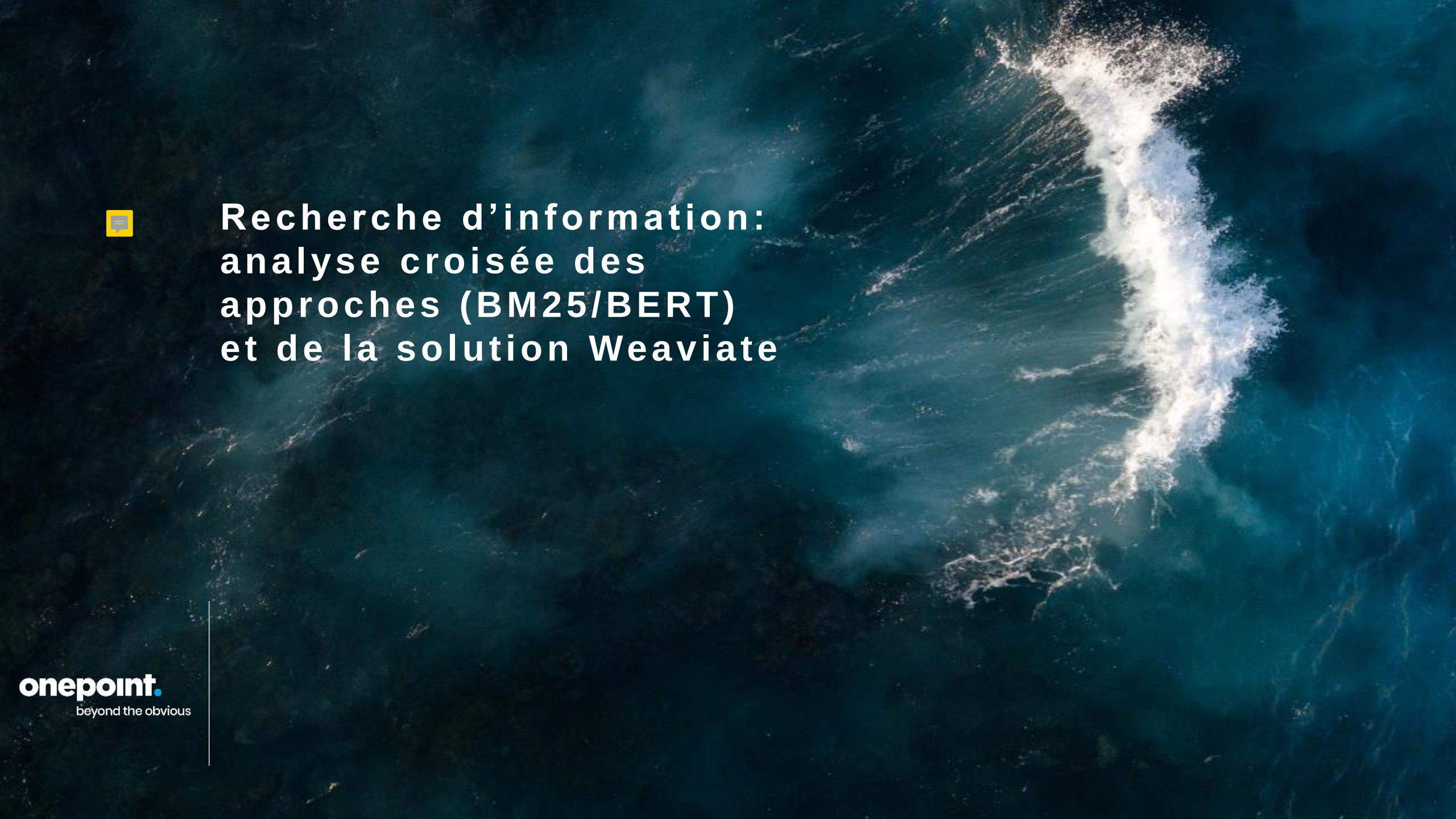


Recherche d’information: analyse croisée des approches (BM25/BERT) et de la solution Weaviate

onepoint.
beyond the obvious

Search

Recherche d’information

RECHERCHE D'INFORMATION

Comment retrouver la ou les informations pertinente dans un corpus de documents ?

