \in t_i} \sum_{c_j \in t_j} dist(c_i, c_j)$$

$$dist(c_i, c_j) = \sum_{p_i \in pos(c_i)} \sum_{p_j \in pos(c_j)} \frac{1}{|p_i - p_j|}$$

4. Topics are ranked as in SR. The1st occurring keyphrase candidate of each topic is selected.

Output:

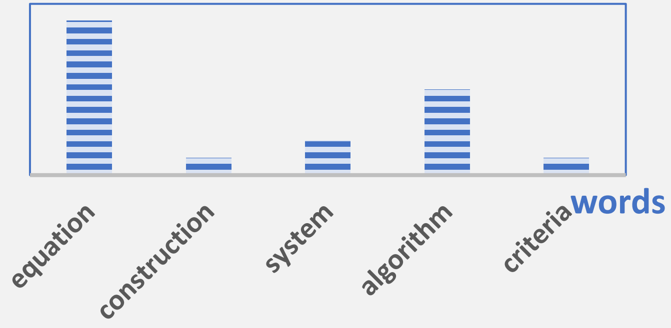
set
systems
solutions
algorithms
criteria
compatibility
strict inequations
types
construction
linear diophantine equations

GRAPH-BASED METHODS: SINGLE TOPICAL PAGERANK

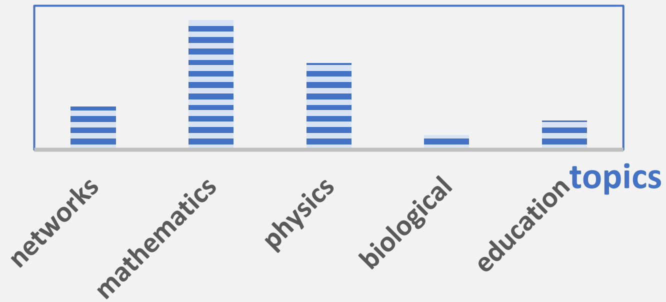
TOPIC-BASED METHODS: LDA

LDA

Topic k



Document D



1. Train an LDA topic model of 1000 topics on a large corpus.

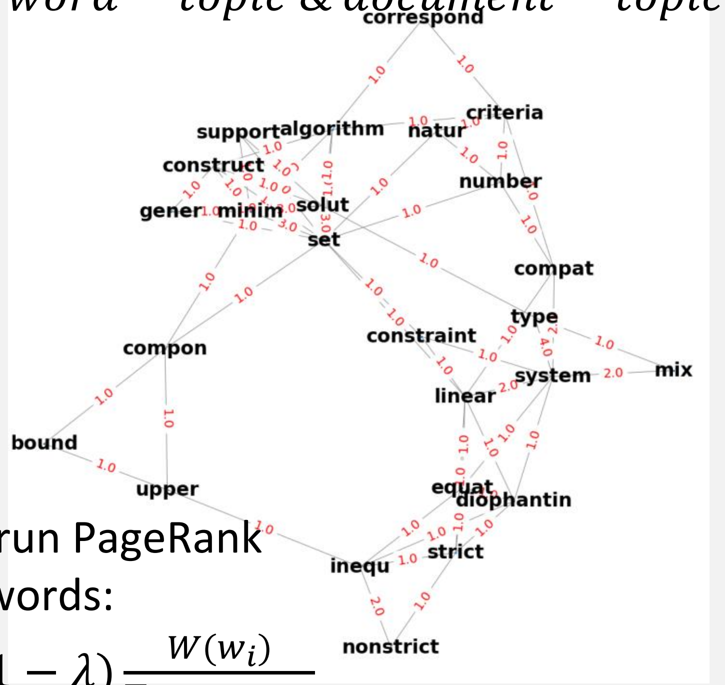
2. Apply the topic model on the target doc to get the doc-topics' probabilities vector. We also have for each word the corresponding word-topics' probabilities vector.

$$V = \{w_1, \dots, w_{|N|}\}$$

3. Compute the word topical importance for each word:
$W(w_i)$: *cosine similarity between the vectors of word − topic & document − topic probabilities*

4. Construct the graph as previously & run PageRank considering the topical importance of words:

$$S(w_i) = \lambda \sum_{j: w_j \to w_i} \frac{e(w_j, w_i)}{w_j} S(w_j) + (1 - \lambda) \frac{W(w_i)}{\sum_{w \in V} W(w)}$$



5. Phrase formation/scoring: Noun phrases following (adjective)*(noun)+. Phrase scores by summing the words' scores.

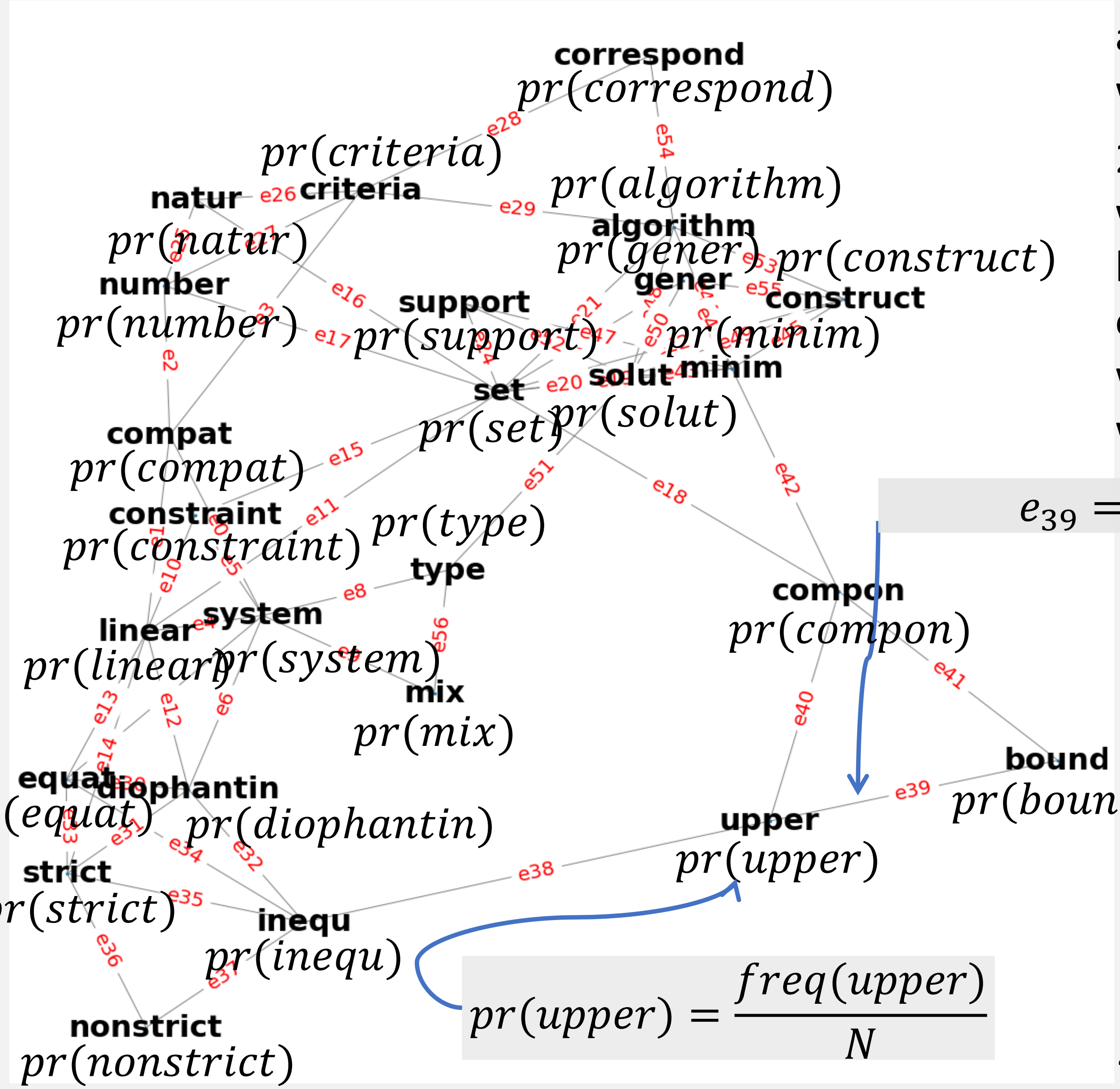
*Compatibility of systems of **linear constraints** over the **set of natural numbers**. Criteria of compatibility of a system of **linear Diophantine equations**, **strict inequations**, and **nonstrict inequations** are considered. **Upper bounds** for components of a minimal set of solutions and algorithms of construction of **minimal generating sets** of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types*

Output:
minimal generating sets
minimal supporting set
minimal set
types systems
linear diophantine equations
systems
set
linear constraints
strict inequations
mixed types

GRAPH-BASED METHODS: WANG ET AL. (2015)

USE OF SEMANTICS: STATIC PRE-TRAINED WORD EMBEDDINGS

*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. **Upper bounds** for components of a minimal set of solutions and algorithms of construction of **minimal generating sets** of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types*



$$pr(upper) = \frac{freq(upper)}{N}$$

1. Selection of noun & adjectives as candidate words. Removal of plurals.
2. Build of the graph-of-words by adding edges between the vertices that co-occur in a window of M words following the weighting scheme below:

$$e_{39} = semantic(upper, bound) \times cooccur(upper, bound)$$

$$semantic(upper, bound) = \frac{1}{1 - cosine(upper, bound)}$$

$$cooccur(upper, bound) \rightarrow PMI(upper, bound)$$

3. For each word, compute the probability distribution (pr) and assign it to the corresponding node.
4. Word scoring by running a personalised weighted PageRank.

GRAPH-BASED METHODS: KEY2VEC

USE OF SEMANTICS: STATIC DOMAIN SPECIFIC PHRASE EMBEDDINGS

1. Exhaustive text-pre-processing on a corpus of scientific abstracts

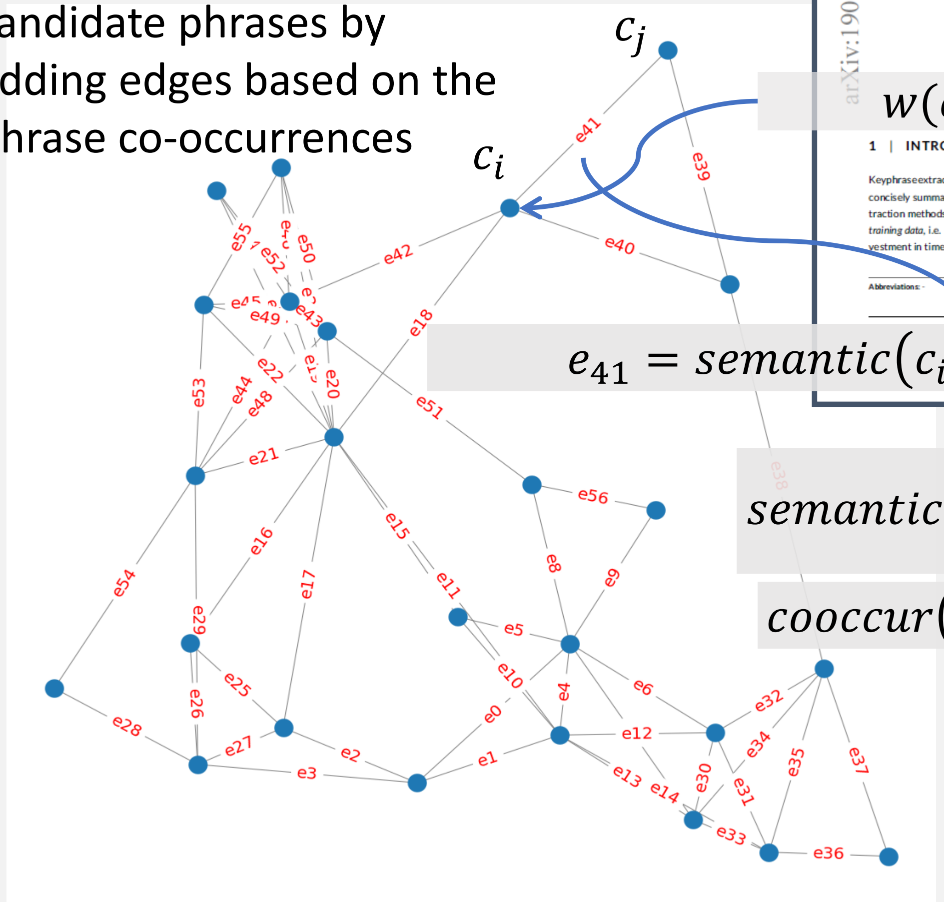
2. Training multi-word phrase embeddings using Fasttext

