Génération automatique de mots-clés

Journée d'études Mots/Machines #5 : Terminologie

Florian Boudin

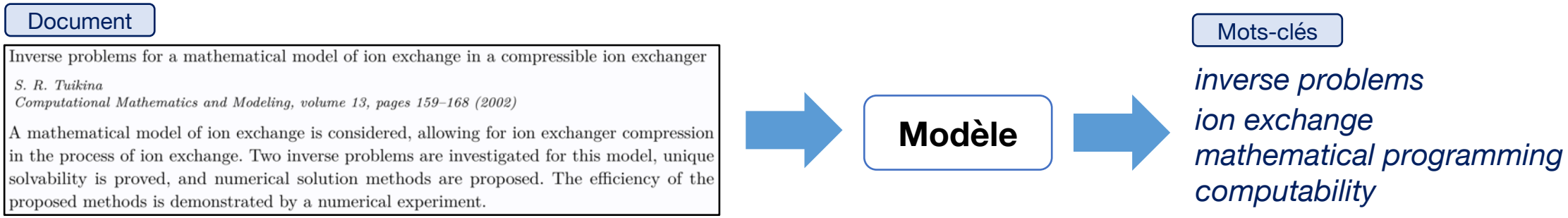
LS2N, Nantes Université

florian.boudin@univ-nantes.fr

Génération automatique de mots-clés

Qu'est ce que c'est ?

- Produire un **ensemble de mots ou d'expressions polylexicales** qui décrivent les **principaux sujets** d'un document



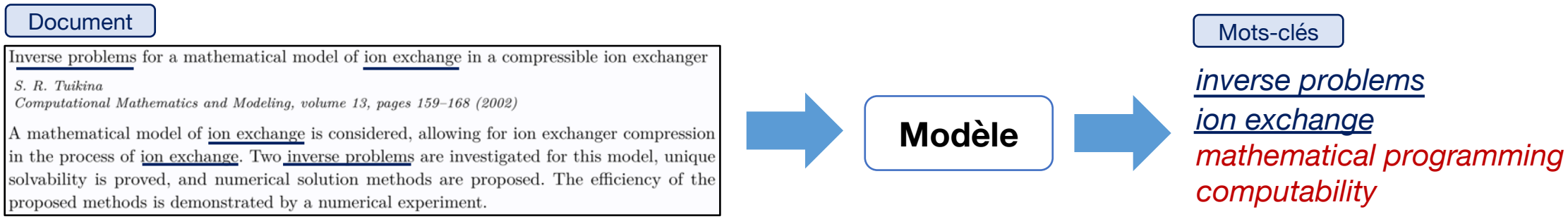
Génération automatique de mots-clés

Extraction de mots-clés

- Identification des unités textuelles les plus importantes

Génération de mots-clés

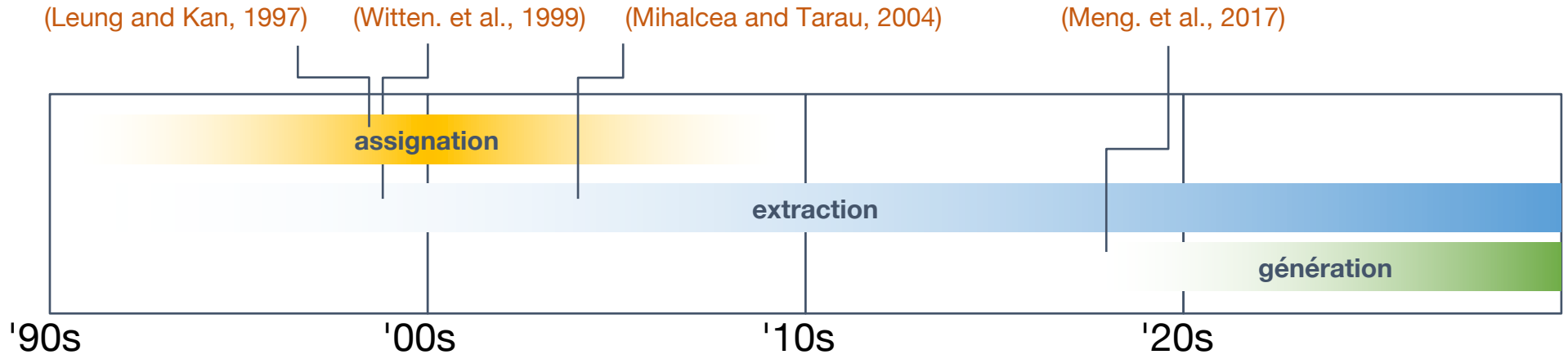
- Extraction + génération de mots-clés « absents » du texte source



Génération automatique de mots-clés

Premiers travaux portent sur le catalogage de documents (Fagan, 1987)

- **assignation** : assigner des entrées d'un thesaurus (e.g. MeSH/UMLS)
- **extraction** : identifier les unités textuelles importantes d'un texte
- **génération** : produire des mots-clés qui résume le contenu d'un texte



(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.

Génération automatique de mots-clés

À quoi ça sert ?

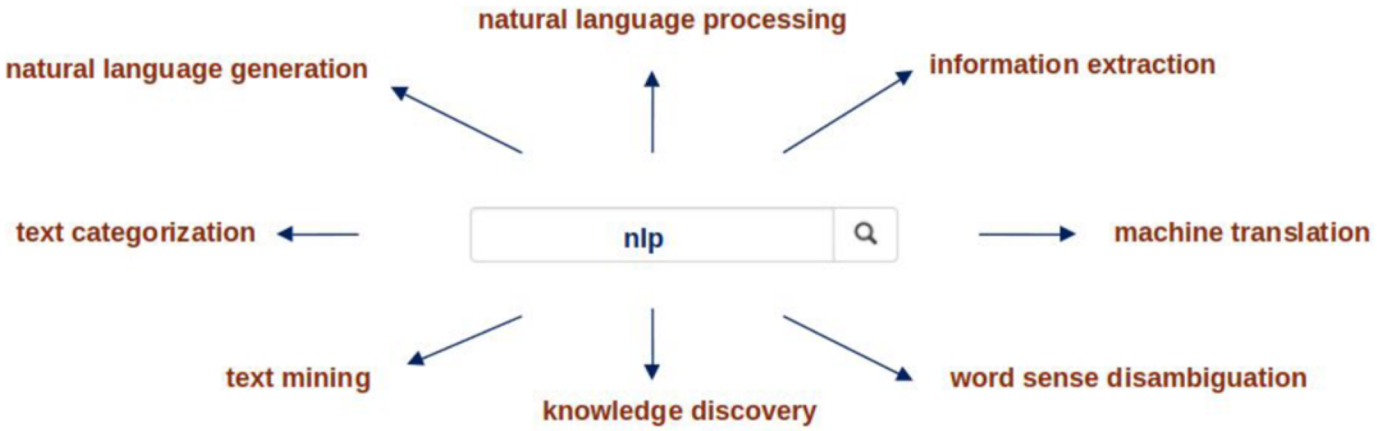
- Les mots-clés **distillent les informations importantes** et sont utiles pour de nombreuses applications en TAL et en RI
  - Indexation documentaire (Jones and Staveley, 1999; Gutwin et al., 1999)
  - Résumé automatique (Zha, 2002; Wan et al., 2007)
  - Catégorisation de texte (Hulth and Megyesi, 2006)
  - *Opinion mining* (Berend, 2011)
  - Recommandation d'articles (Collins and Beel, 2019)