Naissance	8 décembre 1927 Lunebourg
Décès	6 novembre 1998 (à 70 ans) Oerlinghausen
Nationalité	Allemande
Formation	Université de Fribourg-en-Brisgau Université de Bielefeld Université Harvard
Activités	Cybernétiste, sociologue, professeur d'université, avocat

Un exemple : la méthode Zettelkasten de Niklas Luhmann

1. Lire une seule fois chaque ressource en prenant un maximum de notes provisoires
2. Organiser de manière permanente ces notes
    1. Ecrire une note par idée, comme ci elle était à destination d'une personne étrangère au sujet
    2. Ajouter des métadonnées à cette note, la reliant à d'autres notes
    3. Cette note est classée derrière la note permanente la plus proche
3. Certaines notes sont utilisées comme point d'entrée.

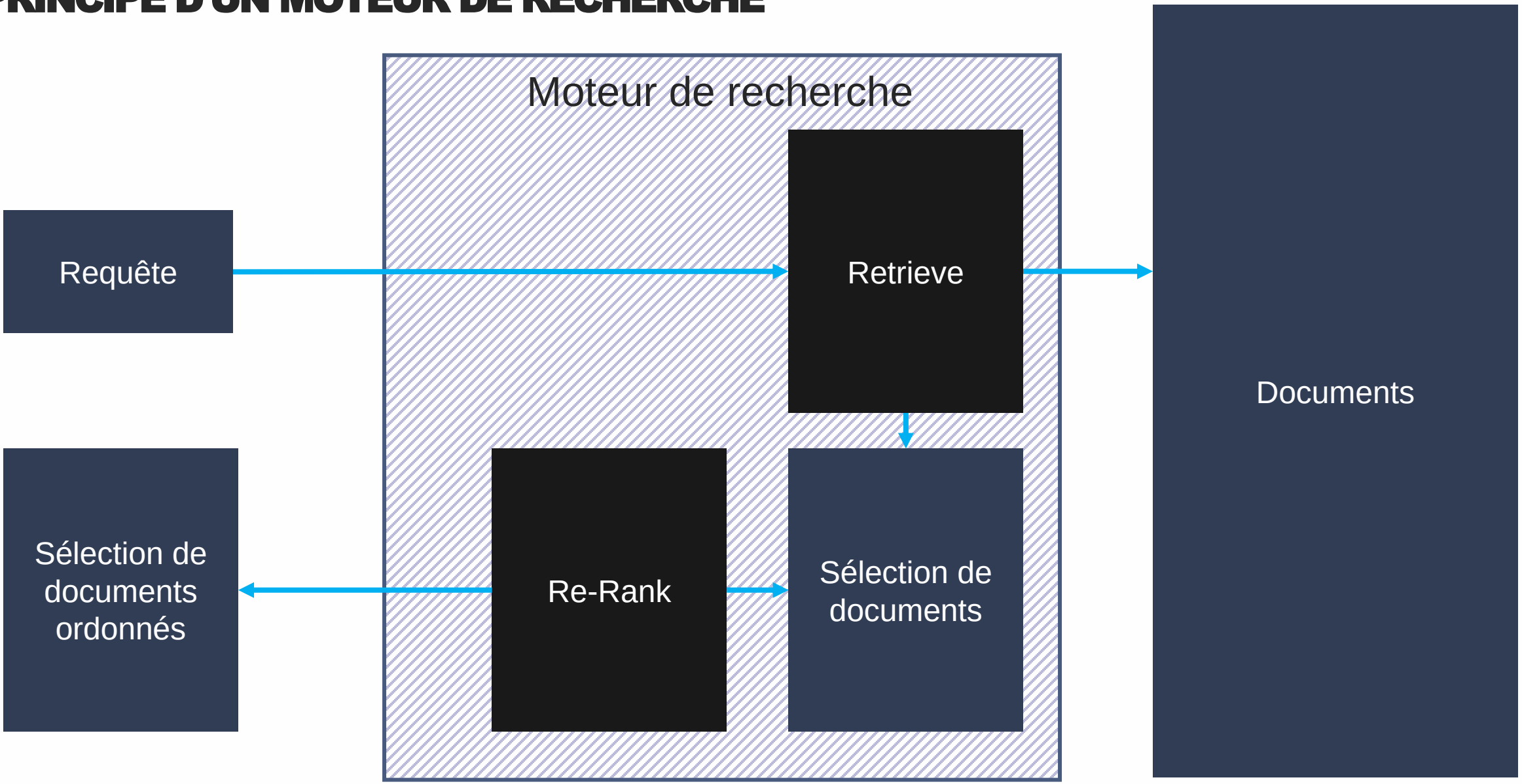
« L'œuvre de Luhmann est abondante. Elle compte plus de 70 livres et près de 400 articles scientifiques abordant les sujets les plus variés, notamment le droit, l'économie, la politique, l'art, la religion, l'écologie, les mass-médias et l'amour »

