SCIENCES SORBONNE UNIVERSITÉ

RECHERCHE D'INFORMATION ET DEEP LEARNING

Laure Soulier

LIP6

MLIA Machine Learning & Deep Learning for Information Access

Plan

1 Et si la RI c'était du Machine Learning ?

2 Deep Learning pour la RI

3 Conclusion

Et si la RI c'était du Machine Learning ?

Formulation du problème d'apprentissage

- Recueillir un corpus d'entraînement de triplets (requête, document, pertinence)
- La pertinence r est binaire (mais peut être multi-classes, avec 3-7 valeurs)
- Le document est représenté par un vecteur de caractéristique $x = (x_1, ..., x_n)$.
- Former un modèle d'apprentissage machine pour prédire la classe r d'une paire document-requête

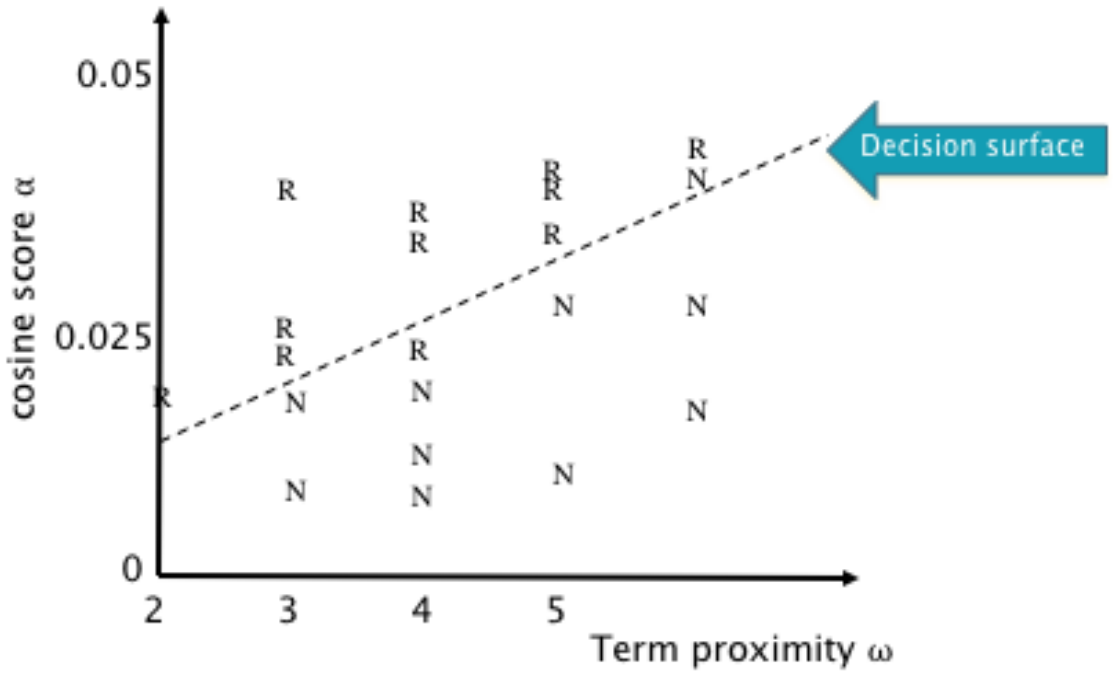
example	docID	query	cosine score	ω	judgment
Φ_1	37	linux operating system	0.032	3	*relevant*
Φ_2	37	penguin logo	0.02	4	*nonrelevant*
Φ_3	238	operating system	0.043	2	*relevant*
Φ_4	238	runtime environment	0.004	2	*nonrelevant*
Φ_5	1741	kernel layer	0.022	3	*relevant*
Φ_6	2094	device driver	0.03	2	*relevant*
Φ_7	3191	device driver	0.027	5	*nonrelevant*

