From Fundamentals to Recent Advances
A Tutorial on Keyphrasification

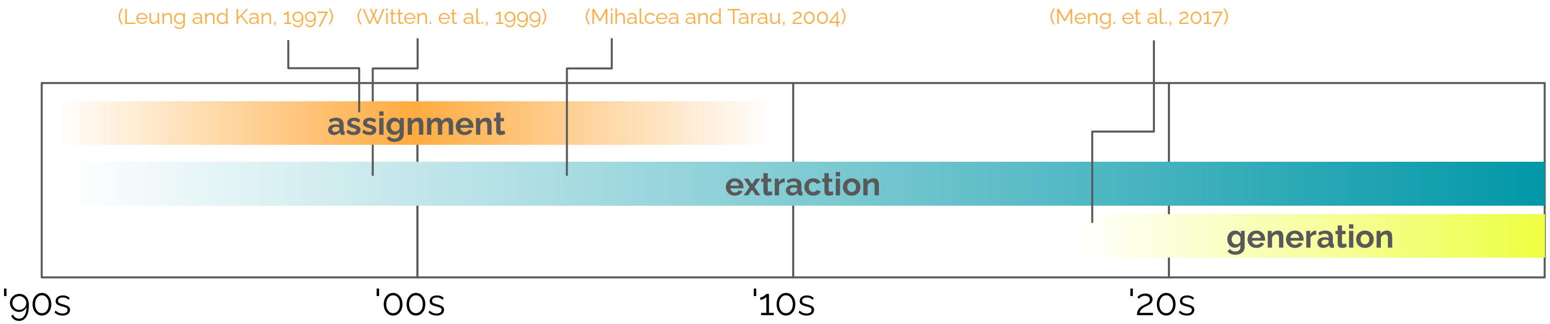
Part 1.3 Traditional Methods for Keyphrase Extraction

Rui Meng, Debanjan Mahata, Florian Boudin

ECIR 2022

An overview of history

- Keyphrases were initially introduced as a means for cataloging and indexing documents in digital libraries (Fagan, 1987)
  - Assignment methods: assign thesaurus entries (e.g. MeSH/UMLS in PubMed)
  - Extractive methods: identify important phrases from text
  - Generative methods: produce phrases that summarize the content



(Fagan, 1987) Automatic phrase indexing for document retrieval, SIGIR.
(Leung and Kan, 1997) A statistical learning approach to automatic indexing of controlled index terms. JASIS.
(Witten et al., 1999) Kea: Practical automatic keyphrase extraction, DL.
(Mihalcea and Tarau, 2004) TextRank: Bringing order into text, EMNLP.
(Meng et al., 2017) Deep keyphrase generation, ACL.

Taxonomy of Methods

Traditional methods | Neural-based methods

extraction

statistical (e.g. TF.IDF)

graph-based (e.g. TextRank)

supervised feature-based (e.g. Kea)

sequence labeling: CRF, Bi-LSTM-CRF (Das Gollapalli et al., 2017) (Alzaidy et al. 2019)

embedding-based (e.g. Key2Vec) (Mahata et al., 2018)

generation

resource-based (e.g. Kea++) (Medelyan and Witten, 2006)

sequence-to-sequence (e.g. CopyRNN)

(Medelyan and Witten, 2006) Thesaurus based automatic keyphrase indexing. JCDL.
(Das Gollapalli et al., 2017) Incorporating expert knowledge into keyphrase extraction. AAAI.
(Mahata et al., 2018) Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings. NAACL.
(Alzaidy et al. 2019) Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. WWW.

Taxonomy of Methods

ECIR 2022

Traditional methods | Neural-based methods

extraction

statistical (e.g. TF.IDF)

graph-based (e.g. TextRank)

supervised feature-based (e.g. Kea)

(Das Gollapalli et al., 2017)

(Alzaidy et al. 2019)

sequence labeling: CRF, Bi-LSTM-CRF

(Mahata et al., 2018)

embedding-based (e.g. Key2Vec)

generation

resource-based (e.g. Kea++)

(Medelyan and Witten, 2006)

sequence-to-sequence (e.g. CopyRNN)

(Medelyan and Witten, 2006) Thesaurus based automatic keyphrase indexing. JCDL.
(Das Gollapalli et al., 2017) Incorporating expert knowledge into keyphrase extraction. AAAI.
(Mahata et al., 2018) Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings. NAACL.
(Alzaidy et al. 2019) Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. WWW.