RECHERCHE D'INFORMATION

Problématique des outils informatiques

- Le corpus : une masse colossale de documents textes plus ou moins homogènes dans leur structure
- Comment fragmenter ces documents en unités pertinentes ?
- Comment convertir ces fragments en vecteurs numériques ?
- Comment créer des connexions entre les fragments proches d'un point de vue sémantique ?
- Comment convertir la requête d'un utilisateur en vecteur numérique compatible avec les vecteurs numériques associés aux fragments de documents ?
- Comment explorer efficacement le corpus pour retrouver les fragments les plus pertinents au regard de la requête ?
- Comment ordonner ces fragments en fonction de leur pertinence ?

PRINCIPE D'UN MOTEUR DE RECHERCHE



BM25 / BERT

L’approche classique

BM25

Classer les documents en fonction de leur pertinence vis-à-vis d'une requête

The Probabilistic Relevance Framework: BM25 and Beyond

By Stephen Robertson and Hugo Zaragoza

BM25

Score utilisé pour classer les documents

Uniquement sur la base des termes présents dans la requête

Favorise les termes discriminants

Fréquence d'apparition dans un document du corpus

Pondère le terme précédent

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)},$$

Pondère le terme suivant

Adapte à la longueur du document

$$tf' = \frac{tf}{B} \tag{3.13}$$

$$w_i^{\text{BM25}}(tf) = \frac{tf'}{k_1 + tf'} w_i^{\text{RSJ}} \tag{3.14}$$

$$= \frac{tf}{k_1 \left((1-b) + b\frac{dl}{avdl}\right) + tf} w_i^{\text{RSJ}} \tag{3.15}$$

BERT ET AL.

- Réseau de neurones avec un système d'attention
- Révolution dans le domaine du traitement automatique du langage naturel
  - Traduction (d'une requête, d'un document ?)
  - Formulation d'une requête plus proche du langage naturel
  - Identification des liens sémantiques => les termes qui n'apparaissent pas dans la requête participent quand même à l'évaluation de la pertinence d'un document
- Pose des problèmes de passage à l'échelle
  - La requête est comparé à chaque extrait du document
  - La longueur de la requête et celle de l'extrait du document sont standardisées