Exemple de caractéristiques des documents

ID	Feature Description	Category
1	$\sum_{q_i \in q \cap d} c(q_i, d)$ in body	Q-D
2	$\sum_{q_i \in q \cap d} c(q_i, d)$ in anchor	Q-D
3	$\sum_{q_i \in q \cap d} c(q_i, d)$ in title	Q-D
4	$\sum_{q_i \in q \cap d} c(q_i, d)$ in URL	Q-D
5	$\sum_{q_i \in q \cap d} c(q_i, d)$ in whole document	Q-D
6	$\sum_{q_i \in q} idf(q_i)$ in body	Q
7	$\sum_{q_i \in q} idf(q_i)$ in anchor	Q
8	$\sum_{q_i \in q} idf(q_i)$ in title	Q
9	$\sum_{q_i \in q} idf(q_i)$ in URL	Q
10	$\sum_{q_i \in q} idf(q_i)$ in whole document	Q
11	$\sum_{q_i \in q \cap d} c(q_i, d) \cdot idf(q_i)$ in body	Q-D
12	$\sum_{q_i \in q \cap d} c(q_i, d) \cdot idf(q_i)$ in anchor	Q-D
13	$\sum_{q_i \in q \cap d} c(q_i, d) \cdot idf(q_i)$ in title	Q-D
14	$\sum_{q_i \in q \cap d} c(q_i, d) \cdot idf(q_i)$ in URL	Q-D
15	$\sum_{q_i \in q \cap d} c(q_i, d) \cdot idf(q_i)$ in whole document	Q-D
16	$\|d\|$ of body	D
17	$\|d\|$ of anchor	D
18	$\|d\|$ of title	D
19	$\|d\|$ of URL	D
20	$\|d\|$ of whole document	D
21	BM25 of body	Q-D
22	BM25 of anchor	Q-D
23	BM25 of title	Q-D
24	BM25 of URL	Q-D
25	BM25 of whole document	Q-D
26	LMIR.ABS of body	Q-D
27	LMIR.ABS of anchor	Q-D
28	LMIR.ABS of title	Q-D
29	LMIR.ABS of URL	Q-D
30	LMIR.ABS of whole document	Q-D
31	LMIR.DIR of body	Q-D
32	LMIR.DIR of anchor	Q-D
33	LMIR.DIR of title	Q-D
34	LMIR.DIR of URL	Q-D
35	LMIR.DIR of whole document	Q-D
36	LMIR.JM of body	Q-D
37	LMIR.JM of anchor	Q-D
38	LMIR.JM of title	Q-D
39	LMIR.JM of URL	Q-D
40	LMIR.JM of whole document	Q-D
41	Sitemap based term propagation	Q-D
42	Sitemap based score propagation	Q-D
43	Hyperlink based score propagation: weighted in-link	Q-D
44	Hyperlink based score propagation: weighted out-link	Q-D
45	Hyperlink based score propagation: uniform out-link	Q-D
46	Hyperlink based propagation: weighted in-link	Q-D
47	Hyperlink based feature propagation: weighted out-link	Q-D
48	Hyperlink based feature propagation: uniform out-link	Q-D
49	HITS authority	Q-D
50	HITS hub	Q-D

Formulation du problème d'apprentissage

- Qu'apprend-on ?
  - Une fonction de score ?
  - Classification, Regression, ...



Mais...

Pourquoi n'est-ce pas adapté à un problème de RI ?

Formulation du problème d'apprentissage

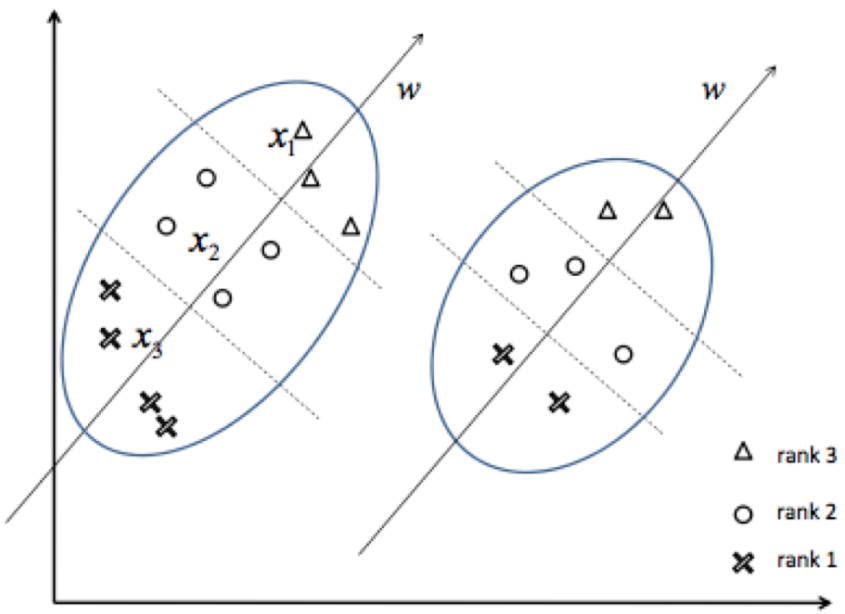
- Apprendre un ordre total sur un ensemble d'articles X induit que :
  - Cet ordre permettra de comparer tous les couples d'articles dans X
  - Compte tenu de cet ordre total, n'importe quel sous-ensemble de X peut être ordonné.