Taxonomy of Methods

Traditional methods | Neural-based methods

extraction

statistical (e.g. TF.IDF)

graph-based (e.g. TextRank)

supervised feature-based (e.g. Kea)

(Das Gollapalli et al., 2017)

(Alzaidy et al. 2019)

sequence labeling: CRF, Bi-LSTM-CRF

(Mahata et al., 2018)

embedding-based (e.g. Key2Vec)

generation

resource-based (e.g. Kea++)

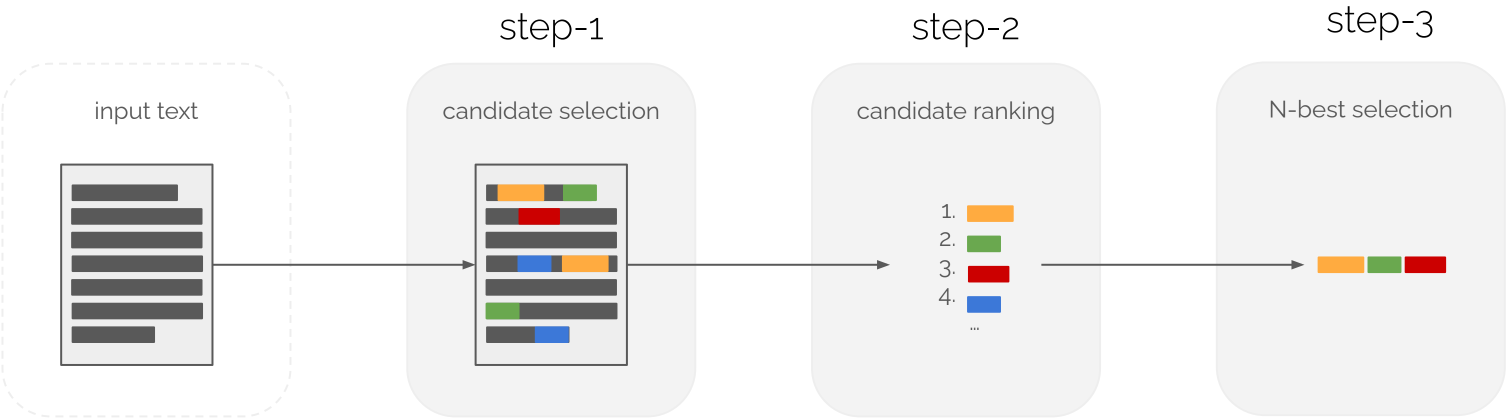
(Medelyan and Witten, 2006)

sequence-to-sequence (e.g. CopyRNN)