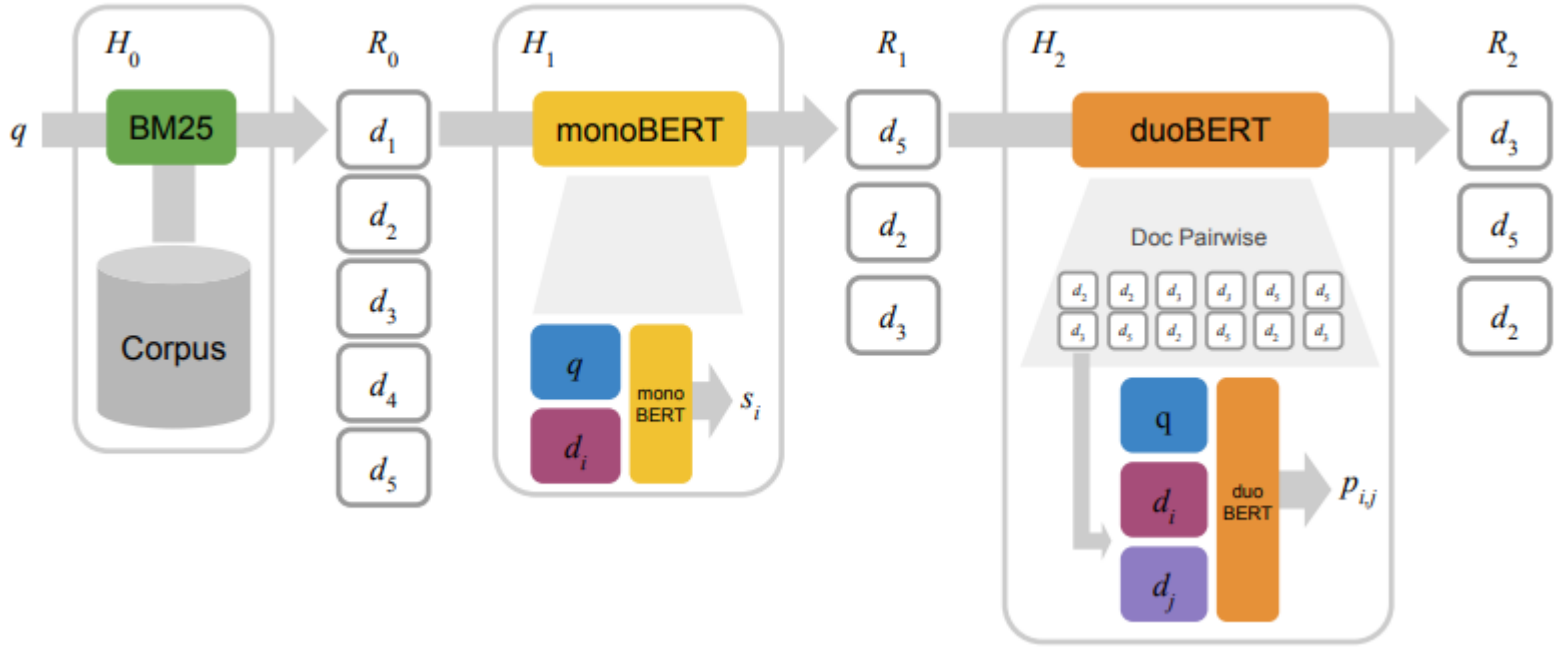
BERT ET AL.

Multi-Stage Document Ranking with BERT

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, Jimmy Lin

The advent of deep neural networks pre-trained via language modeling tasks has spurred a number of successful applications in natural language processing. This work explores one such popular model, BERT, in the context of document ranking. We propose two variants, called monoBERT and duoBERT, that formulate the ranking problem as pointwise and pairwise classification, respectively. These two models are arranged in a multi-stage ranking architecture to form an end-to-end search system. One major advantage of this design is the ability to trade off quality against latency by controlling the admission of candidates into each pipeline stage, and by doing so, we are able to find operating points that offer a good balance between these two competing metrics. On two large-scale datasets, MS MARCO and TREC CAR, experiments show that our model produces results that are either at or comparable to the state of the art. Ablation studies show the contributions of each component and characterize the latency/quality tradeoff space.

Subjects: **Information Retrieval (cs.IR)**; Machine Learning (cs.LG)
Cite as: arXiv:1910.14424 **[cs.IR]**
(or arXiv:1910.14424v1 **[cs.IR]** for this version)
https://doi.org/10.48550/arXiv.1910.14424



BERT ET AL.