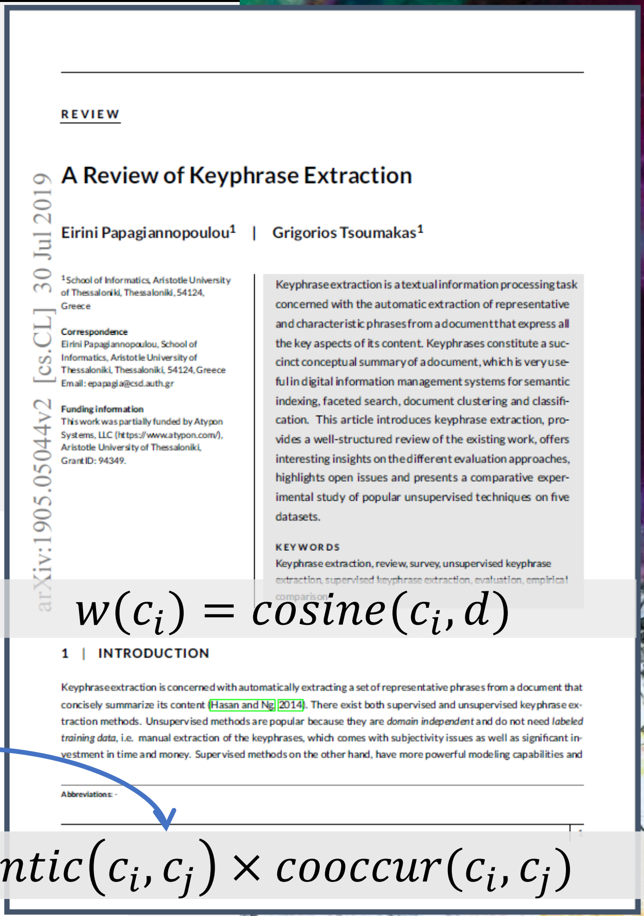
3. Same text-pre-processing on the target text & selection of candidate phrases

4. The first N sentences are extracted from the target text (theme excerpt)

- **theme excerpt → thematic phrases → theme vector $\vec{d}$ (vector addition of the thematic phrases' vectors)**
- **thematic weight w: cosine similarity between candidate's vector $\vec{c}$ and the theme vector $\vec{d}$**

5. Build of a graph of candidate phrases by adding edges based on the phrase co-occurrences





REVIEW

A Review of Keyphrase Extraction

Eirini Papagiannopoulou[1] | Grigorios Tsoumakas[1]

[1]School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece

Correspondence
Eirini Papagiannopoulou, School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece
Email: epapagia@csd.auth.gr

Funding information
This work was partially funded by Atypon Systems, LLC (https://www.atypon.com/), Aristotle University of Thessaloniki, Grant ID: 94349.

Keyphrase extraction is a textual information processing task concerned with the automatic extraction of representative and characteristic phrases from a document that express all the key aspects of its content. Keyphrases constitute a succinct conceptual summary of a document, which is very useful in digital information management systems for semantic indexing, faceted search, document clustering and classification. This article introduces keyphrase extraction, provides a well-structured review of the existing work, offers interesting insights on the different evaluation approaches, highlights open issues and presents a comparative experimental study of popular unsupervised techniques on five datasets.

KEYWORDS
Keyphrase extraction, review, survey, unsupervised keyphrase extraction, supervised keyphrase extraction, evaluation, empirical comparison

arXiv:1905.05044v2 [cs.CL] 30 Jul 2019

1 | INTRODUCTION

Keyphrase extraction is concerned with automatically extracting a set of representative phrases from a document that concisely summarize its content (Hasan and Ng, 2014). There exist both supervised and unsupervised keyphrase extraction methods. Unsupervised methods are popular because they are *domain independent* and do not need *labeled training data*, i.e. manual extraction of the keyphrases, which comes with subjectivity issues as well as significant investment in time and money. Supervised methods on the other hand, have more powerful modeling capabilities and

Abbreviations: -

$$w(c_i) = cosine(c_i, d)$$

$$e_{41} = semantic(c_i, c_j) \times cooccur(c_i, c_j)$$

$$semantic(c_i, c_j) = \frac{1}{1 - cosine(c_i, c_j)}$$

$$cooccur(c_i, c_j) \rightarrow PMI(c_i, c_j)$$

6. Phrase scoring by running a personalised weighted PageRank

DISCUSSION ON GRAPH-BASED METHODS

Useful types of information:

- context via word co-occurrences
- knowledge context via neighbouring documents or citation networks
- heuristics, e.g., position
- topic discovery via clustering or LDA
- semantics from word embeddings

EMBEDDINGS-BASED METHODS

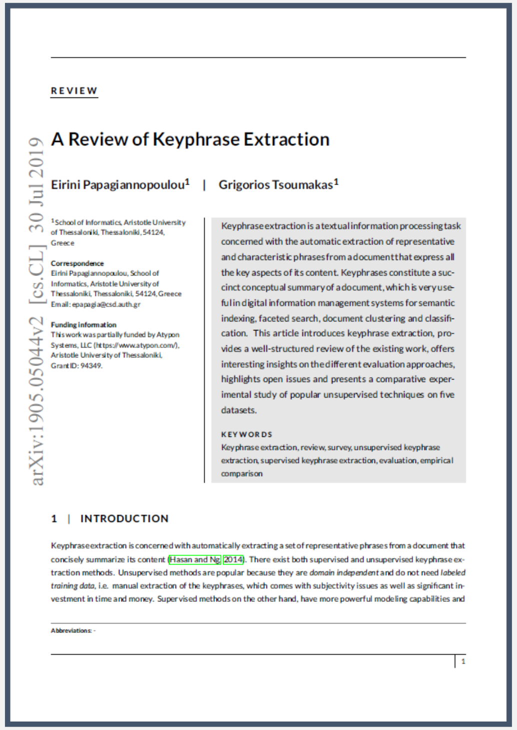
Approaches

- ✓ EmbedRank (Bennani-Smires et al., 2018): use of sentence embeddings
- ✓ RVA (Papagiannopoulou and Tsoumakas, 2018): local word embeddings
- ✓ LV (Papagiannopoulou and Tsoumakas, 2020 - Arxiv): keywords lie far from the main bulk of words in local vector space

ECAI 2020 Santiago de Compostela

EMBEDDINGS-BASED METHODS: EMBEDRANK

USE OF SENTENCE EMBEDDINGS



arXiv:1905.05044v2 [cs.CL] 30 Jul 2019

REVIEW

A Review of Keyphrase Extraction

Eirini Papagiannopoulou[1] | Grigorios Tsoumakas[1]

[1]School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece

Correspondence
Eirini Papagiannopoulou, School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece
Email: epapagia@csd.auth.gr

Funding information
This work was partially funded by Atypon Systems, LLC (https://www.atypon.com/), Aristotle University of Thessaloniki, Grant ID: 94349.

Keyphrase extraction is a textual information processing task concerned with the automatic extraction of representative and characteristic phrases from a document that express all the key aspects of its content. Keyphrases constitute a succinct conceptual summary of a document, which is very useful in digital information management systems for semantic indexing, faceted search, document clustering and classification. This article introduces keyphrase extraction, provides a well-structured review of the existing work, offers interesting insights on the different evaluation approaches, highlights open issues and presents a comparative experimental study of popular unsupervised techniques on five datasets.

KEYWORDS
Keyphrase extraction, review, survey, unsupervised keyphrase extraction, supervised keyphrase extraction, evaluation, empirical comparison

1 | INTRODUCTION

Keyphrase extraction is concerned with automatically extracting a set of representative phrases from a document that concisely summarize its content (Hasan and Ng 2014). There exist both supervised and unsupervised keyphrase extraction methods. Unsupervised methods are popular because they are *domain independent* and do not need *labeled training data*, i.e. manual extraction of the keyphrases, which comes with subjectivity issues as well as significant investment in time and money. Supervised methods on the other hand, have more powerful modeling capabilities and

Abbreviations: -

1

Use of sentence embeddings to represent both the candidates & the doc in the same high-dim vector space

Doc2Vec

OR

Sent2Vec

document embedding

phrases' embeddings

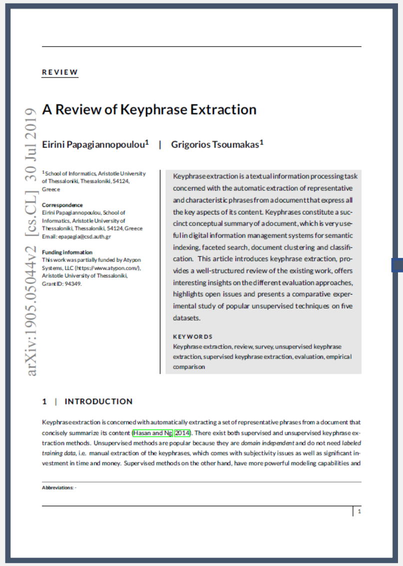
candidate phrases selection

Noun phrases following (adjective)*(noun)+

Scoring/ranking of the candidates based on their cosine similarity with the document embedding

EMBEDDINGS-BASED METHODS: RVA

LOCAL WORD EMBEDDINGS (GLOVE)


REVIEW

A Review of Keyphrase Extraction

Eirini Papagiannopoulou[1] | Grigorios Tsoumakas[1]

[1]School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece

Correspondence
Eirini Papagiannopoulou, School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece
Email: epapagia@csd.auth.gr

Funding information
This work was partially funded by Atypon Systems, LLC (https://www.atypon.com/), Aristotle University of Thessaloniki, Grant ID: 94349.

Keyphrase extraction is a textual information processing task concerned with the automatic extraction of representative and characteristic phrases from a document that express all the key aspects of its content. Keyphrases constitute a succinct conceptual summary of a document, which is very useful in digital information management systems for semantic indexing, faceted search, document clustering and classification. This article introduces keyphrase extraction, provides a well-structured review of the existing work, offers interesting insights on the different evaluation approaches, highlights open issues and presents a comparative experimental study of popular unsupervised techniques on five datasets.

KEYWORDS
Keyphrase extraction, review, survey, unsupervised keyphrase extraction, supervised keyphrase extraction, evaluation, empirical comparison

1 | INTRODUCTION

Keyphrase extraction is concerned with automatically extracting a set of representative phrases from a document that concisely summarize its content (Hasan and Ng, 2014). There exist both supervised and unsupervised keyphrase extraction methods. Unsupervised methods are popular because they are *domain independent* and do not need *labeled training data*, i.e. manual extraction of the keyphrases, which comes with subjectivity issues as well as significant investment in time and money. Supervised methods on the other hand, have more powerful modeling capabilities and

Abbreviations: -

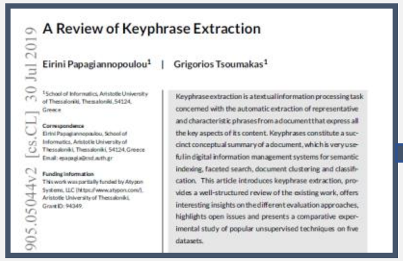
1

arXiv:1905.05044v2 [cs.CL] 30 Jul 2019

Local word vectors' generation

Local GloVe training & graph-based approaches are two alternative views of the same information source (words' co-occurrences in the text).