Méthode

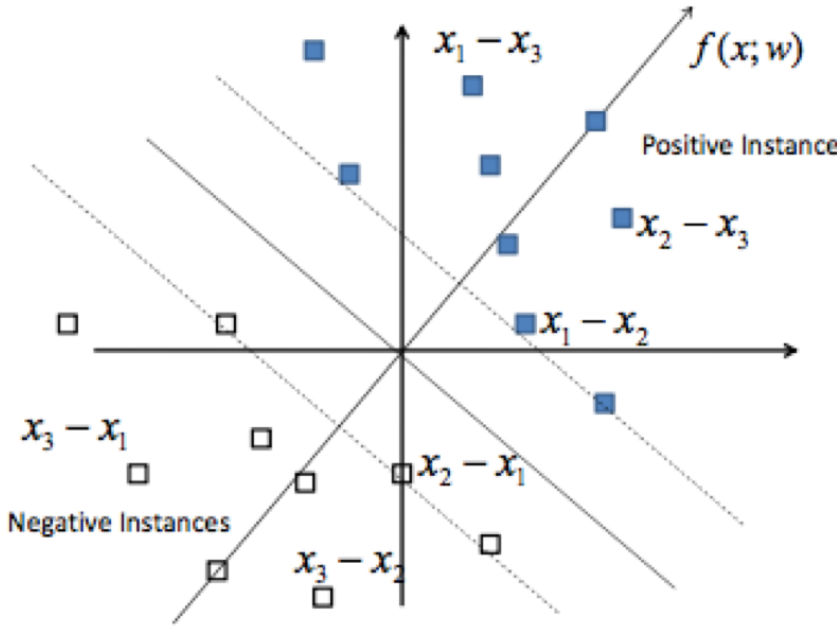
- Apprendre à ordonner des paires d'exemples : Une erreur se produit lorsque deux éléments sont mal ordonnés.
  → Seuls les scores relatifs d'une paire sont importants.
- Il n'est pas nécessaire d'ordonner toutes les paires : seul un petit sous-ensemble des exemples est nécessaire.
  → Ceci fournira un ordre partiel sur les éléments de X
- Une fonction d'ordonnancement f sera apprise sur cet ensemble partiellement ordonné.
  → Elle permettra ensuite d'étendre l'ordre partiel à un ordre total sur tous les éléments

Catégorisation des méthodes d'apprentissage d'ordonnancement

- Ordonnancement (plusieurs classes ordonnées)



- Classification (classes binaires : pertinent / non pertinent)



Mais en fait... Ordonnancement = classification

- L'ordonnancement est souvent formulé comme un problème de classement sur des paires cruciales :
  - Pour chaque couple (x, x'), deux exemples sont créés.
  - On peut déduire : $(x' - x, +1)$ et $(x - x', -1)$
  - Rappel: f est linéaire
- N'importe quel algorithme de classification peut être utilisé sur ce nouvel ensemble

Catégorisation des méthodes d'apprentissage d'ordonnancement

- Pointwise
  - Estimer une note ou un label (pertinent ou non) pour un seul document
  - Les documents sont indépendants (classification, ...)
- Paiwise
  - Estimation d'une préférence d'ordre partiel dans les paires de documents
  - Documents dépendants au sein de la paire
  - Les paires de documents sont indépendantes
- Listwise
  - Estimer l'ordre total dans une liste de documents
  - Les scores des documents dépendent les uns des autres

Catégorisation des méthodes d'apprentissage d'ordonnancement

SCIENCES SORBONNE UNIVERSITÉ

Category	Algorithms
Pointwise Approach	**Regression**: Least Square Retrieval Function (TOIS 1989), Regression Tree for Ordinal Class Prediction (Fundamenta Informaticae, 2000), Subset Ranking using Regression (COLT 2006), ... **Classification**: Discriminative model for IR (SIGIR 2004), McRank (NIPS 2007), ... **Ordinal regression**: Pranking (NIPS 2002), OAP-BPM (EMCL 2003), Ranking with Large Margin Principles (NIPS 2002), Constraint Ordinal Regression (ICML 2005), ...
Pairwise Approach	Learning to Retrieve Information (SCC 1995), Learning to Order Things (NIPS 1998), Ranking SVM (ICANN 1999), RankBoost (JMLR 2003), LDM (SIGIR 2005), RankNet (ICML 2005), Frank (SIGIR 2007), MHR(SIGIR 2007), GBRank (SIGIR 2007), QBRank (NIPS 2007), MPRank (ICML 2007), IRSVM (SIGIR 2006), LambdaRank (NIPS 2006),...
Listwise Approach	**Non-measure specific**: ListNet (ICML 2007), ListMLE (ICML 2008), BoltzRank (ICML 2009) ... **Measure-specific**: AdaRank (SIGIR 2007), SVM-MAP (SIGIR 2007), SoftRank (LR4IR 2007), RankGP (LR4IR 2007), ...