Passage Re-ranking with BERT

Rodrigo Nogueira, Kyunghyun Cho

Recently, neural models pretrained on a language modeling task, such as ELMo (Peters et al., 2017), OpenAI GPT (Radford et al., 2018), and BERT (Devlin et al., 2018), have achieved impressive results on various natural language processing tasks such as question-answering and natural language inference. In this paper, we describe a simple re-implementation of BERT for query-based passage re-ranking. Our system is the state of the art on the TREC-CAR dataset and the top entry in the leaderboard of the MS MARCO passage retrieval task, outperforming the previous state of the art by 27% (relative) in MRR@10. The code to reproduce our results is available at this https URL

Subjects: **Information Retrieval (cs.IR)**; Computation and Language (cs.CL); Machine Learning (cs.LG)
Cite as: arXiv:1901.04085 **[cs.IR]**
(or arXiv:1901.04085v5 **[cs.IR]** for this version)
https://doi.org/10.48550/arXiv.1901.04085

ÉVALUATION

Données

MS MARCO: A Human Generated MAchine Reading COmprehension Dataset

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang
Microsoft AI & Research

Field	Description
Query	A question query issued to Bing.
Passages	Top 10 passages from Web documents as retrieved by Bing. The passages are presented in ranked order to human editors. The passage that the editor uses to compose the answer is annotated as is_selected: 1.
Document URLs	URLs of the top ranked documents for the question from Bing. The passages are extracted from these documents.
Answer(s)	Answers composed by human editors for the question, automatically extracted passages and their corresponding documents.
Well Formed Answer(s)	Well-formed answer rewritten by human editors, and the original answer.
Segment	QA classification. E.g., tallest mountain in south america belongs to the ENTITY segment because the answer is an entity (Aconcagua).

TREC Complex Answer Retrieval Overview

Laura Dietz*, Ben Gamari, Jeff Dalton, Nick Craswell

Homepage: `http://trec-car.cs.unh.edu`
Google group mailinglist: `TREC-CAR`

Introduced by Dietz et al. (2017), in this dataset, the input query is the concatenation of a Wikipedia article title with the title of one of its section. The relevant passages are the paragraphs within that section. The corpus consists of all of the English Wikipedia paragraphs, except the abstracts. The released dataset has five predefined folds, and we use the first four as a training set (approximately 2.3M queries), and the remaining as a validation set (approximately 580k queries). The test set is the same one used to evaluate the submissions to TREC-CAR 2017 (approx. 2,254 queries).

Although TREC-CAR 2017 organizers provide manual annotations for the test set, only the top five passages retrieved by the systems submitted to the competition have manual annotations. This means that true relevant passages are not annotated if they rank low. Hence, we evaluate using the automatic annotations, which provide relevance scores for all possible query-passage pairs.

ÉVALUATION

Les métriques

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\mathrm{rank}_i}.$$

Q : ensemble des requêtes

$Rank_i$: rank position of the *first* relevant document for the *i*-th query.

$$\mathbf{MAP} = \frac{\sum_{q=1}^{Q}\mathbf{AveP(q)}}{Q}$$

Sur l'ensemble des documents proposés

Précision k 1er documents

1 si pertinent, sinon 0

$$\mathbf{AveP} = \frac{\sum_{k=1}^{n} P(k) \times \mathbf{rel}(k)}{\mathbf{number\ of\ relevant\ documents}}$$

Source

Sommes nous certains que nous évaluons des performances utiles pour un moteur de recherche ?