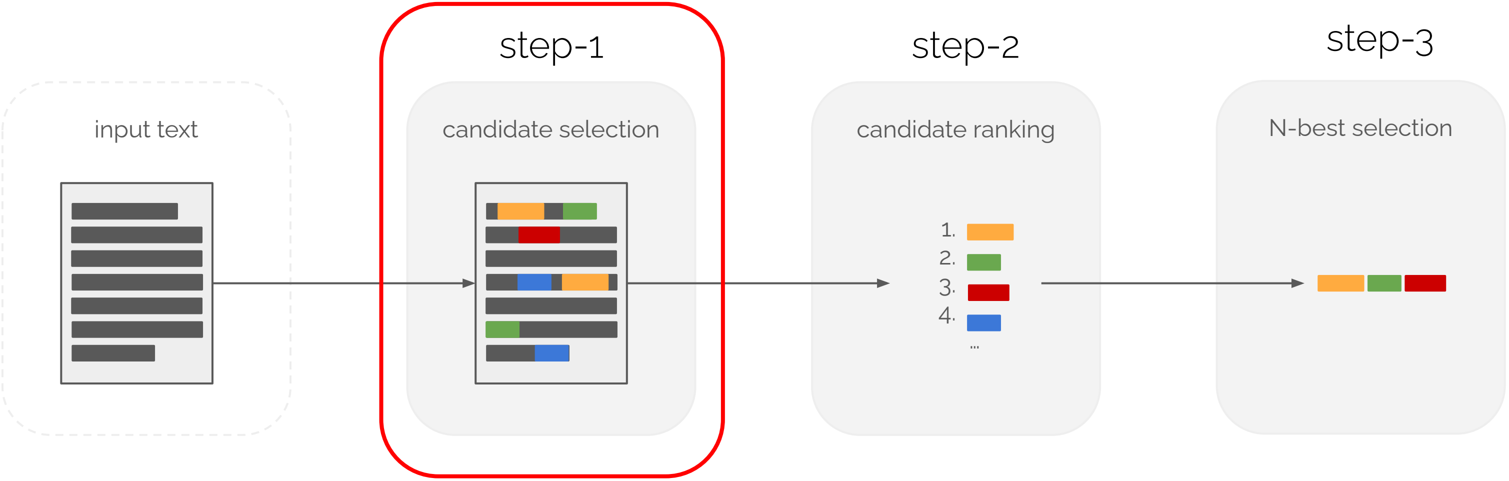
Part 2 of the tutorial

(Medelyan and Witten, 2006) Thesaurus based automatic keyphrase indexing. JCDL.
(Das Gollapalli et al., 2017) Incorporating expert knowledge into keyphrase extraction. AAAI.
(Mahata et al., 2018) Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings. NAACL.
(Alzaidy et al. 2019) Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. WWW.

Traditional Methods for keyphrase extraction



Traditional Methods for keyphrase extraction



Candidate selection

- Identify the words and phrases that are eligible to be keyphrases
  - Mostly noun phrases, composed of up to three words (~90%)

5 most frequent POS-patterns of gold keyphrases in kp20k

Freq.	POS-Pattern	Example
21%	`Noun`	*graphs*
17%	`Noun Noun`	*similarity measure*
15%	`Adj Noun`	*empirical study*
5%	`Verb`	*denoising*
4%	`Adj Noun Noun`	*ant colony optimization*

- Requires text pre-processing
  - tokenization, sentence splitting, POS-tagging, NP-chunking, NER

Candidate selection (cont.)

1895.abstr from Inspec

Inverse problems for a mathematical model of ion exchange in a compressible ion exchanger.

n-gram selection + filtering →

inverse, inverse problems, problems, mathematical, mathematical model, model, ion, ion exchange, exchange, compressible, compressible ion, compressible ion exchanger, ion, ion exchanger, exchanger

NP selection →

inverse problems, mathematical model, compressible ion exchanger

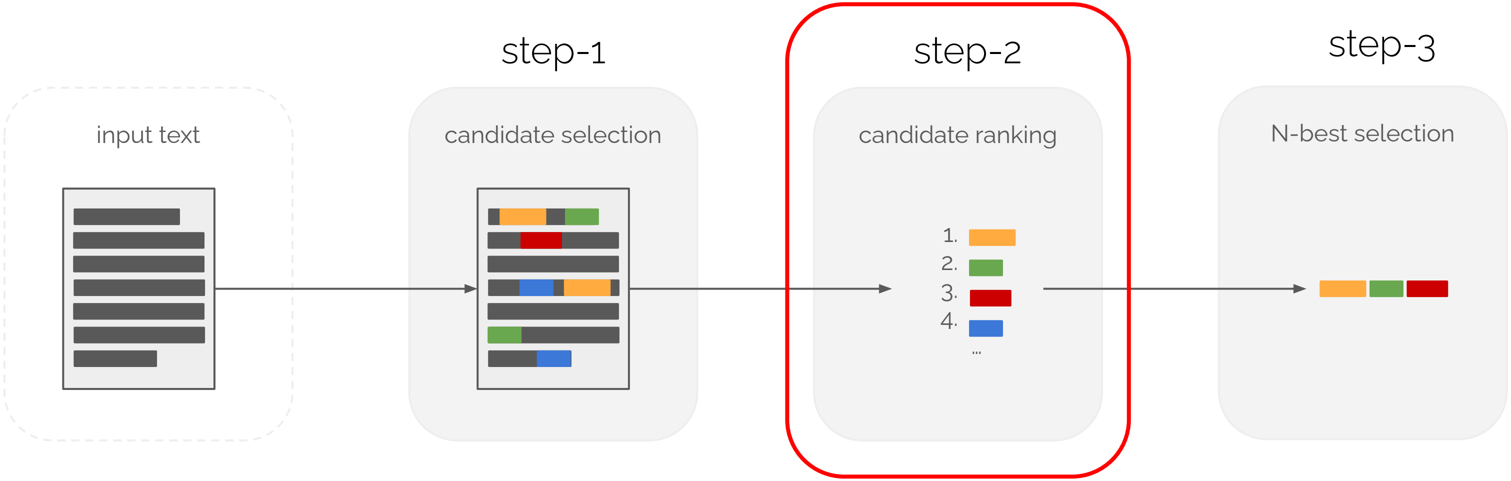
number of candidates to rank

max. reachable recall

- Balances the **search space** and the **upper bound performance**
- Apply filtering techniques to remove spurious candidates
  - e.g. PDF to text → muddled sentences, tables, equations, etc.
  - simple text cleaning → ~+2% in f@10 (boudin et al. 2016)

(Boudin et al., 2016) How Document Pre-processing affects Keyphrase Extraction Performance. WNUT.