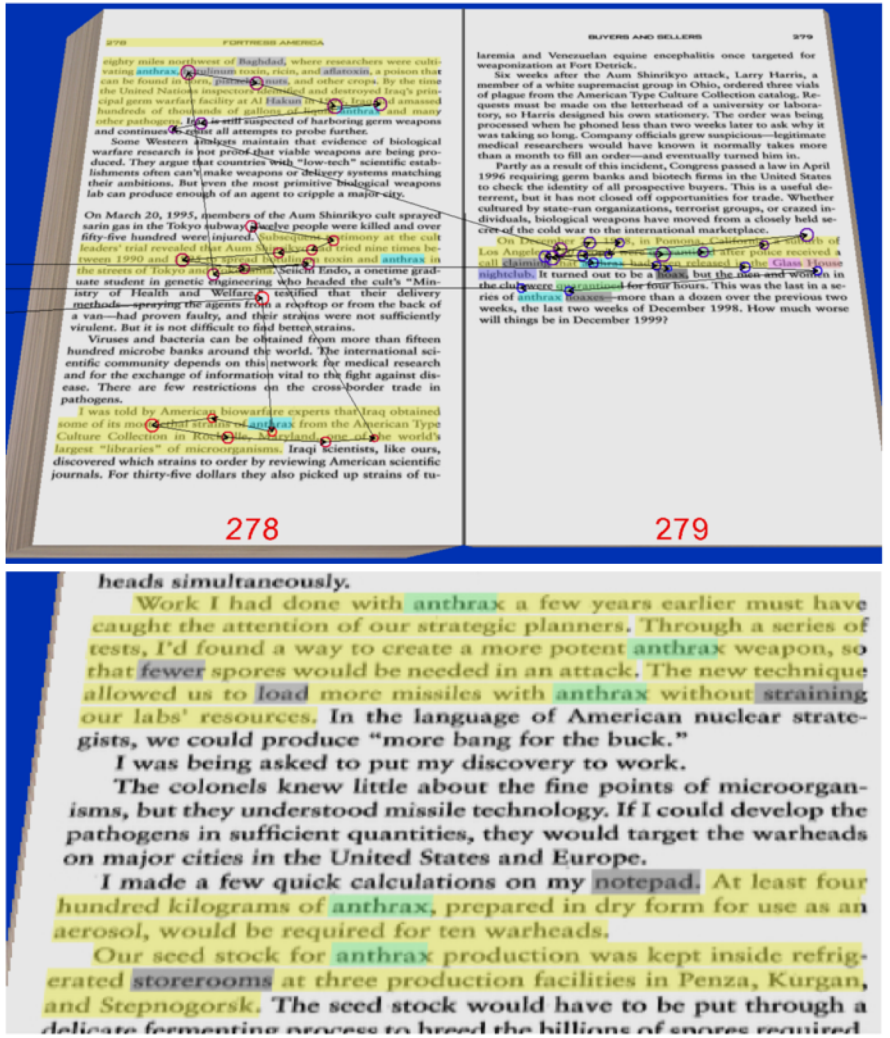
(Gutwin et al., 1999) Improving browsing in digital libraries with keyphrase indexes, Decision Support Systems.
(Jones and Staveley, 1999) Phrasier: a system for interactive document retrieval using keyphrases, SIGIR.
(Zha, 2002) Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering, SIGIR.
(Wan et al., 2007) Towards an Iterative Reinforcement Approach for Simultaneous Document Summarization and Keyword Extraction, ACL.
(Hulth and Megyesi, 2006) A study on automatically extracted keywords in text categorization, ACL.
(Berend, 2011) Opinion Expression Mining by Exploiting Keyphrase Extraction, IJCNLP.
(Collins and Beel, 2019) Document embeddings vs. keyphrases vs. terms for recommender systems: A large-scale online evaluation, JCDL.

Génération automatique de mots-clés

À quoi ça sert ? (cont.)

- Lecture augmentée (Chi et al., 2007)
- Expansion de requête (Song et al., 2006)