Figure 1: source : `https://www.programmersought.com/article/188249556/`

Deep Learning pour la RI

Vers des moteurs de recherche modernes

- Approche traditionnelle

$$RSV(q, d) \tag{1}$$

- Approche moderne (guidée par l'apprentissage machine)

$$f(q, d) \tag{2}$$

Les enjeux

- Effectuer des recherches et comprendre les documents
- Représenter les requêtes et documents sous forme de vecteurs à caractéristiques multiples
- Calcul de plusieurs scores de correspondance entre la requête et le document
- Classement d'entraînement avec des scores correspondants en tant que caractéristiques utilisant l'apprentissage d'ordonnancement

Du lexique vers la sémantique

Query: united states president

The President of the United States of America (POTUS) is the elected head of state and head of government of the United States. The president leads the executive branch of the federal government and is the commander in chief of the United States Armed Forces. Barack Hussein Obama II (born August 4, 1961) is an American politician who is the 44th and current President of the United States. He is the first African American to hold the office and the first president born outside the continental United States.

The President of the United States of America (POTUS) is the elected head of state and head of government of the United States. The president leads the executive branch of the federal government and is the commander in chief of the United States Armed Forces. Barack Hussein Obama II (born August 4, 1961) is an American politician who is the 44th and current President of the United States. He is the first African American to hold the office and the first president born outside the continental United States.

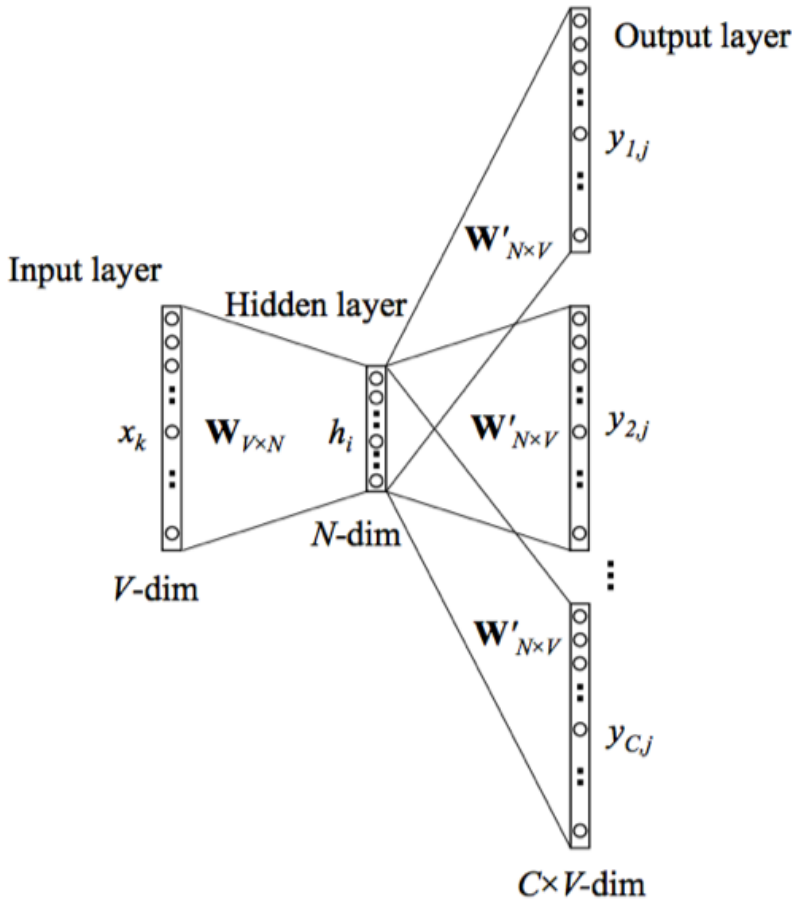
Traditional IR models estimate relevance based on lexical matches of query terms in document

Representation learning based models garner evidence of relevance from all document terms based on semantic matches with query

Both lexical and semantic matching are important and can be modelled with **neural networks**