Embeddings trained on the target document


A Review of Keyphrase Extraction

Eirini Papagiannopoulou[1] | Grigorios Tsoumakas[1]

[1]School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece

Correspondence
Eirini Papagiannopoulou, School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece
Email: epapagia@csd.auth.gr

Funding information
This work was partially funded by Atypon Systems, LLC (https://www.atypon.com/), Aristotle University of Thessaloniki, Grant ID: 94349.

Keyphrase extraction is a textual information processing task concerned with the automatic extraction of representative and characteristic phrases from a document that express all the key aspects of its content. Keyphrases constitute a succinct conceptual summary of a document, which is very useful in digital information management systems for semantic indexing, faceted search, document clustering and classification. This article introduces keyphrase extraction, provides a well-structured review of the existing work, offers interesting insights on the different evaluation approaches, highlights open issues and presents a comparative experimental study of popular unsupervised techniques on five datasets.

Map title's/abstract's unigrams to word vectors

Document (average) vector computation

Unigrams scoring based on their cosine similarity with the average vector

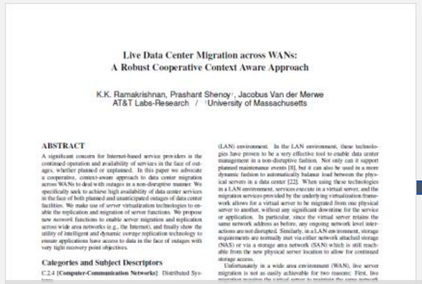
Candidate keyphrases' production

only from title & abstract

Phrase scoring by summing the words' scores

EMBEDDINGS-BASED METHODS: LV (1/2)

KEYWORDS LIE FAR FROM THE MAIN BULK OF WORDS IN LOCAL VECTOR SPACE


Live Data Center Migration across WANs:
A Robust Cooperative Context Aware Approach

K.K. Ramakrishnan, Prashant Shenoy, Jacobus Van der Merwe
AT&T Labs-Research / University of Massachusetts

ABSTRACT

Categories and Subject Descriptors

Local word vectors' generation

Assumption: the distribution's center closer to the non-keywords, since the main bulk of words are neutral or slightly relevant to the documents' topics

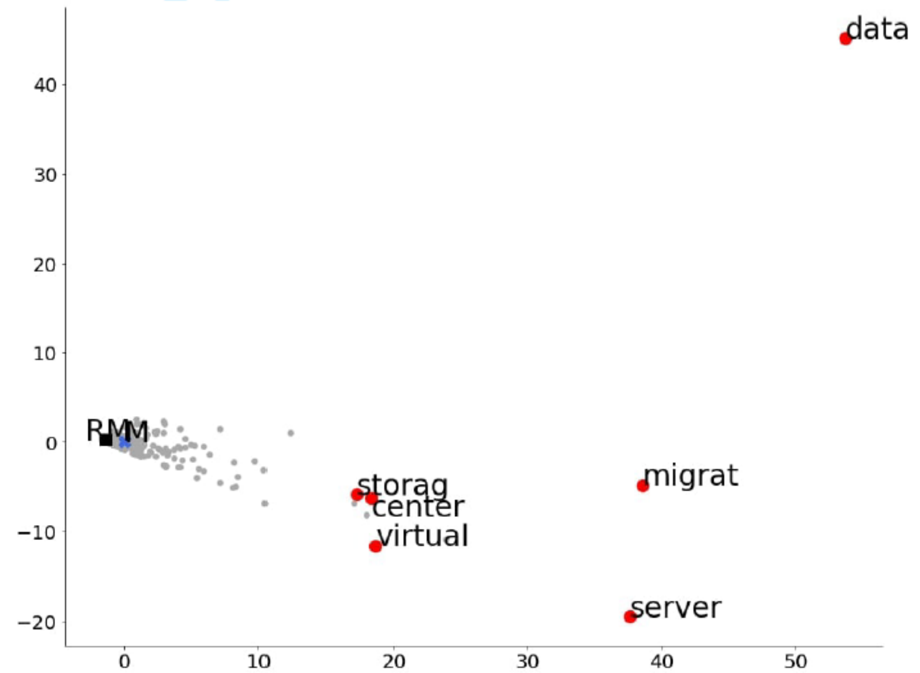
Title: Live data center migration across WANs: a robust cooperative context aware approach

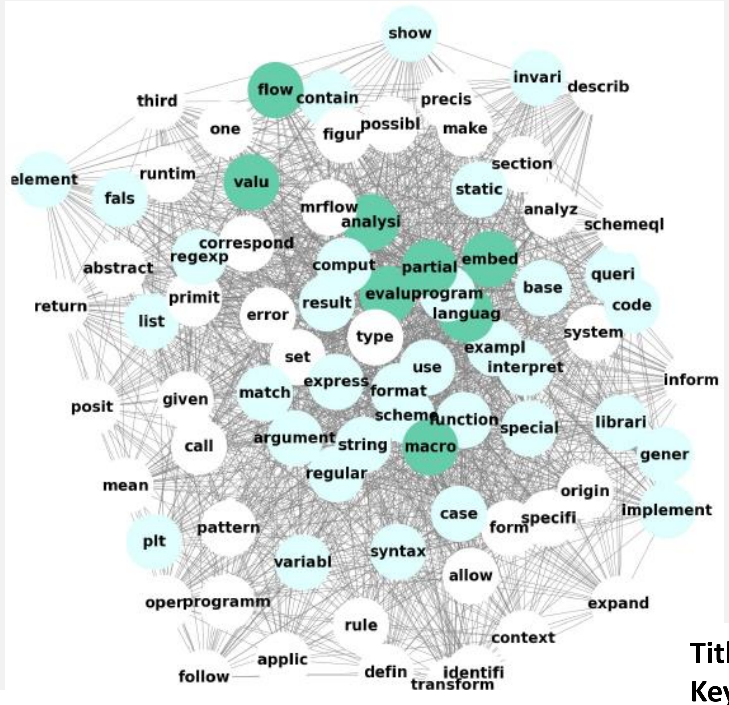
Keywords: data, center, migration, virtual, server, storage

Output:
data
migrat
live
center
wan
servic
server
approach
virtual
context

1. Text pre-processing:
   - removal of stopwords, common adjectives, reporting verbs, determiners & functional words
   - stemming
2. Generation of local vector representations (GloVe or term-term matrix)
3. Estimation of the distribution's center: mean vector $\bar{m}$ from the document's local word vectors
4. Words' scoring function: $S(w) = d(\bar{m}, \bar{v}) \cdot \frac{1}{z}$

   z: index of the 1st sentence where w occurs





EMBEDDINGS-BASED METHODS: LV (2/2)

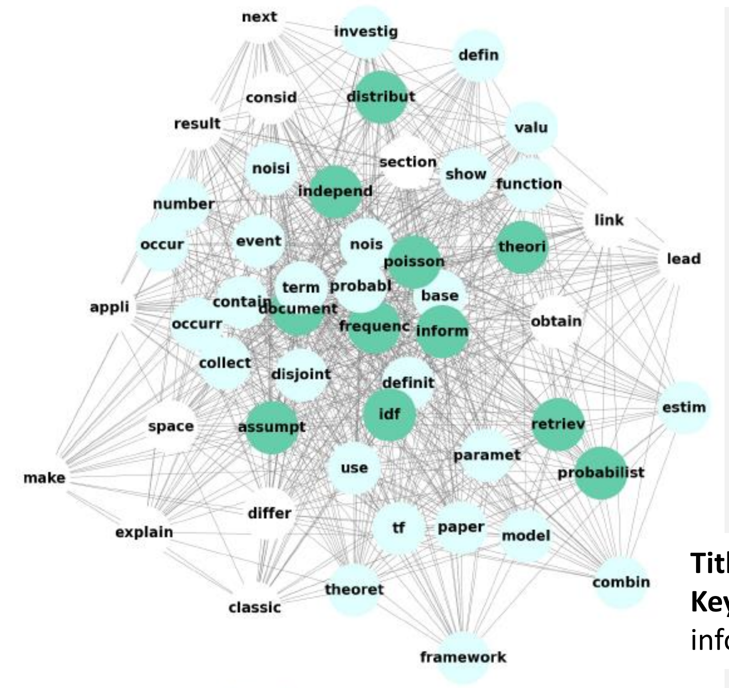
RELATION TO THE GRAPH-BASED APPROACHES

k-Core of *G*=(*V*, *E*)

- ✓ the maximal subgraph that contains vertices of degree *k* or more, where *V* is the set of vertices and *E* the set of edges
- ✓ comprises usually a minority of "qualified" representatives for the whole graph (i.e., candidate keywords)

Title: Improving the static analysis of embedded languages via partial evaluation
Keywords: partial, evaluation, macros, value flow analysis, embedded languages

Dataset	$\|k-Core\|$	$\|V\|$	$\frac{\|k-Core\|}{\|V\|}$
Semeval	70	645	0.110
Krapivin	74	736	0.103
NUS	72	778	0.095



Title: A frequency-based and a Poisson-based definition of the probability of being informative
Keywords: inverse document frequency (idf), independence assumption, probabilistic information retrieval, poisson distribution, information theory

ECAI 2020 Santiago de Compostela

TIMELINE

Year	Methods	Stat.	Stats into Graph	Clustering	LDA	C/N info	Sem.	Lang. Model.
2004	TextRank		×					
2008	SingleRank		×					
	ExpandRank		×			×		
2009	KP-Miner	×						
2013	TopicRank		×	×				
2014	CiteTextRank		×			×		
2015	Single TPR		×		×			
	Wang et al. (2015)		×				×	
2017	PositionRank		×					
2018	YAKE	×						
	EmbedRank						×	
	RVA	×					×	
	Key2Vec		×				×	
2020	LV	×					×	

PERFORMANCE OF UNSUPERVISED KE METHODS

F1	Semeval		NUS		Krapivin	
	@10	@20	@10	@20	@10	@20
TfIdf	0.154	<u>0.176</u>	<u>0.201</u>	<u>0.205</u>	<u>0.126</u>	<u>0.113</u>
KP-Miner	**0.208**	**0.219**	**0.259**	**0.243**	**0.190**	**0.161**
YAKE	<u>0.160</u>	0.169	0.188	0.180	0.124	0.109
SingleRank	0.036	0.053	0.044	0.063	0.026	0.036
TopicRank	0.134	0.142	0.126	0.118	0.099	0.086
PositionRank	0.131	0.127	0.146	0.128	0.102	0.085
RVA	0.096	0.125	0.096	0.115	0.093	0.099

OUTLINE

- Unsupervised methods
- Supervised methods
- Data/Software/Evaluation
- Practical part

SUPERVISED METHODS

- Supervised
  - Classification
  - Sequence Labelling
  - Generation

CLASSIFICATION

KEA

Pre-processing

- Whitespace tokenization
- Splitting of hyphenated words
- Punctuation marks, brackets and numbers replaced by phrase boundaries
- Removal of apostrophes and tokens not containing letters

Selection of candidate phrases

- 1/2/3-grams
- Filter proper names
- Filter starting/ending with stopword

Learning algorithm: Naive Bayes

Features

- Tfidf score
- Normalized position of 1st appearance

Inference

- Ranking by probability, using Tfidf score for breaking ties
- Remove any phrase that is a sub-phrase of a higher-ranking phrase

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. KEA: practical automatic keyphrase extraction. In Proceedings of the fourth ACM conference on Digital libraries (DL '99). Association for Computing Machinery, New York, NY, USA, 254–255.