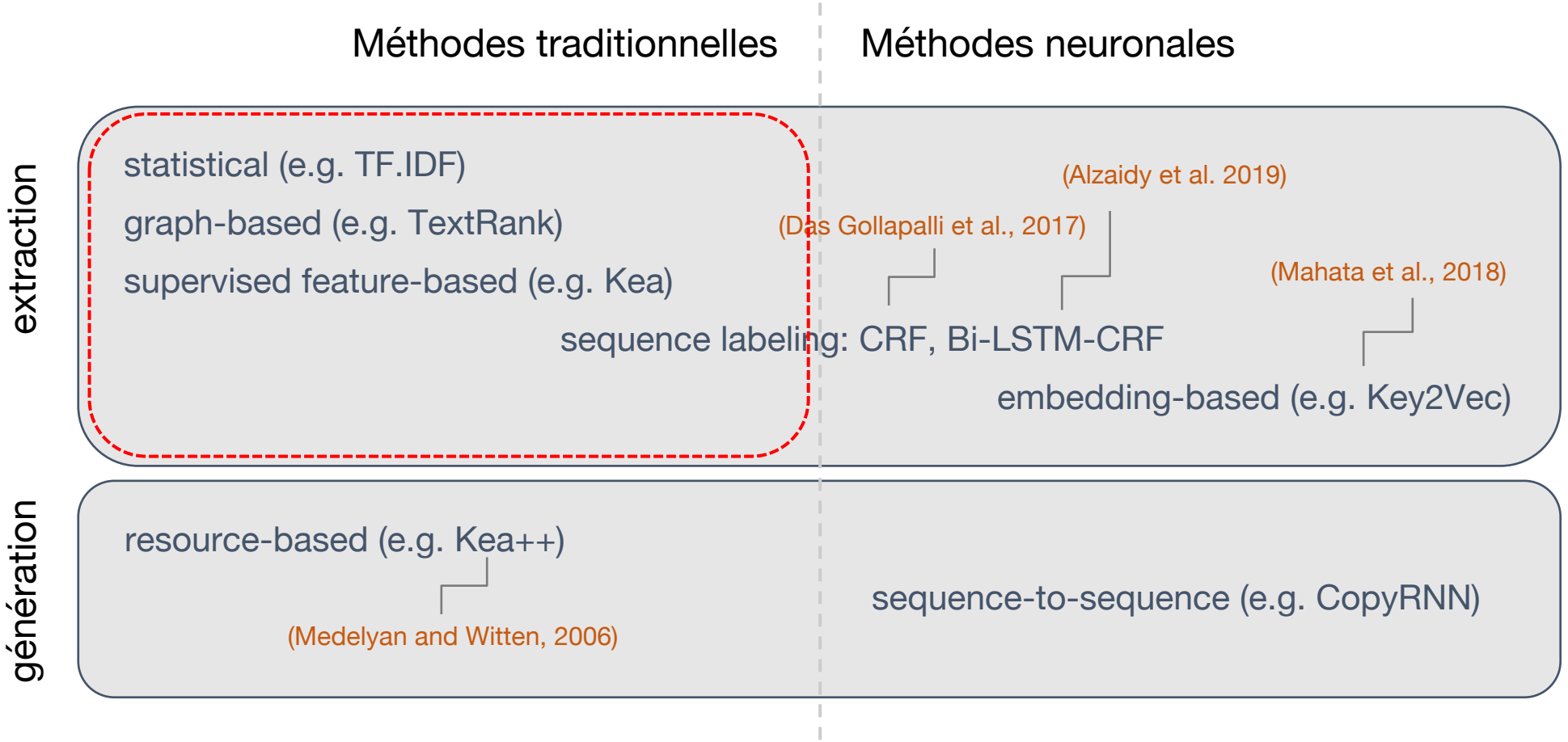


(Chi et al., 2007) Visual foraging of highlighted text: An eye-tracking study. HCI.
(Song et al., 2006) Keyphrase extraction-based query expansion in digital libraries. JCDL.

Plan

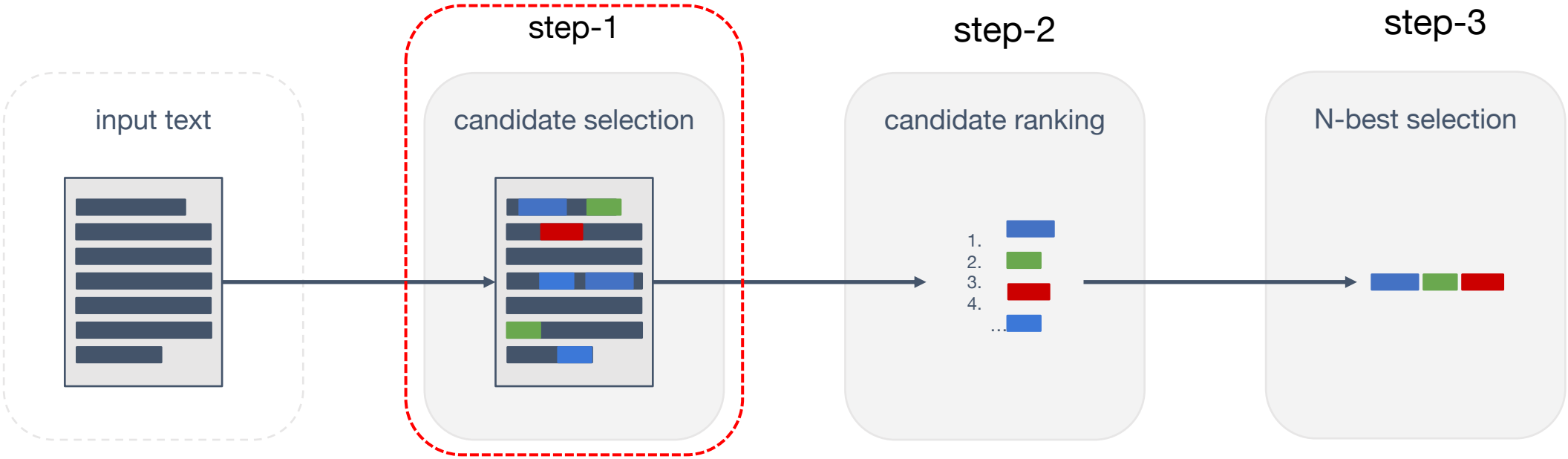
- Introduction
- Méthodes existantes
- Jeux de données et évaluation
- Challenges
- Hands on session

Modèles (taxonomie)



(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.

Méthodes traditionnelles



Sélection des candidats

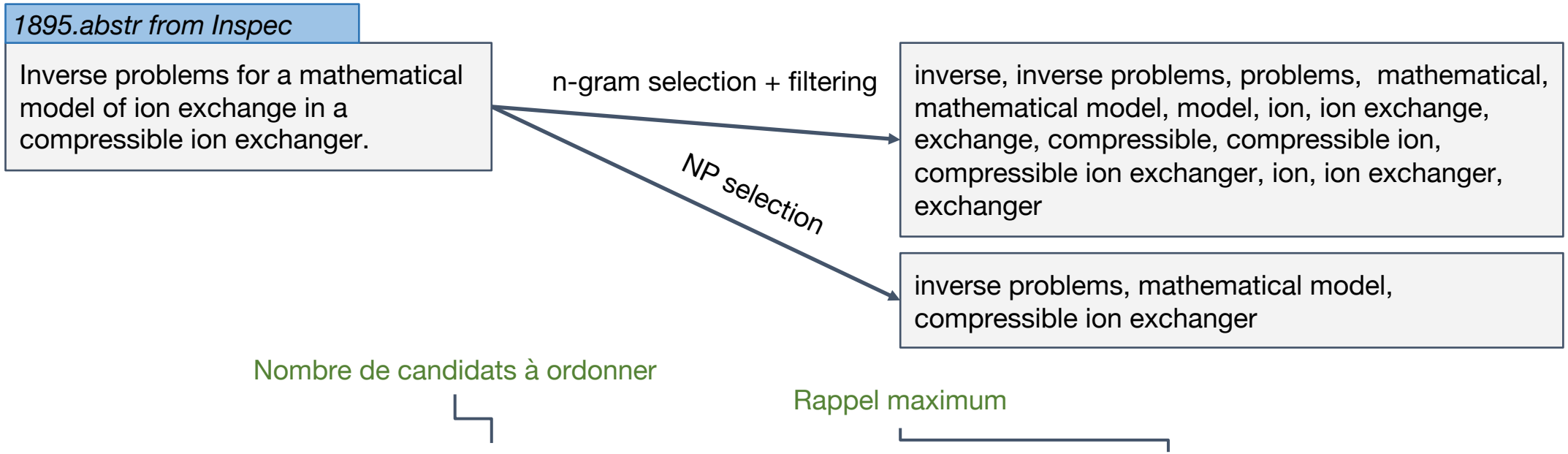
- Identifier les mots/expressions éligibles pour être des mots-clés
  - Principalement des groupes nominaux, 3 mots max. (~90 %)

5 POS-pattern les plus fréquents des mots-clés de référence dans l'ensemble 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*