Traditional Methods for keyphrase extraction



Candidate ranking

- Assign a weight/score to each keyphrase candidates
  - candidates are ranked using a **weighting function** (unsupervised)
  - candidates are **classified as keyphrase or not** (supervised)

- Statistical methods (unsupervised)
  - frequency-based, position-based, lexical/syntactic-based features

    e.g. TF, IDF, PMI, LM — e.g. candidate offsets, distribution — e.g. PoS pattern, casing

  - commonly-used methods are TF.IDF, LM (Tomokiyo and Hurst, 2003), YAKE (Campos et al., 2020)

$$\delta_{\mathbf{w}}(LM^{N}_{\mathbf{fg}} \parallel LM^{1}_{\mathbf{bg}})$$

$$S(t) = \frac{T_{Rel} * T_{Position}}{T_{Case} + \frac{TF_{Norm}}{T_{Rel}} + \frac{T_{Sentence}}{T_{Rel}}}$$

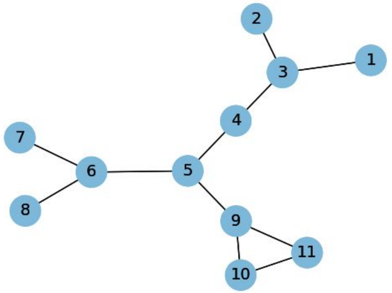
(Campos et al., 2020) YAKE! Keyword extraction from single documents using multiple local features. Information Sciences.
(Tomokiyo and Hurst, 2003) A Language Model Approach to Keyphrase Extraction. MWE workshop.

Candidate ranking (cont.)

- Graph-based ranking methods (unsupervised)
  - Seminal work TextRank (Mihalcea and Tarau, 2004)

  1. build a **graph representation of the document** where nodes are lexical units and edges are semantic relations between them
  2. **rank nodes using a graph-theoretic measure**, from which the top-ranked ones are used to form keyphrases

Page Rank
Iteration init



https://stellasia.github.io/blog/2020-03-07-page-rank-animation-with-networkx-numpy-and-matplotlib/

$$S(c_i) = (1-\lambda) + \lambda \cdot \sum_{c_j \in \mathcal{I}(c_i)} \frac{w_{ij} \cdot S(c_j)}{\sum_{c_k \in \mathcal{O}(c_j)} w_{jk}}$$

- Overview of existing methods
  - node ranking functions : k-core (Tixier et al., 2016), PositionRank (Florescu and Caragea, 2017)
  - topic-based methods : TopicRank (Bougouin et al., 2013), TopicalPageRank (Sterckx et al., 2015)
  - external-resources : ExpandRank (Wan and Xiao, 2008), CiteTextRank (Gollapalli and Caragea, 2014)

(Mihalcea and Tarau, 2004) TextRank: Bringing order into text. EMNLP.
(Wan and Xiao, 2008) Collabrank: Towards a collaborative approach to single-document keyphrase extraction. COLING.
(Bougouin et al., 2013) Topicrank: Graph-based topic ranking for keyphrase extraction. IJCNLP.
(Gollapalli and Caragea, 2014) Extracting Keyphrases from Research Papers Using Citation Networks. AAAI.
(Sterckx et al., 2015) Topical word importance for fast keyphrase extraction. WWW.
(Tixier et al., 2016) A graph degeneracy-based approach to keyword extraction. EMNLP.
(Florescu and Caragea, 2017) Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. ACL.

Candidate ranking (cont.)