Weaviate

Moteur de recherche vectoriel

MOTEUR DE RECHERCHE VECTORIEL

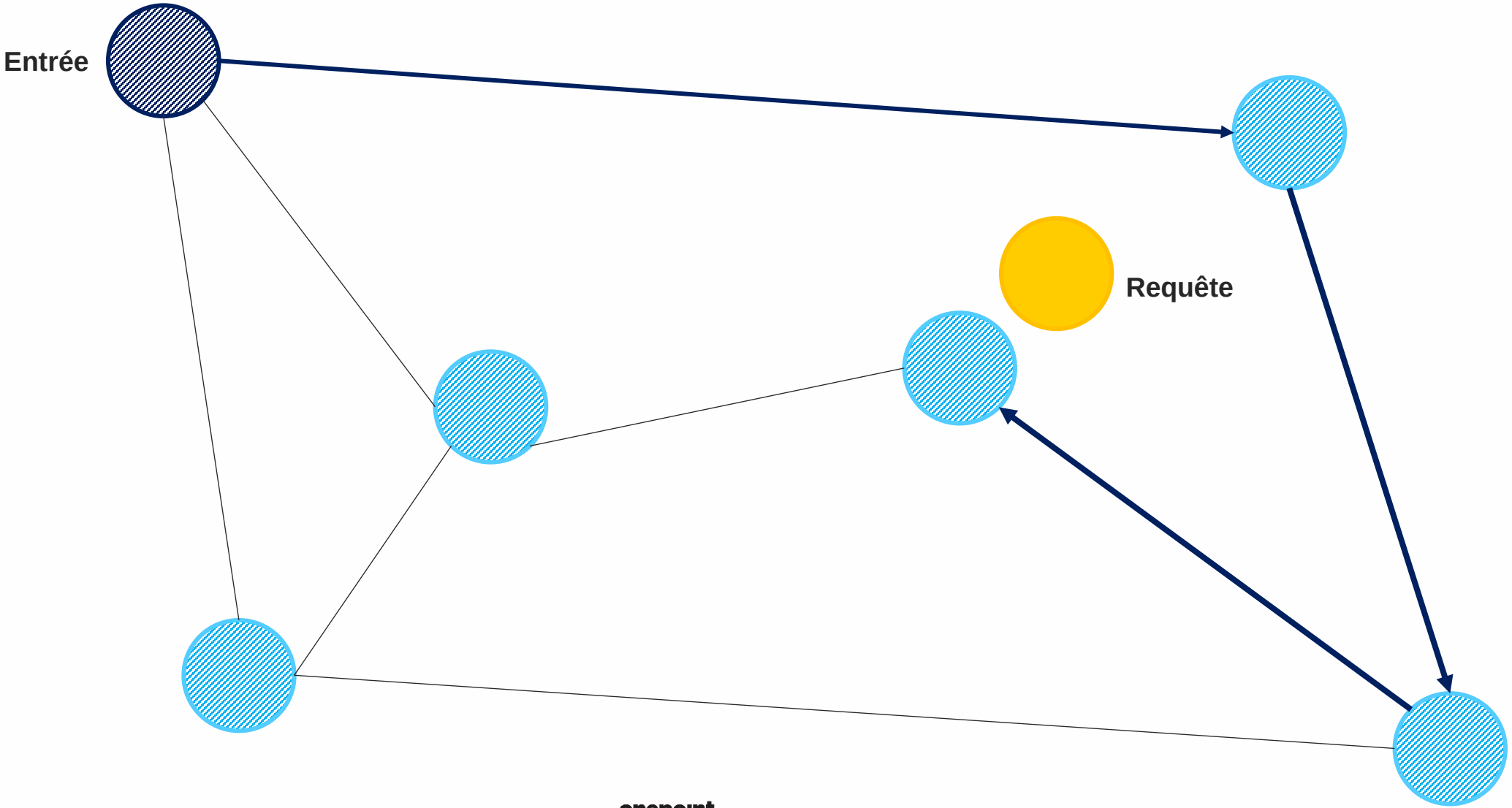
Une autre manière d’utiliser BERT

- Les passages des documents sont convertis en vecteurs numériques
- Les requêtes sont converties en vecteurs numériques, à la volée
- On cherche les K vecteurs numériques les plus proches du vecteur numérique obtenu à partir de la requête
- Dans certain cas une recherche complète est possible (si le nombre de passages stockés ne dépassent pas les quelques centaines de milliers)
- Sinon on se contente d’une approximation des K plus proches voisins
- Les vecteurs numériques sont stockés sous forme d’un graphe : la structure de ce graphe est conçu pour améliorer la rapidité et la qualité de l’approximation des K plus proches voisins