- Nécessite des pré-traitements
  - tokenization, découpage en phrases, POS-tagging, NP-chunking, NER

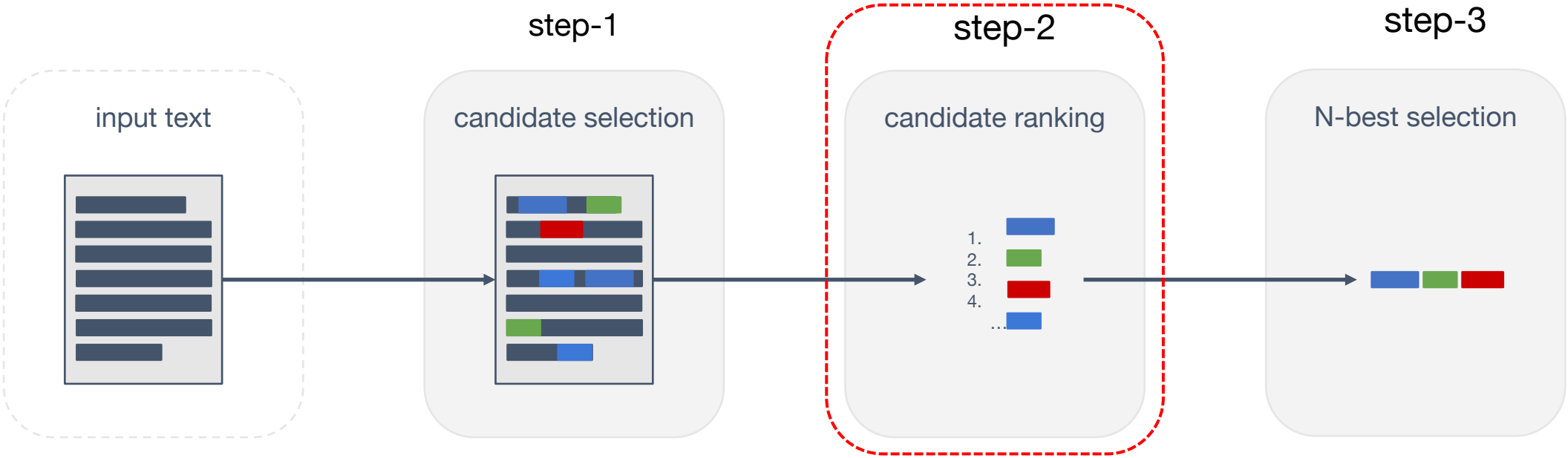
Sélection des candidats (cont.)



- Equilibre l'espace de recherche et la limite sup. de performance
- Techniques de filtrage pour supprimer les « faux » candidats
  - e.g. PDF to text → phrases mélangées, tables, équations, etc.
  - simple nettoyage → ~+2% in f@10 (Boudin et al. 2016)

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

Méthodes traditionnelles



Ordonnancement des candidats

- Calculer un score/poids pour chaque candidat
  - candidats ordonnés selon une **fonction de pondération** (non supervisé)
  - candidats **classés comme mot-clé ou non** (supervisé)

- Méthodes statistiques (non supervisé)
  - frequency-based, position-based, lexical/syntactic-based features

e.g. TF, IDF, PMI, LM

e.g. candidate offsets, distribution

e.g. PoS pattern, casing

  - Méthodes les + utilisées sont TF.IDF, LM (Tomokiyo and Hurst, 2003), YAKE (Campos et al., 2020)

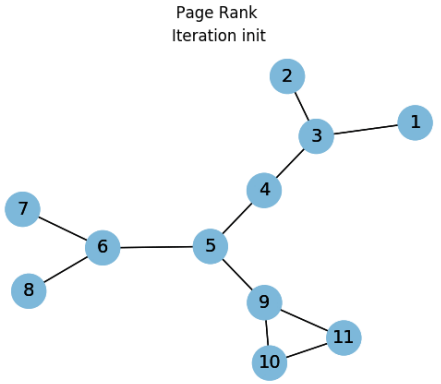
$$\delta_{\mathbf{w}}(LM^{N}_{\mathrm{fg}} \parallel LM^{1}_{\mathrm{bg}})$$

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

(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.

Ordonnancement des candidats (cont.)

- Méthodes de graphes (non supervisé)
  - Travail séminal TextRank (Mihalcea and Tarau, 2004)
  1. construire une représentation graphique du document
  2. ordonner les nœuds avec une mesure de la théorie des graphes



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

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

- Méthodes existantes
  - 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.

Ordonnancement des candidats (cont.)

- Méthodes par classification binaire (supervisé)
  - Entrainer à classer les candidats en mot-clé ou non
  - Méthodes les + utilisées : Kea (Witten et al., 1999), WINGNUS (Nguyen and Luong, 2010)

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

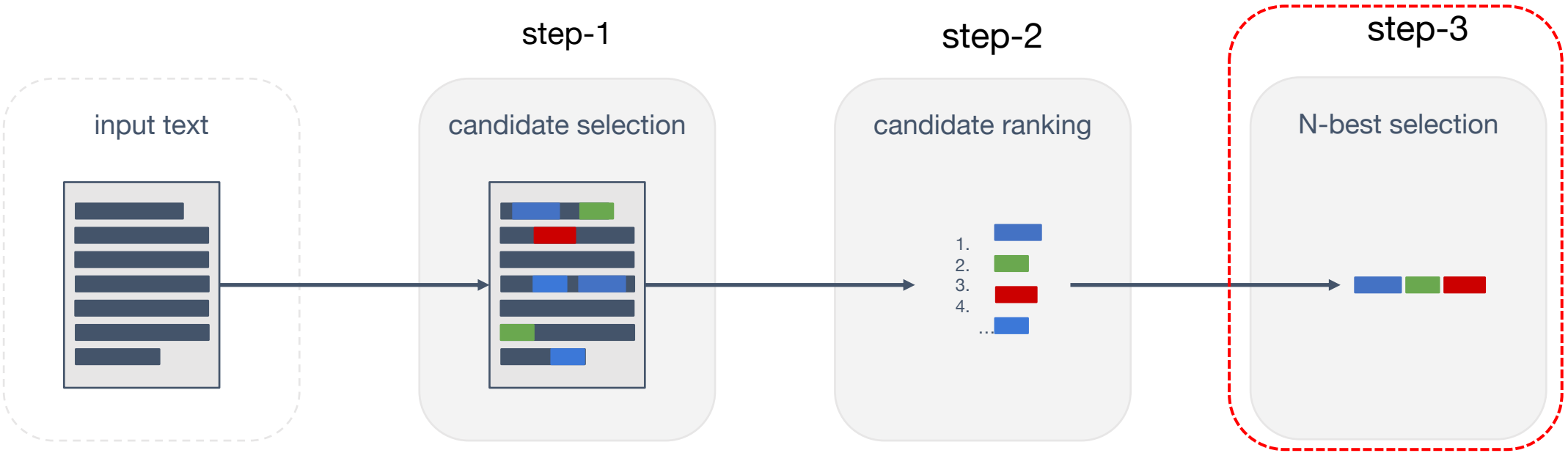
F1-F3 (*n*): TF×IDF, term frequency, term frequency of substrings.
F4-F5 (*n*): First and last occurrences (word offset).
F6 (*n*): Length of phrases in words.
F7 (*b*): Typeface attribute (available when PDF is present) – Indicates if any part of the candidate phrase has appeared in the document with 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.