- Keyphrase extraction as a binary classification task (supervised)
  - Train to classify candidates as **keyphrase** or **not keyphrase**
  - commonly-used methods : Kea (Witten et al., 1999), WINGNUS (Nguyen and Luong, 2010)

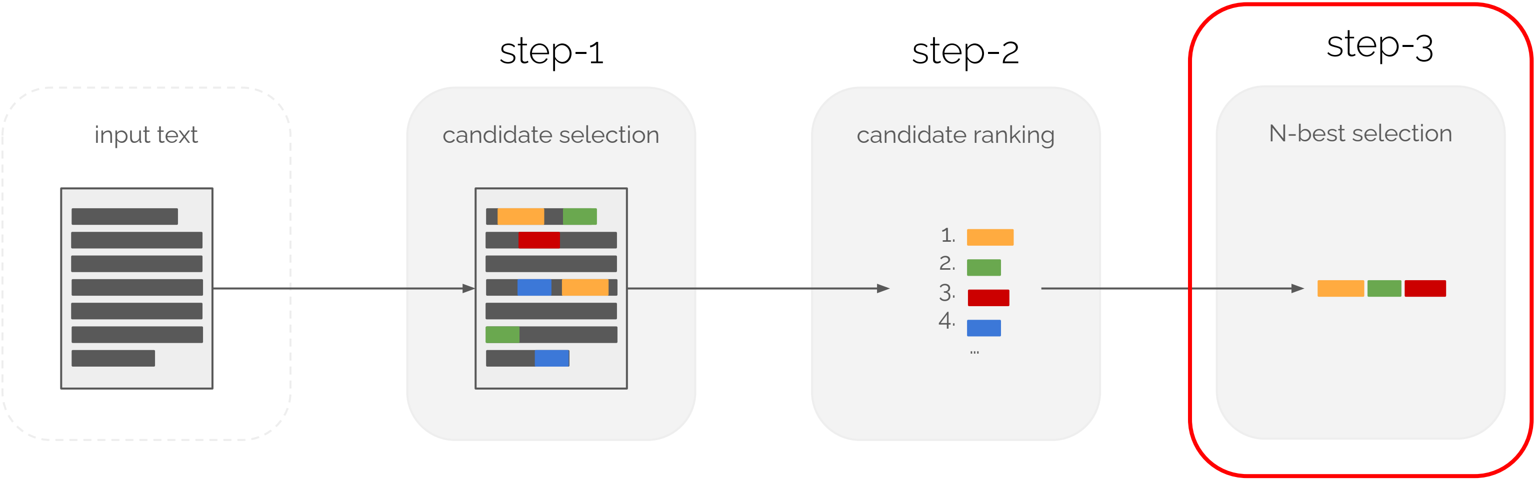
$$P[yes] = \frac{Y}{Y+N} P_{TF\times IDF}[t \mid yes]\, P_{distance}[d \mid yes]$$

F1-F3 (*n*): TF×IDF, term frequency, term frequency of substrings.
F4-F5 (*n*): First and last occurrences (word offset).
F6 (*n*): Length of phrases in words.
F7 (*b*): Typeface attribute (available when PDF is present) – Indicates if any part of the candidate phrase has appeared in the document with bold or italic format, a good hint for its relevance as a keyphrase.
F8 (*b*): InTitle – shows whether a phrase is also part of the document title.
F9 (*n*): TitleOverlap – the number of times a phrase appears in the title of other scholarly documents (obtained from a dump of the DBLP database).
F10-F14 (*b*): Header, Abstract, Intro, RW, Concl – indicate whether a phrase appears in headers, abstract, introduction, related work or conclusion sections, respectively.
F15-F19 (*n*): HeaderF, AbstractF, IntroF, RWF, ConclF - indicate the frequency of a phrase in the headers, abstract, introduction, related work or conclusion sections, respectively.

- Require few training samples, outperform unsupervised methods (Gallina et al., 2020)

(Witten et al., 1999) Kea: Practical automatic keyphrase extraction, DL.
(Nguyen and Luong, 2010) WINGNUS: Keyphrase Extraction Utilizing Document Logical Structure. SemEval.
(Gallina et al., 2020) Large-Scale Evaluation of Keyphrase Extraction Models. JCDL.

Traditional Methods for keyphrase extraction



N-best selection

ECIR 2022

- Select the **N** highest-ranked candidates as keyphrases
  - ⚠️ redundancy within the selected keyphrases should be minimized!
  - Major issue for methods that rank candidates according to their component words
    - Over-generation errors (Hasan et al., 2014)

Rank	keyphrase
1.	machine learning
2.	computer algorithms
~~3.~~	~~machine~~
~~4.~~	~~learning~~
3.	experience
4.	artificial intelligence
5.	study

(Hasan and Ng, 2014) Automatic Keyphrase Extraction: A Survey of the State of the Art. ACL.