MAUI

Pre-processing

- As in KEA

Selection of candidate phrases

- As in KEA

Learning algorithm

- Naive Bayes
- <u>Bagged Decision Trees</u>

Standard features

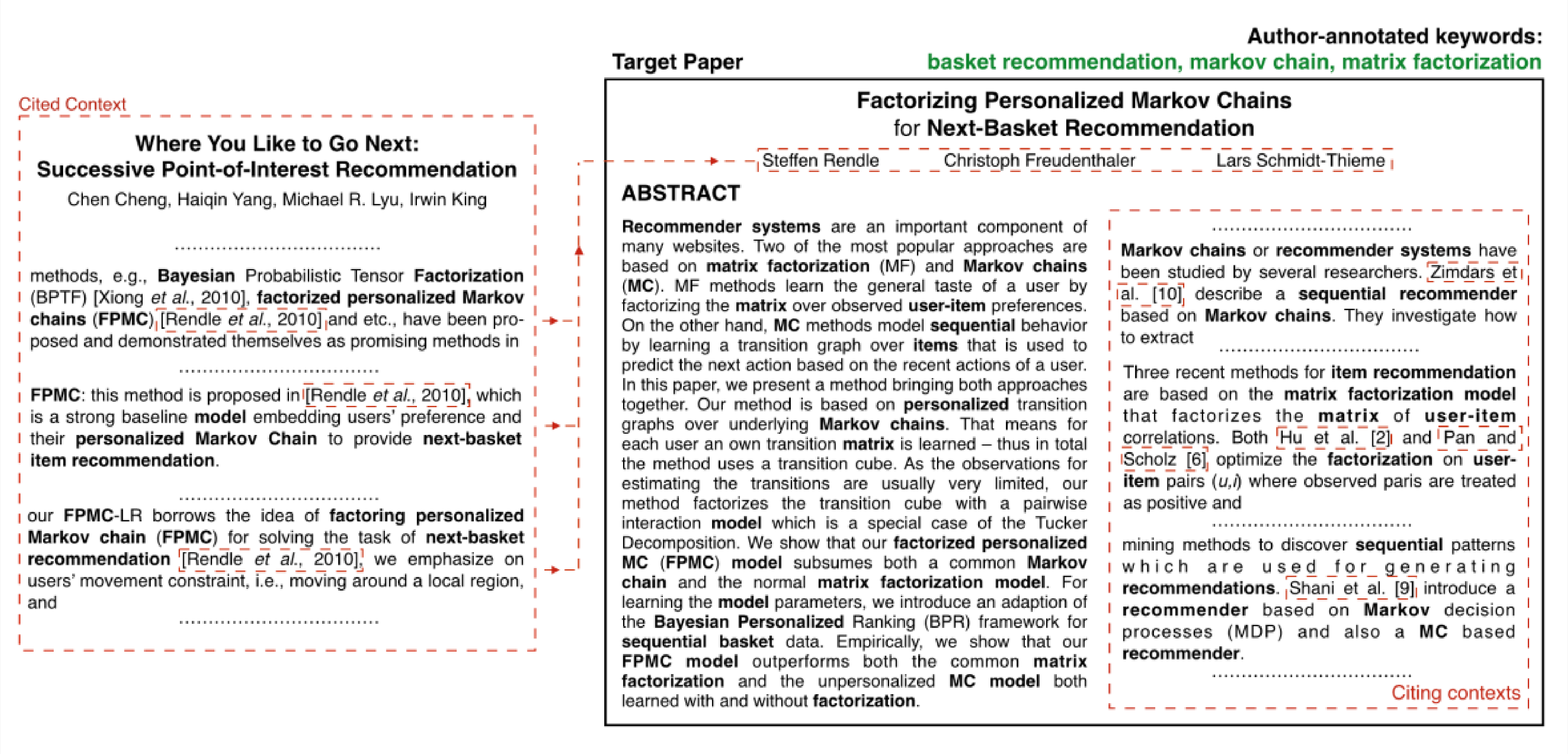
- As in KEA, phrase length
- Keyphraseness: frequency as golden keyphrase in the training corpus
- Spread: normalized distance between last and first occurrence

Wikipedia features

- Wikipedia keyphraseness: number of times it appears as link, divided by number of all pages containing it
- Node degree: number of links in Wikipedia page to other candidate keyphrases' Wikipedia pages
- Semantic relatedness: similarity of Wikipedia page to other candidate keyphrases' Wikipedia pages
- inverse Wikipedia linkage: number of incoming links to the Wikipedia page of the phrase divided by total number of links in Wikipedia

Medelyan, O., Frank, E., & Witten, I. H. (2009). Human-competitive tagging using automatic keyphrase extraction. In ACL and AFNLP. Retrieved from https://www.aclweb.org/anthology/D09-1137

CITATION-ENHANCED KEYPHRASE EXTRACTION (CeKE)



Caragea, C., Bulgarov, F., Godea, A., & Gollapalli, S. Das. (2014). Citation-enhanced keyphrase extraction from research papers: A supervised approach. EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 1435–1446. https://doi.org/10.3115/v1/d14-1150

CITATION-ENHANCED KEYPHRASE EXTRACTION (CeKE)

Selection of candidate phrases

- 1/2/3-grams
- POS: Keep only nouns and adjectives
- Stemming
- Delete phrases ending in adjectives

Standard features

- Tfidf
- Normalized position of 1st appearance
- POS of phrase

Learning algorithm

- Naive Bayes with a 0.985 threshold

Citation network features

- inCited: phrase within cited contexts
- inCiting: phrase within citing contexts
- Citation tfidf: tfidf of phrase computed based on citation contexts

Extensions to standard features

- Absolute position of 1st appearance
- Tfidf larger than a threshold
- Absolute position of 1st appearance below some value

Results

- Context = 50 tokens on each side of a citation mention
- Cited+Citing > Citing > Cited

Caragea, C., Bulgarov, F., Godea, A., & Gollapalli, S. Das. (2014). Citation-enhanced keyphrase extraction from research papers: A supervised approach. EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 1435–1446. https://doi.org/10.3115/v1/d14-1150

SEQUENCE LABELLING

SEQUENCE LABELING WITH A CRF

Sentence	Keywords	extraction	for	social	snippets
Labels	I	I	O	I	I

Basic features

- lowercased tokens
- allPunct, isCapital, isStopWord
- Parse-tree features (POS tag, phrase tag)

isInTitle feature

Unsupervised features

- In top-10 of TFIDF / TextRank / SingleRank / ExpandRank

Feature templates at position i

Unigram features	F_i, F_{i-1}, F_{i+1}
Bigram features	$F_{i-1}F_i$ and F_iF_{i+1}
Skipgram features	$F_{i-1}F_{i+1}$
Compound features	F_iG_i

E.g. for term "social" above

- BIG1-JJ_NNS, BIG-1-for_social
- SKIP-for-snippets, SKIP-PP-NP
- CMPD-JJ-NP

Gollapalli, S. Das, & Li, X. (2016). Keyphrase Extraction using Sequential Labeling. Retrieved from http://arxiv.org/abs/1608.00329

SEQUENCE LABELING WITH A BI-LSTM-CRF

Implementation details

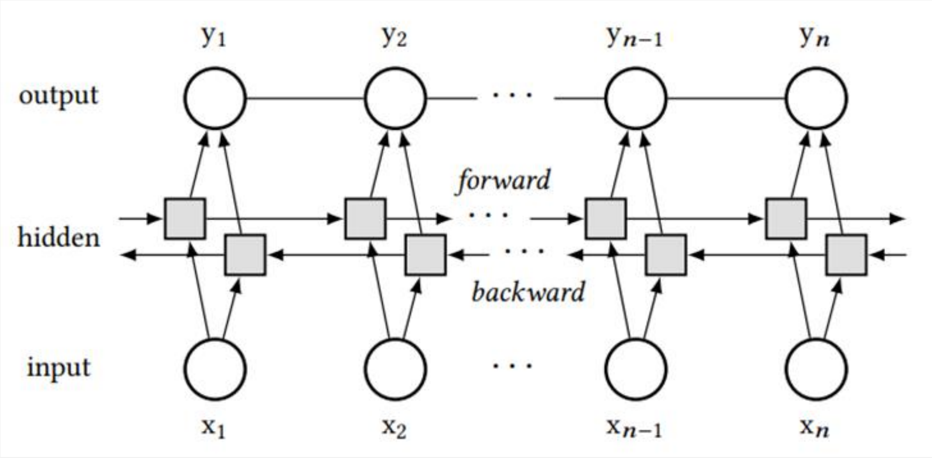
- Single 100-dimension hidden layer
- Word embeddings initialized with 100-dimension Glove pre-trained embeddings
- Dropout

Training

- Kp527k, a large dataset of 527k scientific documents with keyphrases

Results

- Bi-LSTM-CRF > CRF >> Bi-LSTM / LSTM
- Input level: document > sentence

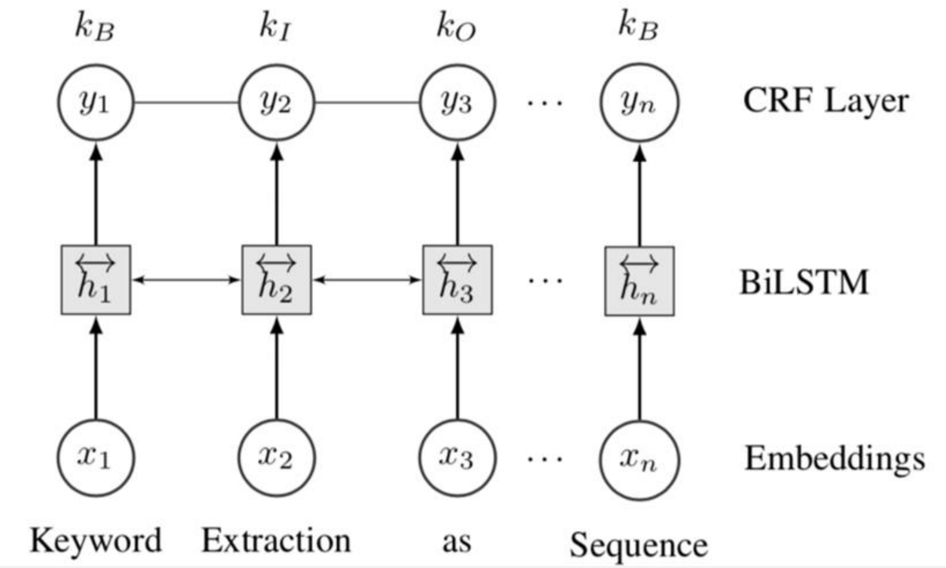


Alzaidy, R., Caragea, C., & Giles, C. L. (2019). Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. The World Wide Web Conference on - WWW '19, 2551–2557.