- Nécessite peu de données, performance > non supervisé (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.

Méthodes traditionnelles



Extraction des mots-clés

- Sélection des N-meilleurs candidats comme mots-clés
  - ⚠️ redondance dans les mots-clés extraits
  - Problème majeur pour les méthodes de pondération des mots
    - Erreurs de sur-génération (Hasan et al., 2014)

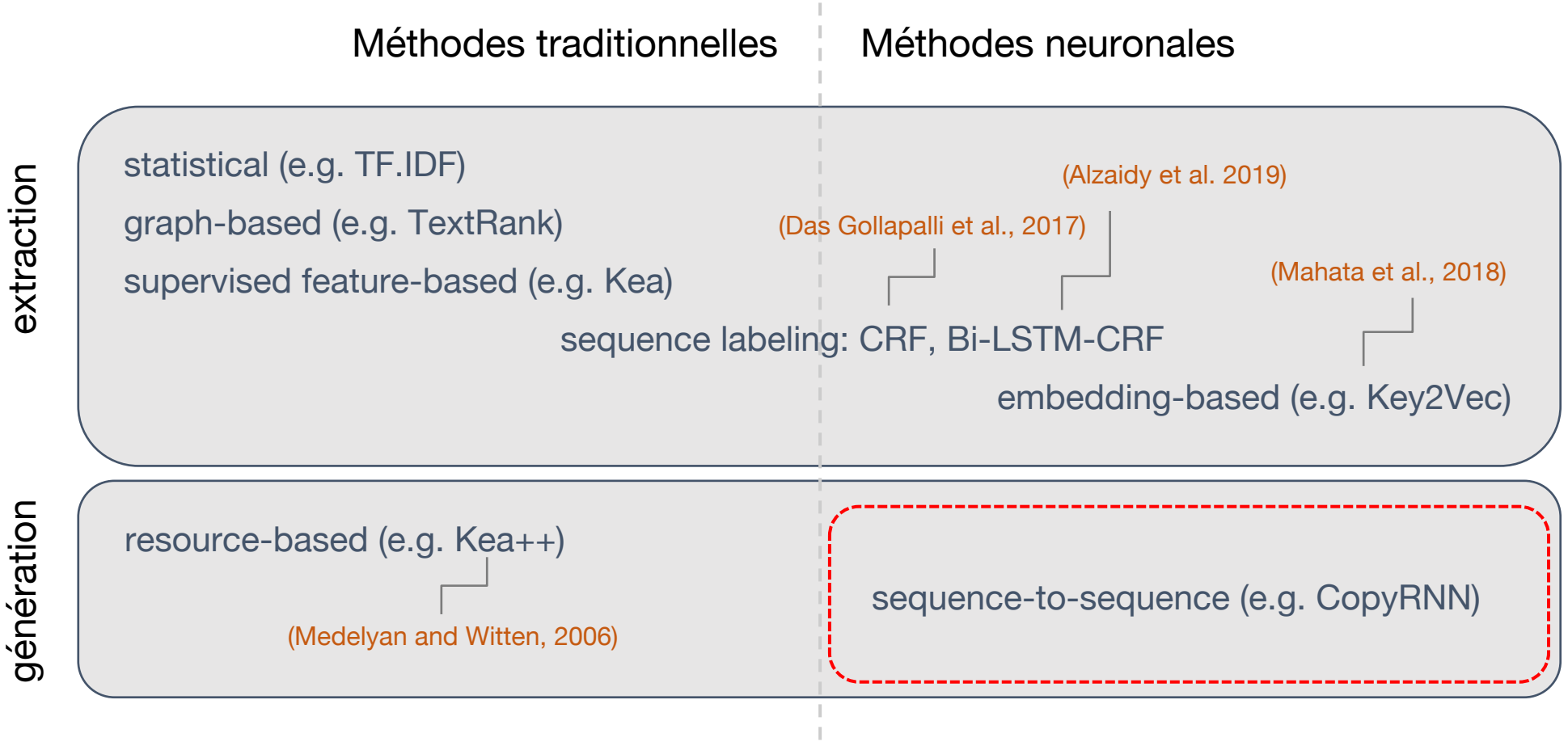
Rang	mot-clé
1.	machine learning
2.	computer algorithms
3.	~~machine~~
4.	~~learning~~
3.	experience
4.	artificial intelligence
5.	study

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

Méthodes traditionnelles

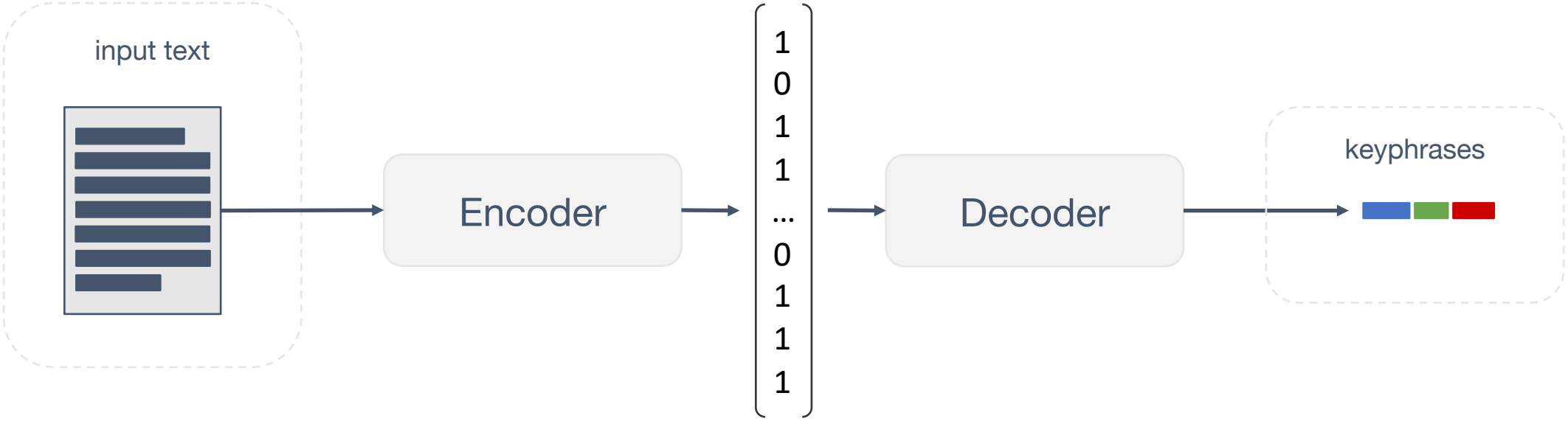
- Avantages
  - Efficacité
  - Interprétabilité
  - Généralisation (langues, domaines)
- Inconvénients
  - Approche *pipeline* : propagation les erreurs
  - Produisent que des mots-clés présents
  - Performance

Modèles (taxonomie)



(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.