Results

- Large scale evaluation of traditional methods (Gallina et al., 2020)

		Scientific articles						*Paper abstracts*						*News articles*					
		PubMed		**ACM**		**SemEval**		**Inspec**		**WWW**		**KP20k**		**DUC-2001**		**KPCrowd**		**KPTimes**	
	Model	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP
Statistical methods (unsupervised)	FirstPhrases	15.4	14.7	13.6	13.5	13.8	10.5	29.3	27.9	10.2	9.8	13.5	12.6	24.6	22.3	17.1	16.5	9.2	8.4
	TF×IDF	16.7	16.9	12.1	11.4	17.7	12.7	**36.5**	**34.4**	9.3	10.1	11.6	12.3	23.3	21.6	16.9	15.8	9.6	9.4
Graph-based methods (unsupervised)	TextRank	1.8	1.8	2.5	2.4	3.5	2.3	35.8	31.4	8.4	5.6	10.2	7.4	21.5	19.4	7.1	9.5	2.7	2.5
	PositionRank	4.9	4.6	5.7	4.9	6.8	4.1	34.2	32.2	11.6†	8.4	14.1†	11.2	28.6†	**28.0**†	13.4	12.7	8.5	6.6
	MultipartiteRank	15.8	15.0	11.6	11.0	14.3	10.6	30.5	29.0	10.8†	10.4	13.6†	13.3†	25.6	24.9†	**18.2**	**17.0**	11.2†	10.1†
Classification method (supervised)	Kea	18.6†	18.6†	14.2†	13.3	19.5†	**14.7**†	34.5	33.2	11.0†	10.9†	14.0†	13.8†	26.5†	24.5†	17.3	16.7	11.0†	10.8†
Neural-based method (supervised)	CopyRNN	**24.2**†	**25.4**†	**24.4**†	**26.3**†	**20.3**†	13.8	28.2	26.4	**22.2**†	**24.9**†	**25.4**†	**28.7**†	10.5	7.2	8.4	4.2	**39.3**†	**50.9**†

- Outperformed by neural-based methods on 6/9 datasets
  - Still useful when no training data is available
  - Three models could be considered as baselines TFxIDF, MultipartiteRank and Kea

(Gallina et al., 2020) Large-Scale Evaluation of Keyphrase Extraction Models. JCDL.

Summary

- Pros
  - Efficiency
  - Interpretability
  - Generalization (languages, domains)
- Cons
  - Pipeline approach : errors are propagated
  - Produce only present keyphrases
  - Overall performance

- A basis for unsupervised neural extractive methods (Part 2.1 of the tutorial)
- Used for producing silver-standard training data for unsupervised keyphrase generation (Part 2.2 of the tutorial)

Part I - Outline

- Introduction to keyphrasification
  - Definitions and applications
- Datasets and evaluations
- Traditional methods for keyphrase extraction
- Hands-on practice with PKE

Overview of pke

- pke is an open source python-based keyphrase extraction toolkit
- end-to-end pipeline in which each component can be modified
- Installation

```
pip install git+https://github.com/boudinfl/pke.git
python -m spacy download en_core_web_sm
```

English model

Overview of pke (cont.)

- standardized API for extracting keyphrases from a document

```
import pke

# initialize keyphrase extraction model, here TopicRank
extractor = pke.unsupervised.TopicRank()

# load the content of the document, here document is expected to be a simple
# test string and preprocessing is carried out using spacy
extractor.load_document(input='text', language='en')

# keyphrase candidate selection, in the case of TopicRank: sequences of nouns
# and adjectives (i.e. `(Noun|Adj)*`)
extractor.candidate_selection()

# candidate weighting, in the case of TopicRank: using a random walk algorithm
extractor.candidate_weighting()

# N-best selection, keyphrases contains the 10 highest scored candidates as
# (keyphrase, score) tuples
keyphrases = extractor.get_n_best(n=10)
```

FirstPhrases, TfIdf, KPMiner, YAKE, TextRank, SingleRank, TopicRank, TopicalPageRank, PositionRank, MultipartiteRank, Kea

Each step can be parameterized or modified

Hands-on session in **1 min**

https://github.com/keyphrasification/hands-on-with-pke

Part 1 : Getting started with pke and keyphrase extraction

Part 2 : Model parameterization

Part 3 : Benchmarking models