USING CONTEXTUAL EMBEDDINGS

Sentence	Keywords	extraction	for	social	snippets
Labels	B	I	O	B	I

	Inspec	SE-2010	SE-2017
SciBERT	0.593	**0.357**	0.521
BERT	0.591	0.330	**0.522**
ELMo	0.568	0.225	0.504
Transformer-XL	0.521	0.222	0.445
OpenAI-GPT	0.523	0.235	0.439
OpenAI-GPT2	0.531	0.240	0.439
RoBERTa	**0.595**	0.278	0.508
Glove	0.457	0.111	0.345
FastText	0.524	0.225	0.426
Word2Vec	0.473	0.208	0.292



Results considering only extractive keyphrases

Sahrawat, D., Mahata, D., Zhang, H., Kulkarni, M., Sharma, A., Gosangi, R., … Zimmermann, R. (2020). Keyphrase extraction as sequence labeling using contextualized embeddings. *Proc. 42nd European Conference on IR Research (ECIR 2020)*. https://doi.org/10.1007/978-3-030-45442-5_41

GENERATION

CopyRNN

Encoder

- Bidirectional GRU
- $x = (x_1, x_2, \dots, x_T)$
- $h = (h_1, h_2, \dots, h_T)$
- $h_t = f(x_t, h_{t-1})$

Decoder

- Forward GRU
- $y = (y_1, y_2, \dots, y_{T'})$
- $s_t = f(y_{t-1}, s_{t-1}, c_t)$
- $c_t = \sum_{j=1}^{T} a_{tj} h_j$
- $a_{\mathrm{tj}} = \frac{\exp\left(a(s_{t-1}, h_j)\right)}{\sum_{k=1}^{T} \exp\left(a(s_{t-1}, h_k)\right)}$
- $p_{gen}(y_t | y_{1\dots t-1}, x) = g(y_{t-1}, s_t, c_t)$

Encoder-Decoder model

- For each source text construct as many training samples (x,y) as its keyphrases
- Phrases generated via beam search and a max heap to maintain the ones with the highest probability

Copying mechanism

- Limited vocabulary in RNNs
- $p(y_t | y_{1\dots t-1}, x) = p_{gen}(y_t | y_{1\dots t-1}, x) + p_{\mathrm{copy}}(y_t | y_{1\dots t-1}, x)$
- $p_{\mathrm{copy}}(y_t | y_{1\dots t-1}, x) = \frac{1}{Z} \sum_{j: x_j = y_t} \exp\left(\sigma\left(h_j^T W\right)[y_{t-1}; s_t; c_t]\right)$

Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., & Chi, Y. (2017). Deep keyphrase generation. ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), 1, 582–592. https://doi.org/10.18653/v1/P17-1054

CorrRNN

Builds upon CopyRNN

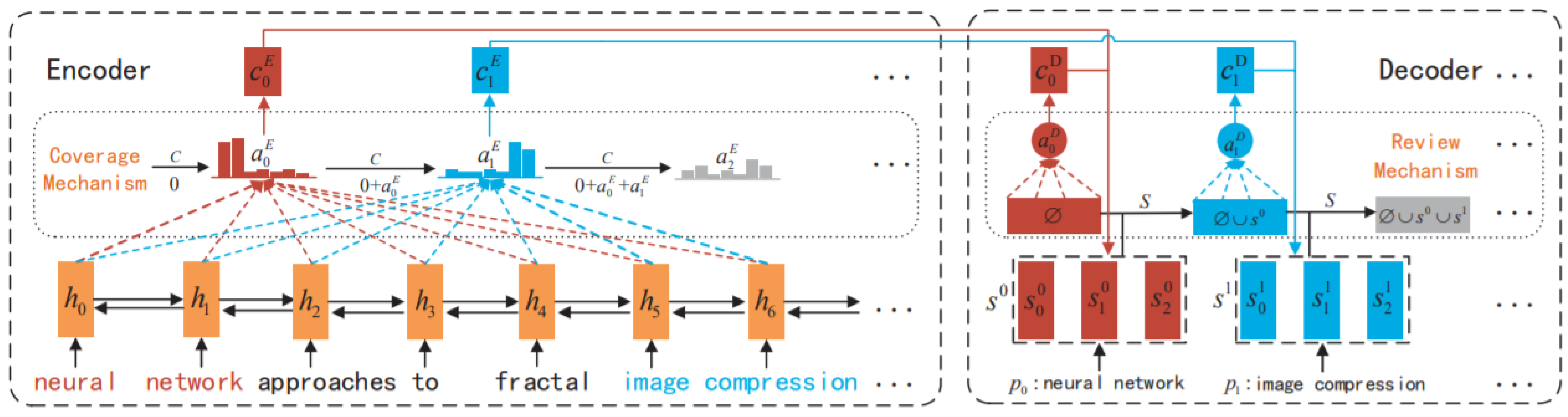
- Ignoring keyphrase correlations leads to duplication and coverage issues
- Duplication: multiple keyphrases expressing the same meaning
- Coverage: missed keyphrases

Coverage mechanism

- Context vector takes into account the sum of all past attention distributions

Review mechanism

- Decoder attention in hidden states of previous keyphrases is introduced

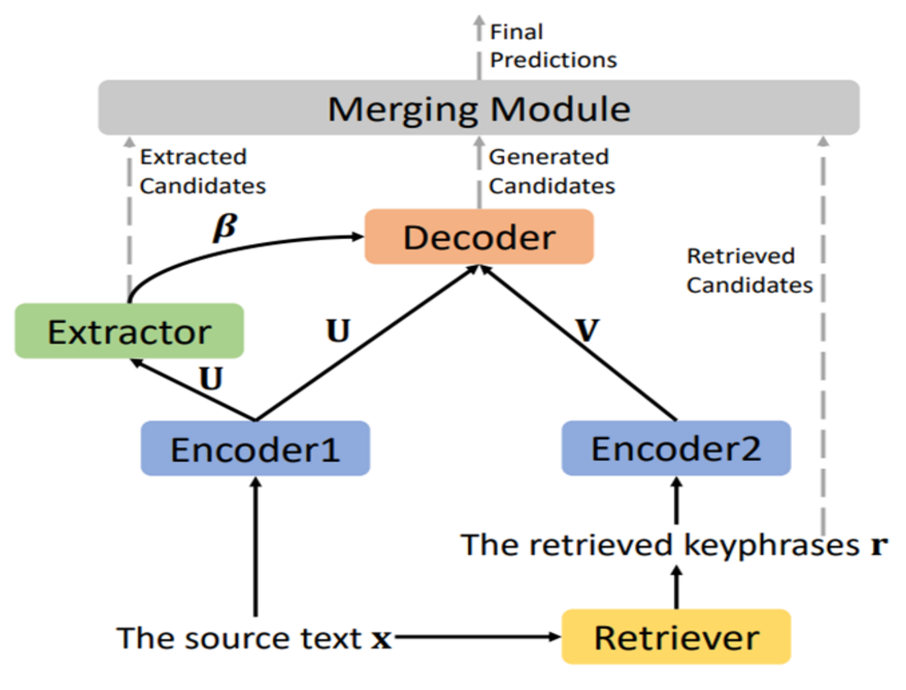


Chen, J., Zhang, X., Wu, Y., Yan, Z., & Li, Z. (2018). Keyphrase generation with correlation constraints. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 4057–4066. https://doi.org/10.18653/v1/d18-1439

ECAI 2020 Santiago de Compostela

GENERATION WITH RETRIEVAL AND EXTRACTION

Source text $x = (x_1, x_2, \dots, x_{T_x})$



Encoder 1

- BiGRU
- $u = (u_1, u_2, \dots, u_{T_x})$
- $u_i = f(x_i, \vec{u}_{i-1}, \overleftarrow{u}_{i+1})$

Extractor

- Classification layer outputting sequence of importances $\beta = (\beta_1, \beta_2, \dots, \beta_{T_x})$
- $p(\beta_j = 1 | u_j, s_j, d) = \mathrm{sigmoid}(\mathrm{W_c u_j} + \mathrm{u_j^T W_s d} - \mathrm{u_j^T W_n} \tanh(\mathrm{s_j}) + \mathrm{b})$
- $s_j = \sum_{i=1}^{j-1} u_i \beta_i$ current summary representation
- $d = \tanh(W_d[\vec{u}_{T_x}; \overleftarrow{u}_1] + b)$ global document representation

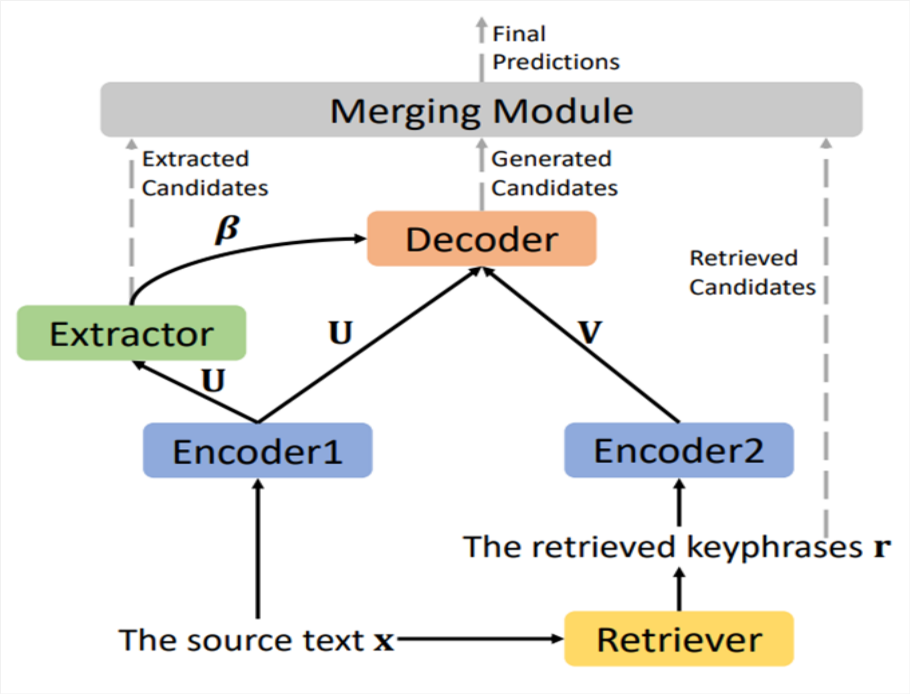
Chen, W., Chan, H. P., Li, P., Bing, L., & King, I. (2019). An integrated approach for keyphrase generation via exploring the power of retrieval and extraction. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2846–2856.

GENERATION WITH RETRIEVAL AND EXTRACTION

Source text $x = (x_1, x_2, \dots, x_{T_x})$

Retriever

- Find the KNNs of the source text
- Take the keyphrases of these texts
- Concatenate them with a separator into a sequence $r = (r_1, r_2, \dots, r_{T_r})$



Encoder 2

- BiGRU
- $v = (v_1, v_2, \dots, v_{T_r})$
- $v_i = f(x_i, \vec{v}_{i-1}, \overleftarrow{v}_{i+1})$

Decoder

- As in CopyRNN with attention and copying mechanisms
- Source text attention scores are rescaled by the extractor scores

Chen, W., Chan, H. P., Li, P., Bing, L., & King, I. (2019). An integrated approach for keyphrase generation via exploring the power of retrieval and extraction. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2846–2856.

GENERATION WITH RETRIEVAL AND EXTRACTION

Extracted candidates

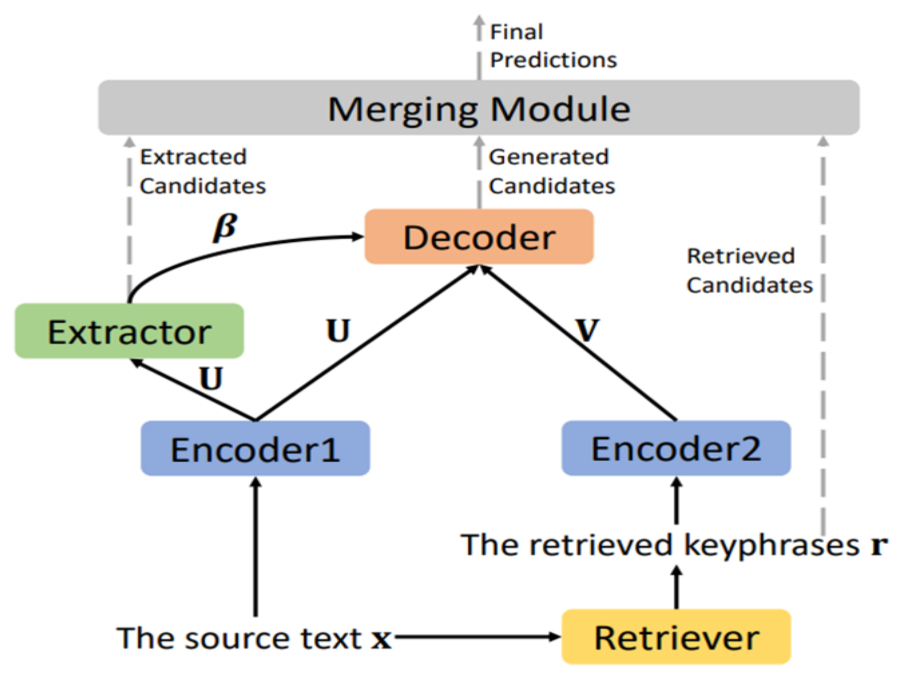
- Extract word if β_j > threshold
- Merge adjacent words into phrases
- Score phrases by the mean of the importance scores of their words

Generated candidates

- Score them with their beam search score

Retrieved candidates

- Find the kNNs of the source text
- Take the keyphrases of these texts
- Score them by Jaccard similarity between source text and their text
- Duplicates with lower score are removed



Merge scores

- Score candidates using a popular natural language inference (NLI) model
- Rank candidates by normalized NLI-weighted sum of scores

Chen, W., Chan, H. P., Li, P., Bing, L., & King, I. (2019). An integrated approach for keyphrase generation via exploring the power of retrieval and extraction. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2846–2856.