APPROXIMATE NEAREST NEIGHBORS SEARCH

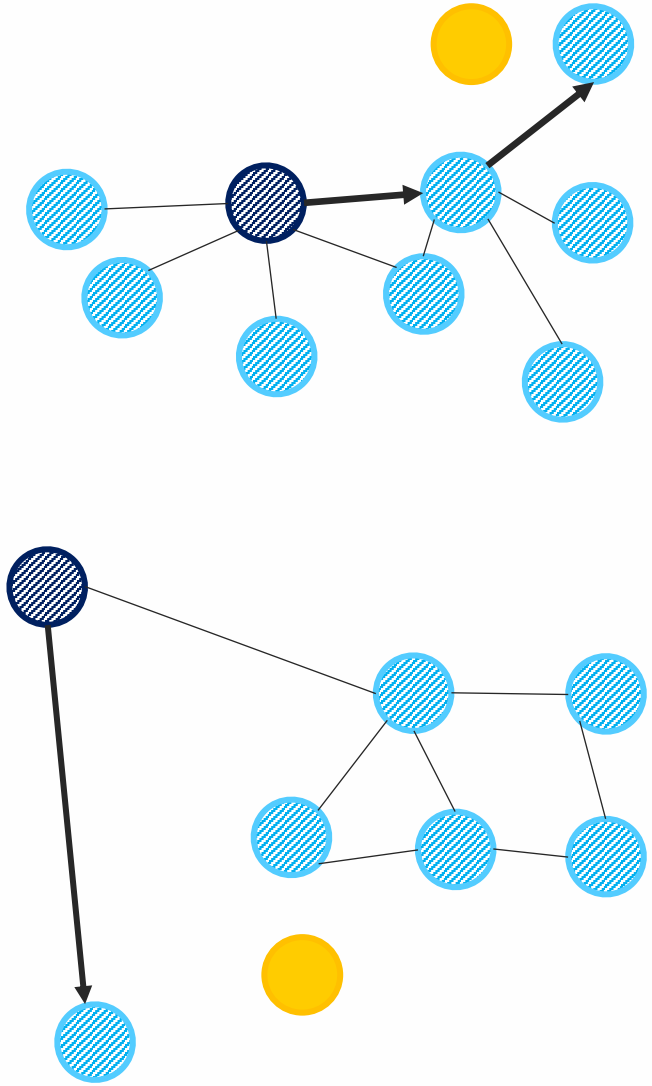
- Impossible de parcourir l'ensemble des données
- Recherche d'une approximation
- Ils existent différentes familles de solutions mais elles peuvent ne pas maintenir leurs performances :
  - si la dimension de l'espace vectoriel est petite
  - si les vecteurs forment des clusters

SMALL NAVIGABLE WORD



SMALL NAVIGABLE WORD

- Graphe navigable : le nombre de nœuds augmente de manière logarithmique ou polylogarithmique
- Procédure de construction :
  - Les éléments sont sélectionnés aléatoirement
  - Chaque élément est connecté aux M plus proches voisins parmi les éléments insérés précédemment
- Algorithme de recherche : algorithme glouton => à chaque étape il choisi le voisin le plus proche de la requête
- Lorsque le degré des nœuds augmente : le temps de recherche augmente
- Lorsque le degré des nœuds diminue : risque de tomber dans une impasse