Méthodes neuronales

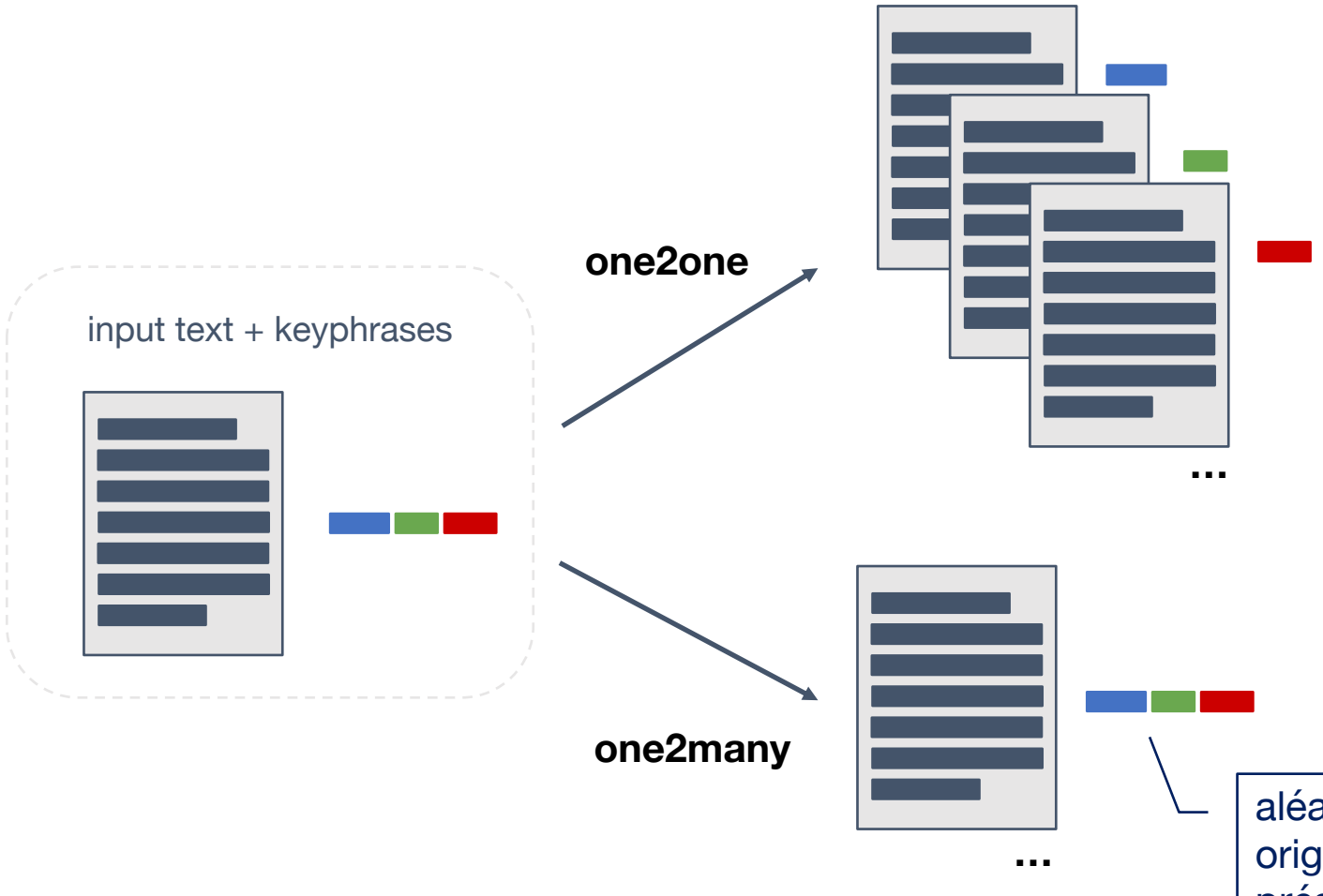


Méthodes neuronales (cont.)

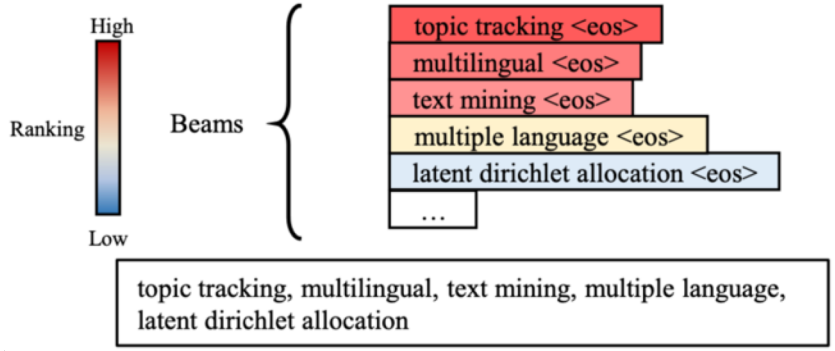
- Paradigme encodeur-décodeur
  - Modèles utilisés : RNN (Meng et al., 2017), transformers (Diao et al., 2020), CNN (Zhang et al., 2017), GCN (Kim et al., 2021)
- Décodeur génère une séquence de mots
  - Vocabulaire différent du document (+ contenu du document)
    - Génération de mots-clés absents
  - Entraînement de bout-en-bout : *one2one*, *one2many*

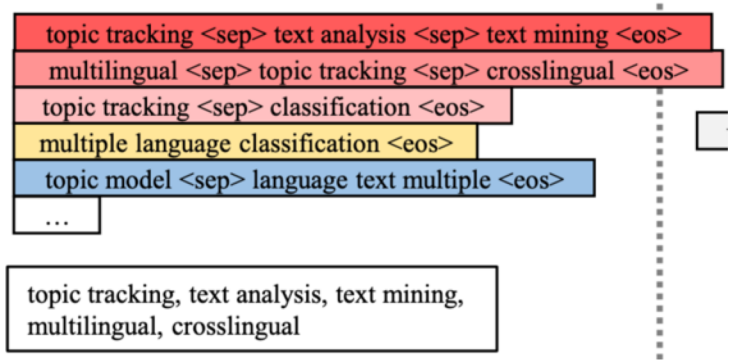
(Meng et al., 2017) Deep keyphrase generation, ACL.
(Diao et al., 2020) Keyphrase Generation with Cross-Document Attention. Arxiv.
(Zhang et al., 2017) Deep keyphrase generation with a convolutional sequence to sequence model. ICSAI.
(Kim et al., 2021) Structure-Augmented Keyphrase Generation. EMNLP.