TYPES OF FEATURES

Category	Feature
Statistical	Tf/Idf/TfIdf
	Number of sentences containing the phrase
	Words or phrase entropy
	Correlations between features and the phrase
	Topic distributions (LDA)
Positional	Appearance in specific parts of the fulltext
	Position of the (1st or last) occurrence
	Distance between phrase and citation
	Section occurrence vector
	Sentence boundaries
	Spread
Context	Previous/next token of the phrase
	POS/syntactic features of previous/next token of phrase
	Relative position of the phrase in given text
	Learning embeddings/features
	Bigram, skipgram, compound features
Linguistic	Stemmed unigram
	Boolean features: IsCapilazed, IsStopword
	POS tags, NP-chunking
	Phrase length
	Suffix sequence
	Acronym status
Stacking	Unsupervised methods output
	Supervised methods output
External knowledge	Existence of the phrase in ontologies or as a Wikipedia link
	Wikipedia based Idf/phraseness
	(Pretrained)Word embedding of the phrase
	Supervised keyphraseness
	Bias based on previous research
	TitleOverlap
	Semantic feature weight (returned by HITS with Wikipedia Info)

TIMELINE

Year	Method	ML Algorithm	Stat.	Posit.	Ling.	Cont.	Stack.	Ext.
1999	KEA	Naive Bayes	×	×				
2009	MAUI	Bagged Decision Trees	×	×	×			×
	Ranking SVM	SVM	×	×	×			
2014	CeKE	Naive Bayes	×	×	×	×		
2016	TopicCoRank	Graph-based Method	×					×
2017	PCU-ICL	Ensemble (RF/SVM)	×	×	×	×	×	×
	MIKE	Random-walk Parametric Model	×	×	×	×		
	Gollapalli et al.	CRFs	×	×	×	×	×	×
	CopyRNN	seq2seq Learning				×		
2018	CorrRNN	seq2seq Learning				×		
	Ye & Wang	Multi-task Learning (seq2seq Model)				×		
2019	Chen et al.	Multi-task Learning (multiple neural)				×		

PERFORMANCE OF SUPERVISED KEYPHRASE EXTRACTION METHODS

Model	Inspec		Krapivin		NUS		SemEval		KP20k	
	F_1@5	F_1@10	F_1@5	F_1@10	F_1@5	F_1@10	F_1@5	F_1@10	F_1@5	F_1@10
TF-IDF	0.188	0.269	0.092	0.120	0.103	0.142	0.076	0.135	0.087	0.113
TextRank	0.194	0.244	0.142	0.128	0.147	0.153	0.107	0.130	0.151	0.132
Maui	0.037	0.032	0.196	0.181	0.205	0.234	0.032	0.036	0.223	0.204
CorrRNN*	0.229_7	0.248_9	0.255_2	0.238_4	0.273_5	0.265_4	0.197_3	0.221_5	0.291_2	0.264_2
CopyRNN*	0.251_7	0.279_3	0.268_4	0.243_1	0.275_2	0.268_2	0.190_6	0.214_5	0.306_1	0.273_0
KG-KE	0.254_4	0.281_2	0.265_3	0.240_1	0.278_4	0.273_1	**0.207_4**	**0.227_7**	0.307_0	0.274_0
KG-KR	0.244_2	0.275_1	0.266_5	0.247_1	0.278_2	0.276_2	0.189_7	0.215_7	0.311_1	0.278_0
KG-KE-KR	0.245_1	0.278_4	0.267_3	0.246_2	0.285_9	0.279_4	0.194_4	0.220_2	0.314_0	0.280_0
KG-KE-KR-M	**0.257_2**	**0.284_3**	**0.272_3**	**0.250_2**	**0.289_4**	**0.286_4**	0.202_6	0.223_3	**0.317_0**	**0.282_0**

Model	NUS		SemEval		Krapivin	
	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10
Tf-idf	0.136	0.184	0.128	0.194	0.129	0.160
TextRank	0.195	0.196	0.176	0.187	0.189	0.162
SingleRank	0.140	0.173	0.135	0.176	0.189	0.162
ExpandRank	0.132	0.164	0.139	0.170	0.081	0.126
TopicRank	0.115	0.123	0.083	0.099	0.117	0.112
Maui	0.249	0.268	0.044	0.039	0.249	0.216
KEA	0.069	0.084	0.025	0.026	0.110	0.152
RNN	0.169	0.127	0.157	0.124	0.135	0.088
CopyRNN	0.334	0.326	0.291	0.304	0.311	0.266
CopyRNN_F	0.323	0.289	0.270	0.270	0.293	0.222
CorrRNN_C	**0.361**	**0.335**	0.296	0.319	0.311	0.273
CorrRNN_R	0.354	0.328	0.306	0.312	0.314	0.270
CorrRNN	0.358	0.330	**0.320**	**0.320**	**0.318**	**0.278**

Method	kp20k		
	Pr%	Re%	F1%
Bi-LSTM-CRF	**64.19**	24.66	**35.63**
copyRNN @5	27.71	**41.79**	33.29
Tf-Idf @5	8.97	13.49	10.77
TextRank @5	15.29	23.01	18.37
SingleRank @5	8.42	12.70	10.14
KEA	15.14	22.78	18.19

SUBJECTIVITY

Relevant phrases that are not annotated by humans as keyphrases are considered as negative training examples

Authors select as keyphrases:

- those that promote their work in a particular way
- those that are popular, …

Readers select as keyphrases:

- terms related to their field/background knowledge
- absent synomyms or more general/narrow phrases, …

subjectivity

Unlabeled phrases are not reliable as negative examples

- Problems: affect the evaluation/learning process
- Solutions: multiple annotators/positive unlabelled learning

Sterckx, L., Caragea, C., Demeester, T., & Develder, C. (2016). Supervised keyphrase extraction as positive unlabeled learning. EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings, 1924–1929. https://doi.org/10.18653/v1/d16-1198

OUTLINE

- Unsupervised methods
- Supervised methods
- Data/Software/Evaluation
- Practical part

Type	Dataset	Created By	Docs	Language	Annotation Type
Full-text Papers	NUS	Nguyen and Kan (2007)	211	English	Authors/Readers
	Krapivin	Krapivin et al. (2008)	2304	English	Authors
	Semeval2010	Kim et al. (2010)	244	English	Authors/Readers
	Citeulike-180	Medelyan et al. (2009)	180	English	Readers
Paper Abstracts	Inspec	Hulth (2003)	2000	English	Indexers
	KDD	Gollapalli and Caragea (2014)	755	English	Authors
	KP20k	Meng et al. (2017)	567830	English	Authors
	WWW	Gollapalli and Caragea (2014)	1330	English	Authors
News	DUC-2001	Wan and Xiao (2008)	308	English	Readers
	500N-KPCrowd	Marujo et al. (2012)	500	English	Readers
	110-PT-BN-KP	Marujo et al. (2012)	110	Portuguese	Readers
	Wikinews	Bougouin et al. (2013)	100	French	Readers

More datasets on https://github.com/LIAAD/KeywordExtractor-Datasets

KE DATASETS

COMMERCIAL TEXT ANALYSIS APIS

RELATED TO THE TASK

Aylien: keyphrase extraction → English, German, French, Italian, Spanish, Portuguese

Amazon Comprehend API: keyphrase extraction → English and Spanish (otherwise conversion to English or Spanish via the Amazon Translate)

Textrazor API: entity recognition service that offers the confidence score and the relevance score of the returned entity to the source text

IBM Watson Natural Language Understanding API: keyphrase extraction

Microsoft's Text Analytics APIs: keyphrase extraction

Google Cloud Natural Language API: NOT keyphrase extraction service

ONLY entity recognition which identifies entities and labels by types, such as person, organization, location, event, product, and media

many supported languages

F1*	Semeval		NUS		Krapivin		Inspec		500N-KPCrowd	
	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20
IBM	0.100	**0.118**	0.115	0.117	**0.114**	**0.106**	**0.256**	**0.270**	0.081	0.133
GOOGLE	0.089	0.106	**0.135**	**0.141**	0.106	0.096	0.168	0.192	**0.143**	0.210
Amazon	0.037	0.062	0.035	0.063	0.034	0.054	0.063	0.109	0.058	0.093
Textrazor	0.073	0.084	0.099	0.113	0.096	0.097	0.116	0.140	0.062	0.098
Aylien	**0.101**	0.092	0.121	0.108	0.080	0.062	0.123	0.132	**0.143**	**0.229**

Performance of Commercial APIs

** This empirical study is conducted in the context of the survey on the task using very domain-specific texts from keyphrase extraction data collections. Thus, such type of evaluation of commercial general purpose APIs, whose internal working is not actually known, should not be considered as a positive or negative attitude in favor of the APIs with high performance on the datasets*

Name	Implementation Language	Languages
Maui	Java	multilingual
YAKE	Python	Multilingual
TopicCoRank	Python	English/French
RAKE	Python	Multilingual
KEA	Java (Python Wrapper)	Multilingual
PKE (supervised/unsupervised methods)	Python	Multilingual
seq2seq	Python	English
KE package (TfIdf, TextRank, SingleRank, ExpandRank)	C++	English*
TextRank	Python	Multilingual
Sequential Labeling (Maui, Kea, Ceke, crf)	Java	English*
CiteTextRank (TfIdf, TextRank, SingleRank, ExpandRank)	Java	English*

No other supported languages are explicitly mentioned

NON-COMMERCIAL SOFTWARE

EVALUATION MEASURES

$$precision = \frac{number\ of\ correctly\ matched}{total\ number\ of\ extracted} \qquad recall = \frac{number\ of\ correctly\ matched}{total\ number\ of\ assigned}$$

$$F_1 - measure = 2 \times \frac{precision \times recall}{precision + recall}$$

Ranking quality measures

$$MRR = \frac{1}{D} \sum_{d \in D} \frac{1}{rank_d}$$

$$AP = \frac{\sum_{r=1}^{|L|} P(r) \cdot rel(r)}{|L_R|} \qquad MAP = \frac{1}{n} \sum_{1}^{n} AP_i$$

Binary preference measure

$$Bpref = \frac{1}{R} \sum_{r \in R} 1 - \frac{|n\ ranked\ higher\ than\ r|}{M}$$