HIERARCHICAL NAVIGABLE SMALL WORLD

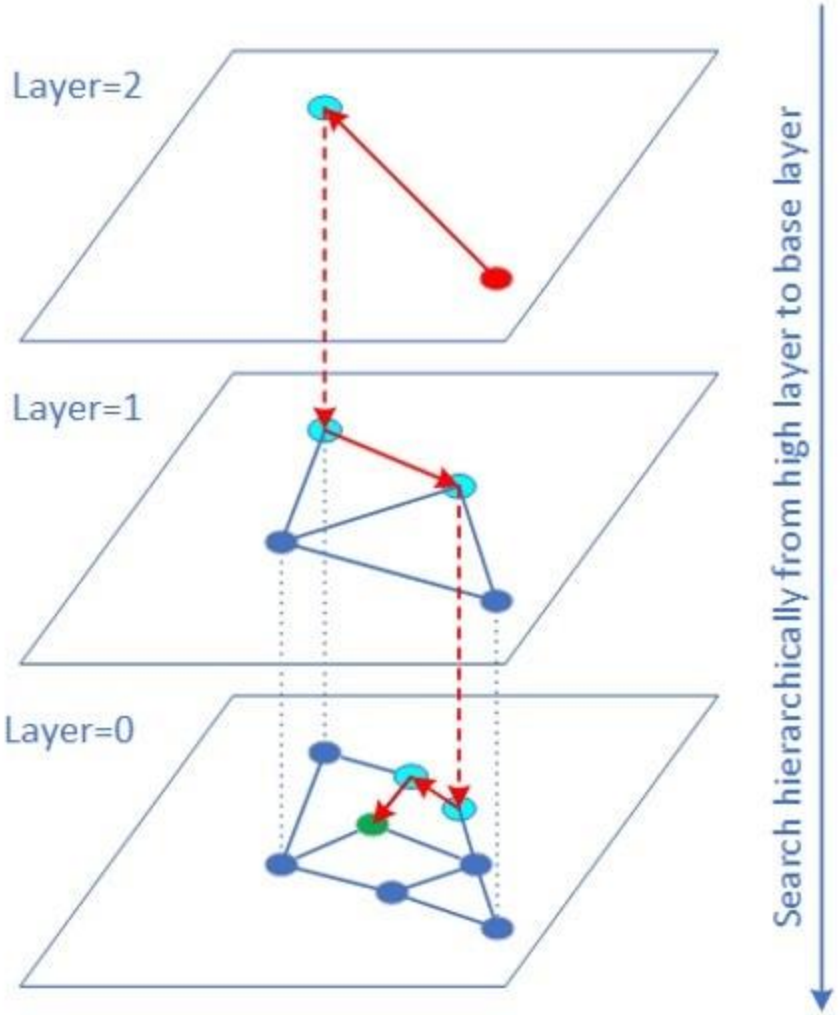
Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs

Yu. A. Malkov, D. A. Yashunin

We present a new approach for the approximate K-nearest neighbor search based on navigable small world graphs with controllable hierarchy (Hierarchical NSW, HNSW). The proposed solution is fully graph-based, without any need for additional search structures, which are typically used at the coarse search stage of the most proximity graph techniques. Hierarchical NSW incrementally builds a multi-layer structure consisting from hierarchical set of proximity graphs (layers) for nested subsets of the stored elements. The maximum layer in which an element is present is selected randomly with an exponentially decaying probability distribution. This allows producing graphs similar to the previously studied Navigable Small World (NSW) structures while additionally having the links separated by their characteristic distance scales. Starting search from the upper layer together with utilizing the scale separation boosts the performance compared to NSW and allows a logarithmic complexity scaling. Additional employment of a heuristic for selecting proximity graph neighbors significantly increases performance at high recall and in case of highly clustered data. Performance evaluation has demonstrated that the proposed general metric space search index is able to strongly outperform previous opensource state-of-the-art vector-only approaches. Similarity of the algorithm to the skip list structure allows straightforward balanced distributed implementation.

Comments: 13 pages, 15 figures
Subjects: **Data Structures and Algorithms (cs.DS)**; Computer Vision and Pattern Recognition (cs.CV); Information Retrieval (cs.IR); Social and Information Networks (cs.SI)
Cite as: arXiv:1603.09320 **[cs.DS]**
(or arXiv:1603.09320v4 **[cs.DS]** for this version)
https://doi.org/10.48550/arXiv.1603.09320

HIERARCHICAL NAVIGABLE SMALL WORLD



Source : Deep Learning with Intel® AVX512 and Intel® Deep Learning Boost Tuning Guide on 3rd Generation Intel® Xeon® Scalable Processors