Méthodes neuronales (cont.)



Décodage





(Meng et al., 2021) An Empirical Study on Neural Keyphrase Generation. NAACL

Méthodes neuronales

- Avantages
  - Performance
  - Génération de mots-clés absents
  - Approche de bout-en-bout
- Inconvénients
  - Nécessitent de **grandes quantités de données annotées**
  - Efficacité (entraînement et inférence)
  - Généralisation

Plan

- Introduction
- Méthodes existantes
- Jeux de données et évaluation
- Challenges
- Hands on session

Jeux de données

	Corpus	Lang.	Ann.	#Entr.	#Test	#mots
Publications scientifiques	CSTR (Witten et al., 1999)	en	A	130	500	11501
	NUS (Nguyen and Kan, 2007)	en	A∪L	-	211	8398
	PubMed (Schutz, 2008)	en	A	-	1320	5323
	ACM (Krapivin et al., 2009)	en	A	-	2304	9198
	Citeulike-180 (Medelyan et al., 2009)	en	L	-	182	8590
	SemEval-2010 (Kim et al., 2010)	en	A∪L	144	100	7961
	LDKP (Mahata et al., 2022)	en	A	1.3M	10K	4484
Notices bibliographiques	Inspec (Hulth, 2003)	en	I	1000	500	135
	KDD (Caragea et al., 2014)	en	A	-	755	191
	WWW (Caragea et al., 2014)	en	A	-	1330	164
	TermITH-Eval (Bougouin et al., 2016)	fr	I	-	400	165
	KP20k (Meng et al., 2017)	en	A	530K	20K	176
	KPBiomed (Houbre et al., 2022)	en	A	5.6M	20K	271
Articles journalistiques	DUC-2001 (Wan and Xiao, 2008)	en	L	-	308	847
	500N-KPCrowd (Marujo et al., 2012)	en	L	450	50	465
	Wikinews (Bougouin et al., 2013)	fr	L	-	100	314
	KPTimes (Gallina et al., 2019)	en	I	260K	20K	921

Évaluation des mots-clés

- Stratégies d'évaluation
  - Appariement exact
  - Appariement partiel
  - Au travers d'une application
  - Évaluation manuelle

  } Automatisé

- Métriques d'évaluation
  - Précision@K, Rappel@K et F1@K → Méthodes traditionnelles
  - Précision, rappel et F1 → Annotation de séquences
  - Précision@O/M, Rappel@O/M et F1@O/M → Génération de mots-clés

Évaluation des mots-clés (cont.)

Texte source - The development of a mobile manipulator imaging system for bridge crack inspection. A mobile manipulator imaging system is developed for the automation of bridge crack inspection. During bridge safety inspections, an eyesight inspection is made for preliminary evaluation and screening before a more precise inspection. The inspection for cracks is an important part of the preliminary evaluation. Currently, the inspectors must stand on the platform of a bridge inspection vehicle or a temporarily erected scaffolding to examine the underside of a bridge. However, such a procedure is risky. To help automate the bridge crack inspection process, we installed two CCD cameras and a four-axis manipulator system on a mobile vehicle. The parallel cameras are used to detect cracks. The manipulator system is equipped with binocular charge coupled devices for examining structures that may not be accessible to the eye. The system also reduces the danger of accidents to the human inspectors. The manipulator system consists of four arms. Balance weights are placed at the ends of arms 2 and 4, respectively, to maintain the center of gravity during operation. Mechanically, arms 2 and 4 can revolve smoothly. Experiments indicated that the system could be useful for bridge crack inspections.

Référence - mobile manipulator, imaging system, bridge crack inspection, automation, eyesight inspection, ccd cameras, parallel cameras, charge coupled devices, four-axis manipulator

Évaluation des mots-clés (cont.)

Référence - mobile manipulator, imaging system, bridge crack inspection, automation, eyesight inspection, ccd cameras, parallel cameras, charge coupled devices, four-axis manipulator

Mots-clés générés – [bridge crack inspection process, bridge inspection vehicle, bridge safety inspections, bridge crack inspection, mobile manipulator imaging system **]**, precise inspection, eyesight inspection, four-axis manipulator system, inspection, manipulator system

$$precision@K = \frac{top\ K\ in\ Y_{pred} \cap Y_{gold}}{K}$$

$$recall@K = \frac{top\ K\ in\ Y_{pred} \cap Y_{gold}}{|Y_{gold}|}$$

$$F1@K = 2 \times \frac{precision@K \times recall@K}{precision@K + recall@K}$$

K = 5

Précision@5 = 1/5 = 0.2

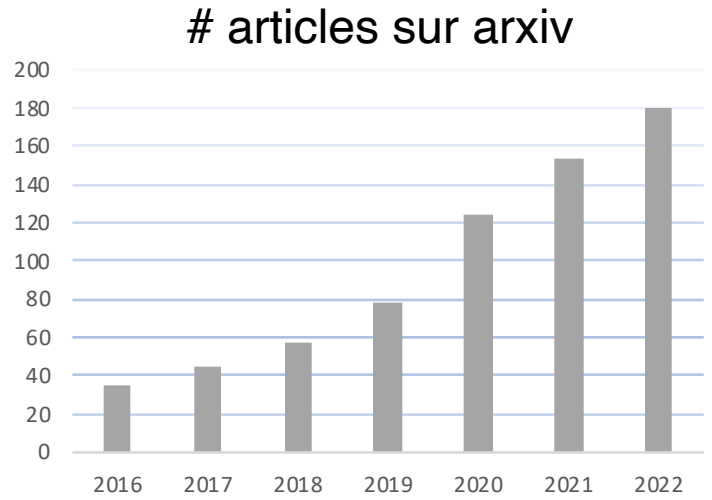
Rappel@5 = 1/9 = 0.11

F1@5 = 2*(0.2*0.11) / (0.2+0.11) = 0.14

Plan

- Introduction
- Méthodes existantes
- Jeux de données et évaluation
- Challenges
- Hands on session