Average of Correctly Extracted Keyphrases

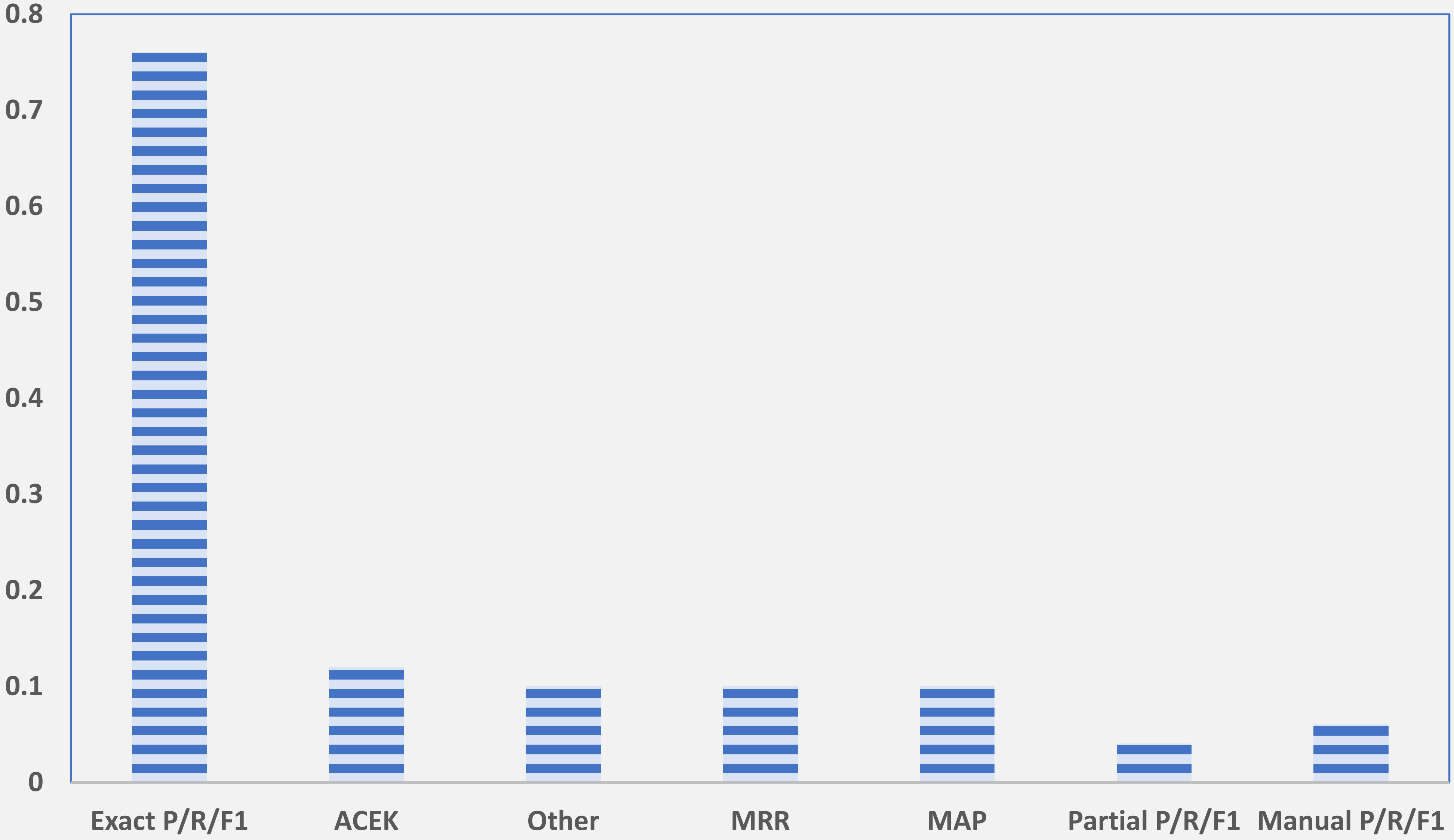
$$CEK = |extracted \cap gold| \qquad ACEK = \frac{1}{n} \sum_{1}^{n} CEK_i$$

ECAI 2020 Santiago de Compostela

EVALUATION APPROACHES

Approach	Description	Problems
exact match evaluation	the number of correctly matched phrases with the golden ones are determined based on string matching (after stemming)	• usually suboptimal evaluation e.g., gold keyphrases: “approximate search” and “similarity search” output keyphrase: “approximate similarity search”
manual evaluation	experts decide whether the returned keyphrases by a system are wrong or right	• investment of time and money • great subjectivity
partial match evaluation	Precision, Recall and F_1-measure between the set of words found in all golden keyphrases and the set of words found in all extracted keyphrases (after stemming)	• cannot evaluate the syntactic correctness of the phrases • cannot deal with over-generation problems & overlapping keyphrase candidates
machine translation/ summarization evaluation	BLEU, ROUGE, etc.	• not widely adopted by the keyphrase extraction community

POPULARITY OF EVALUATION MEASURES/ APPROACHES

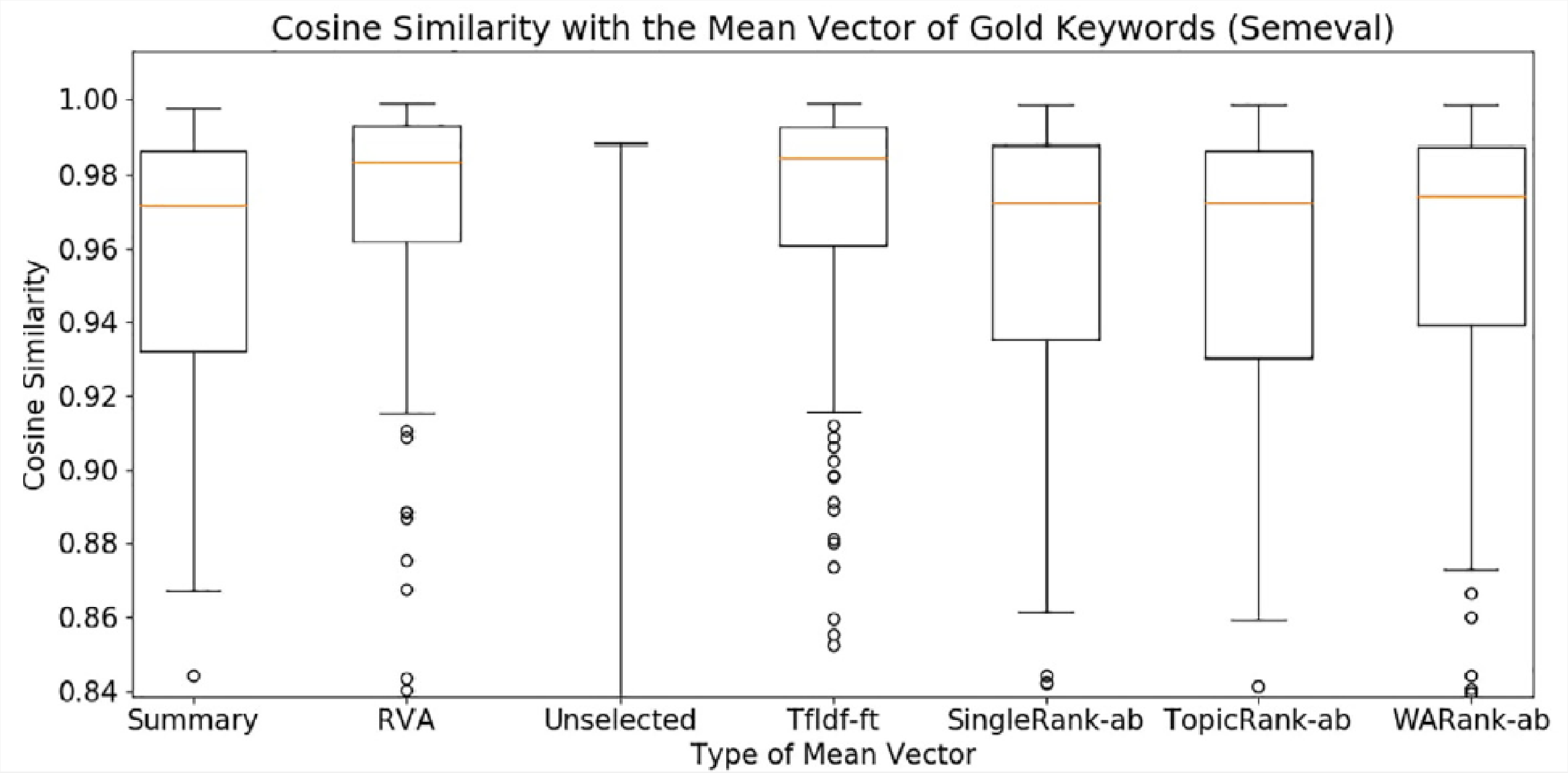


There are also libraries dedicated to evaluation, some of them in python, e.g.:

Gysel, C. V., & Rijke, M. d. (2018). Pytrec_eval: An Extremely Fast Python Interface to trec_eval. Proceedings of the 41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'18) (pp. 873 - 876). Ann Arbor, USA. July 8- 12: ACM Press.

THE NEED FOR SEMANTIC EVALUATION

"gold" keyphrases' comparison with the returned keyphrases of a system
- utilization of the word vector representation



Plot of cosine similarities between the mean word vector derived from the ground truth's keyphrases and the mean word vector of the system's phrases.

Setup:

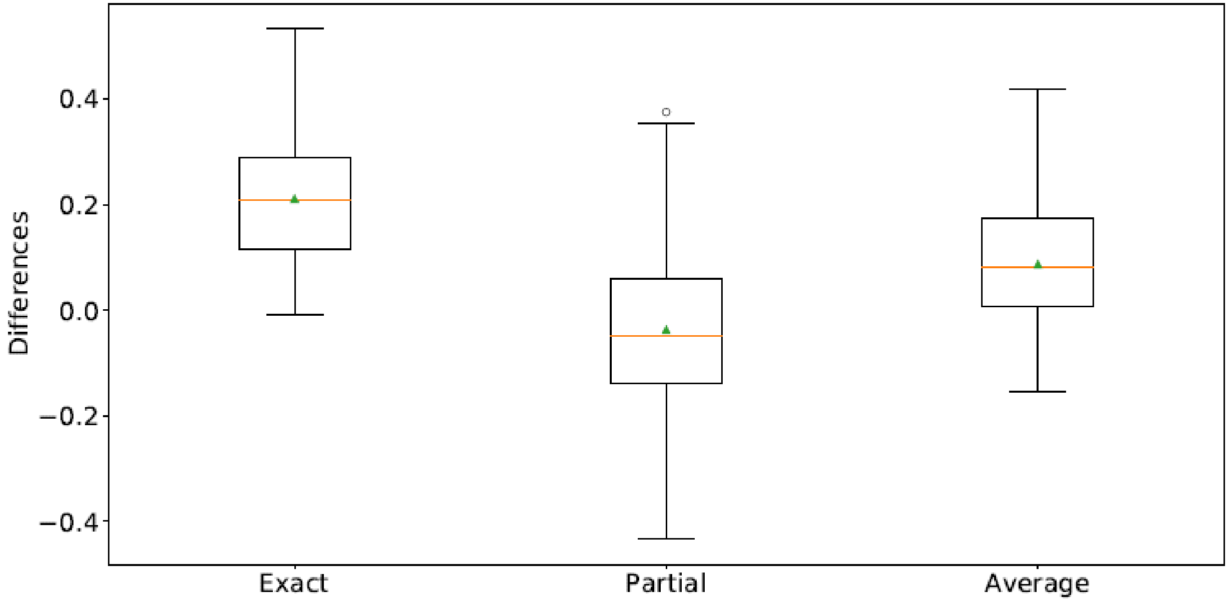
- random selection of 50 full-text articles from the Krapivin dataset
- keyphrase extraction using one statistical method (KPMiner) and one graph-based method (MultipartiteRank)
- F_1@10 calculation for each article and method based on
  - exact match evaluation
  - partial match evaluation
  - manual evaluation
- 3 statistical tools to study the relation between the exact/partial match evaluation and the manual evaluation:
  - Spearman coefficient
  - Wilcoxon signed-rank non-parametric test at a significance level of 0.05
  - Mean Squared Error (MSE)

	Spearman			**Wilcoxon**			**MSE**		
	Exact	Partial	Average	Exact	Partial	Average	Exact	Partial	Average
KPMiner	**0.637**	0.350	0.492	**~0**	0.090	**~0**	0.060	0.031	**0.026**
MultipartiteRank	**0.344**	0.203	0.266	**~0**	0.322	**~0**	0.055	0.032	**0.028**

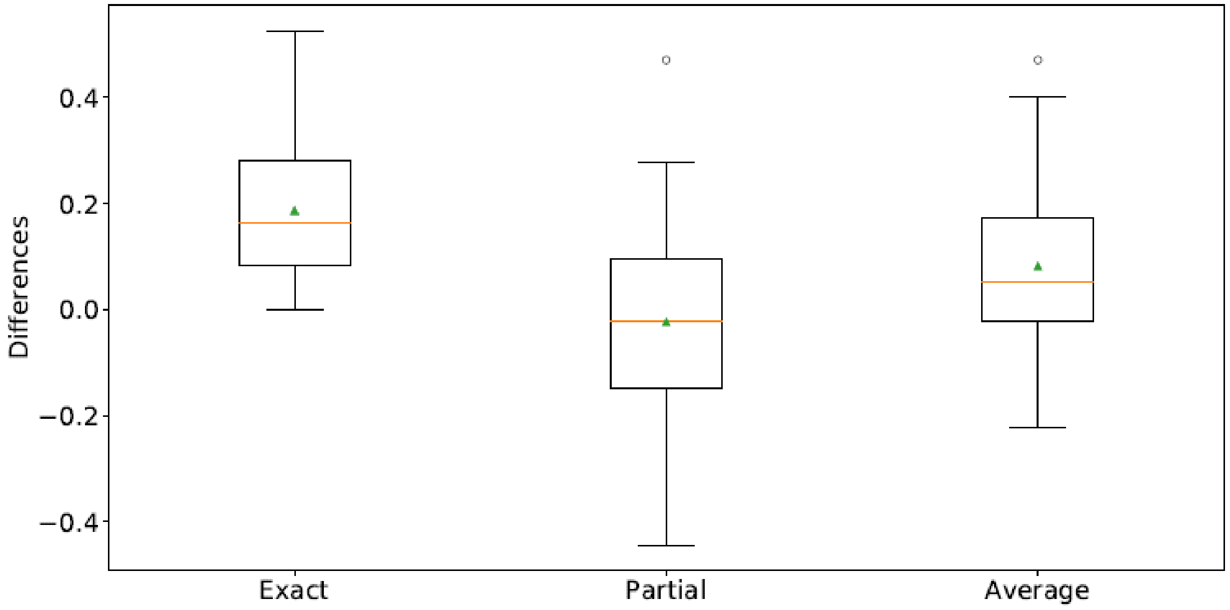
Spearman correlation coefficient, Wilcoxon signed-rank test p value, and MSE between the F_1@10 scores obtained via manual evaluation and those obtained via exact/partial matching along with their average.

EXACT VS PARTIAL MATCHING

Distribution of differences between the F_1@10 scores based on the manual evaluation and the F_1@10 based on the exact (Exact), partial (Partial) and average (Average) evaluation approaches for the 50 manually evaluated documents given on the x axis.



(a) KPM



(b) MR

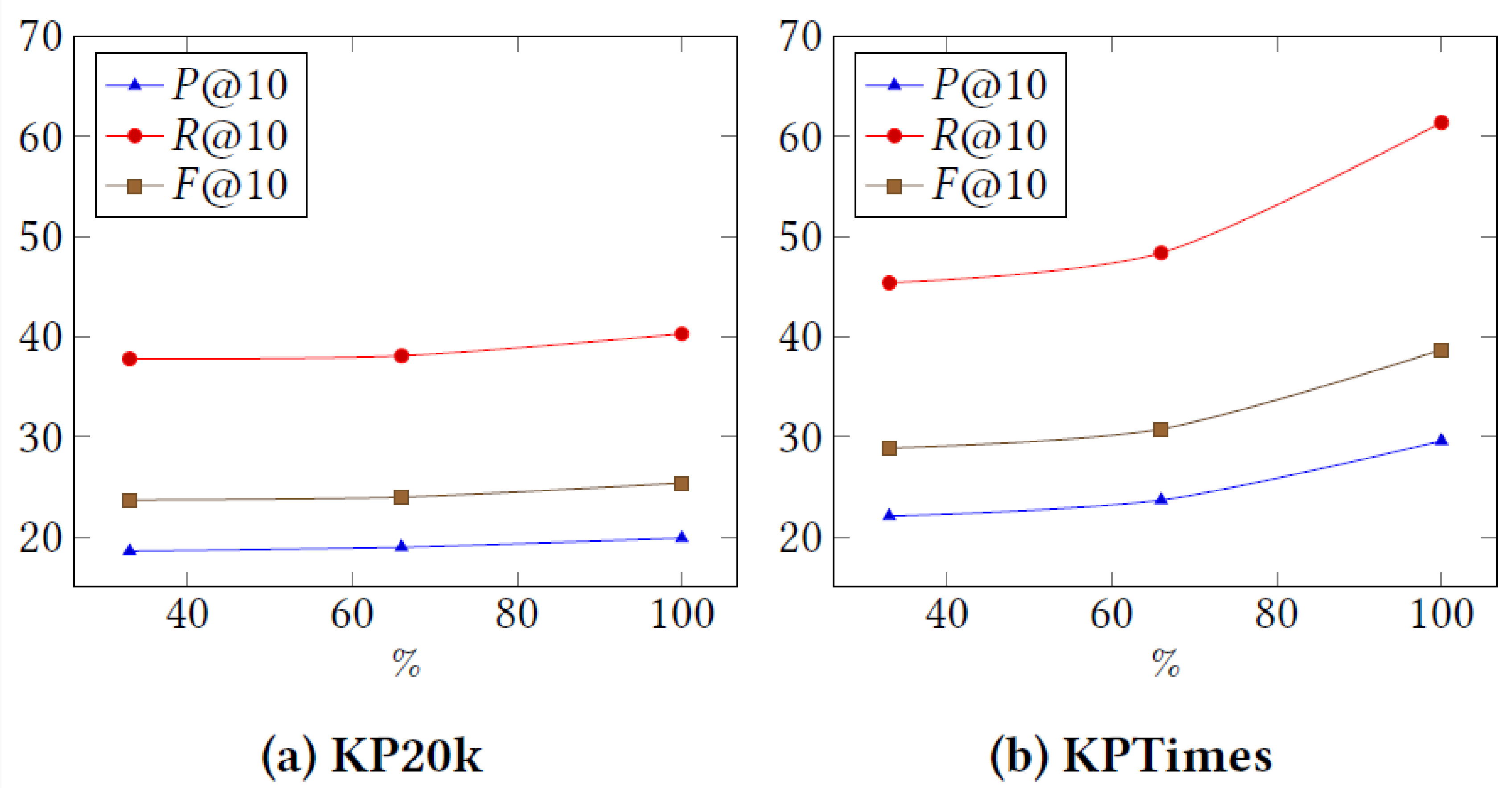
EXACT VS PARTIAL MATCHING

Our analysis suggests that researchers should consider **the average of exact and partial matching** for empirical comparison of keyphrase extraction methods.

Model	Dataset (metric)	Orig.	Ours
PositionRank	WWW (F@8)	12.3	11.7
MultipartiteRank	Semeval (F@10)	14.5	14.3
EmbedRank	Inspec (F@10)	37.1	35.6
CopyRNN	KP20k (F@10 on present)	26.2	28.2
CorrRNN	Krapivin (F@10 on present)	27.8	23.5

ORIGINAL VS RE-IMPLEMENTATION SCORES

Ygor Gallina, Florian Boudin, and Béatrice Daille. 2020. Large-Scale Evaluation of Keyphrase Extraction Models. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020 (JCDL '20). Association for Computing Machinery, New York, NY, USA, 271–278.



Performance of CopyRNN with different sizes of training data.

LEARNING CURVES

Ygor Gallina, Florian Boudin, and Béatrice Daille. 2020. Large-Scale Evaluation of Keyphrase Extraction Models. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020 (JCDL '20). Association for Computing Machinery, New York, NY, USA, 271–278.

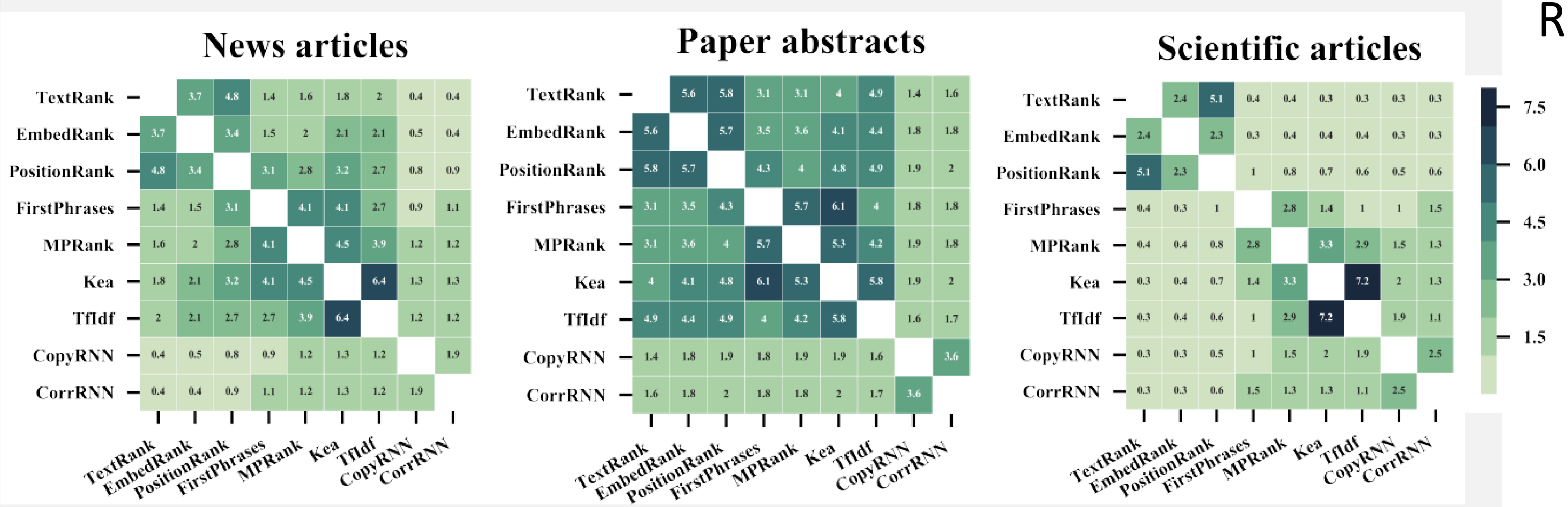
RESULTS (1/2)

F@10	Scientific articles			Paper abstracts			News articles		
	PubMed	ACM	SemEval	Inspec	WWW	KP20k	DUC-2001	KPCrowd	KPTimes
FirstPhrases	15.4	13.6	13.8	29.3	10.2	13.5	24.6	<u>17.1</u>	9.2
TextRank	1.8	2.5	3.5	<u>35.8</u>	8.4	10.2	21.5	7.1	2.7
TfIdf	16.7	12.1	17.7	**36.5**	9.3	11.6	23.3	*16.9*	9.6
PositionRank	4.9	5.7	6.8	34.2	*11.6*	*14.1*	<u>28.6</u>	13.4	8.5
EmbedRank	3.7	2.1	2.5	*35.6*	10.7	12.4	**29.5**	12.4	4.0
Kea	*18.6*	*14.2*	<u>19.5</u>	34.5	11.0	14.0	*26.5*	**17.3**	*11.0*
CopyRNN	**24.2**	**24.4**	**20.3**	28.2	**22.2**	**25.4**	10.5	8.4	**39.3**
CorrRNN	<u>20.8</u>	<u>21.1</u>	*19.4*	27.9	<u>19.9</u>	<u>21.8</u>	10.5	7.8	<u>20.5</u>

Performance of keyphrase extraction models.

Ygor Gallina, Florian Boudin, and Béatrice Daille. 2020. Large-Scale Evaluation of Keyphrase Extraction Models. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020 (JCDL '20). Association for Computing Machinery, New York, NY, USA, 271–278.

RESULTS (2/2)



Average number of keyphrases in common between model outputs.

Ygor Gallina, Florian Boudin, and Béatrice Daille. 2020. Large-Scale Evaluation of Keyphrase Extraction Models. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020 (JCDL '20). Association for Computing Machinery, New York, NY, USA, 271–278.

OUTLINE

- **Unsupervised methods**
- **Supervised methods**
- **Data/Software/Evaluation**
- **Practical part**

ECAI 2020 Santiago de Compostela

THANK YOU

Acknowledgments
Ricardo Campos was financed by the ERDF – European Regional Development Fund through the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project PTDC/CCI-COM/31857/2017 (NORTE-01-0145-FEDER-03185).

ECAI 2020 Santiago de Compostela