Challenges

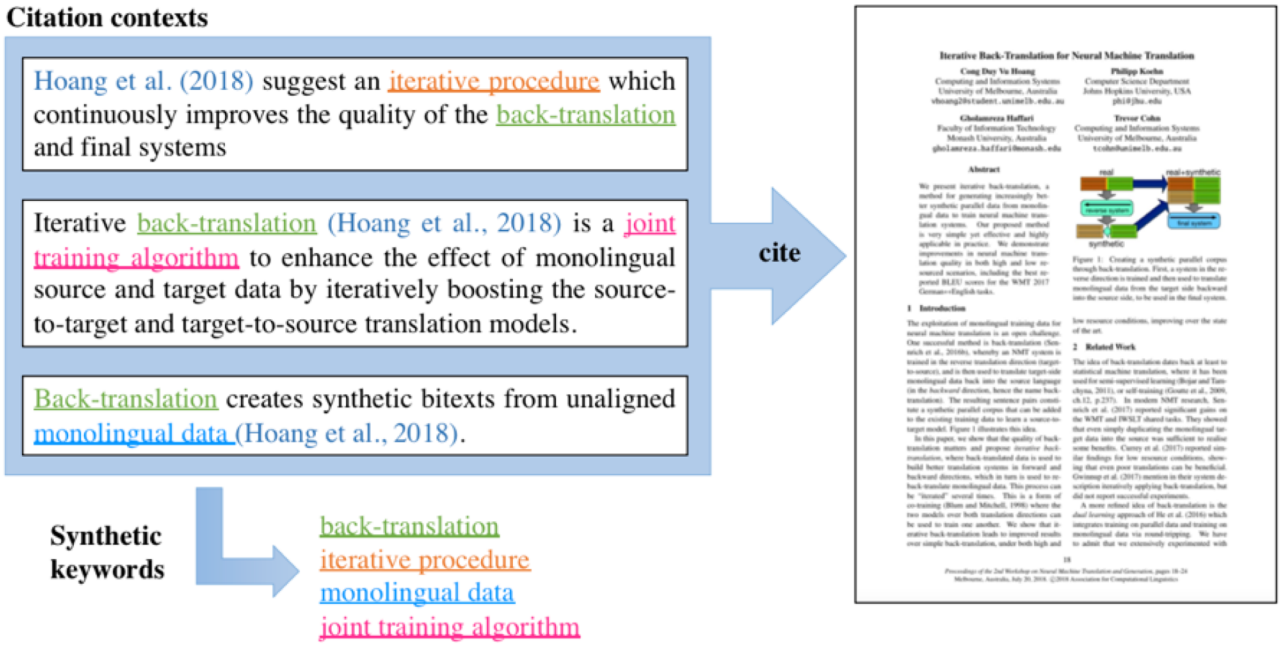


- La génération de mots-clés est un sujet de recherche 🚀
- Nombreux challenges sont à relever
- Présentation de deux enjeux (les plus prégnants) et de pistes de recherche prometteuses (🤞)

Challenges (cont.)

- [C1] Les méthodes neuronales nécessitent beaucoup de données
  - Données annotées sont rares pour beaucoup de domaines / langues
  - Problème aggravé par la faible généralisation des modèles actuels

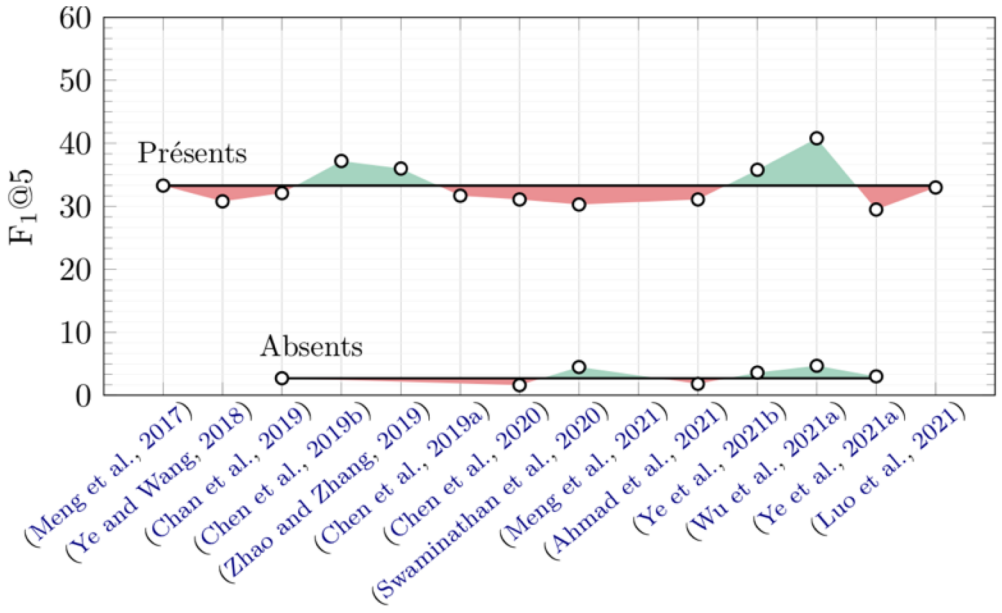
→ Génération de données synthétiques

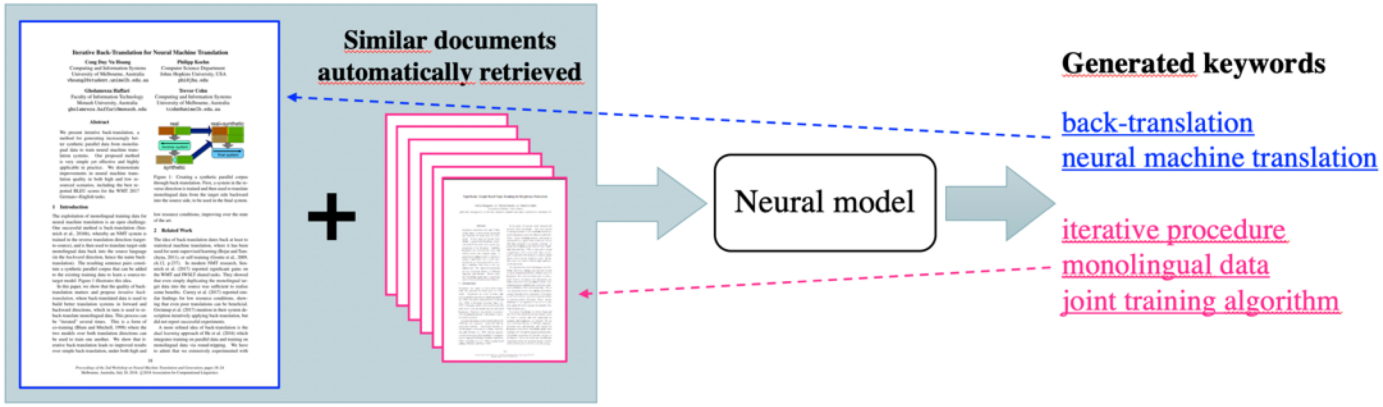


Challenges (cont.)

- [C2] Les méthodes n'arrivent pas à générer de mots-clés absents
  - Elles sont limitées au seul contenu du texte source (notice!)

→ Incorporer des ressources/informations externes





Plan

- Introduction
- Méthodes existantes
- Jeux de données et évaluation
- Challenges
- Hands on session

Hands on session

- Démonstration avec la bibliothèque Python `pke`

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

```
1  import pke

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

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

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

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

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