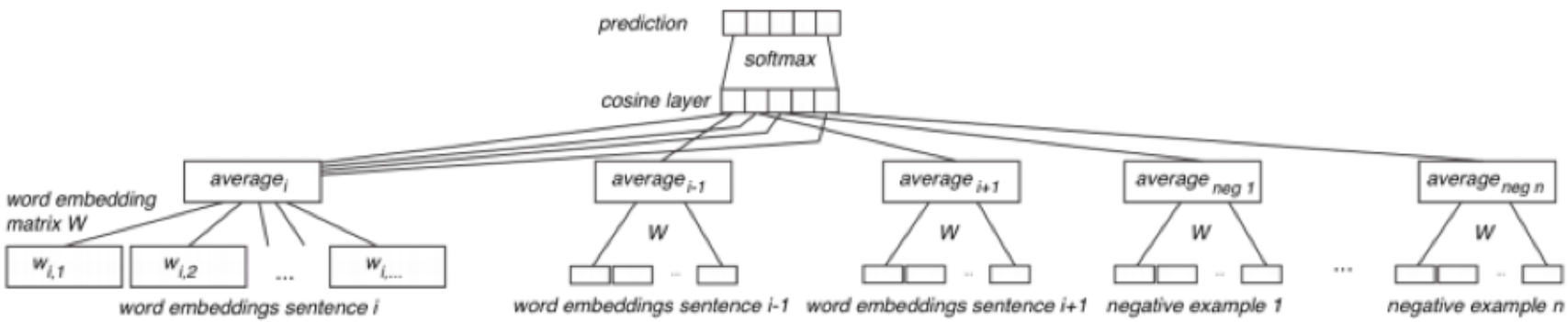
Deep learning pour la sémantique

- Approche pionnière pour capturer la sémantique des textes
  - Skip-gram & CBOW
  - Gloves, FastText, ...
  - Bert, ...
- Principe : Apprendre la représentation des mots à partir des informations du contexte
- Approches non supervisées



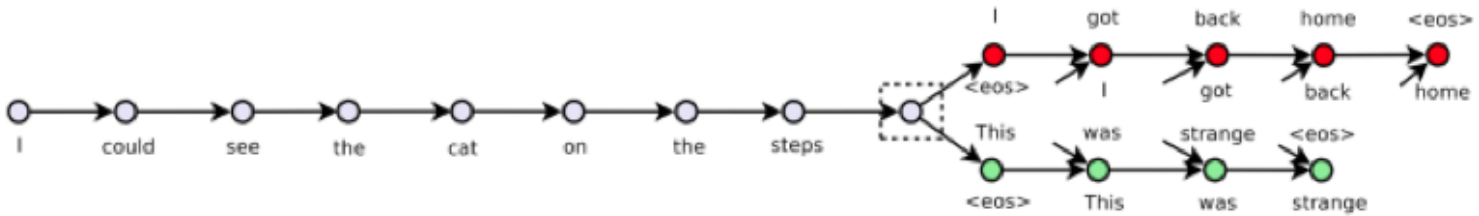
De la représentation des mots à la représentation des phrases

- Principe de compositionalité
  - Une phrase : Moyenne des représentations des mots
  - Éventuellement pondérées par TF-IDF
- Parfois guidé en fonction de la tâche

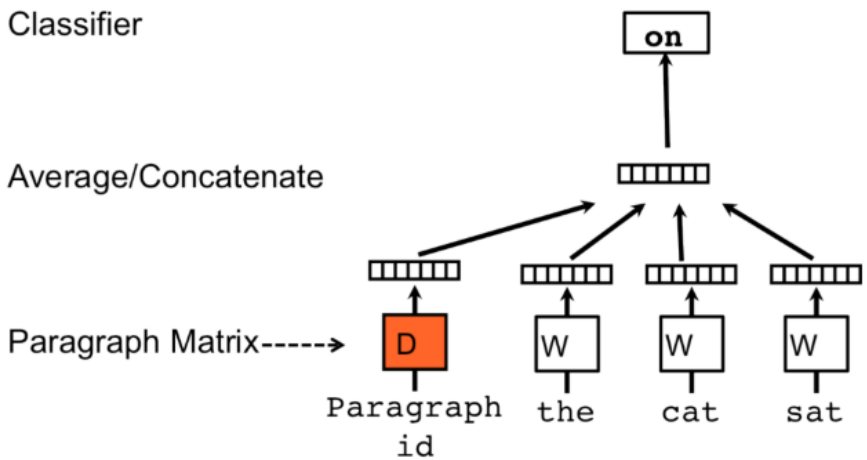


De la représentation des mots à la représentation des phrases

- Architecture encodeur-décodeur
- Phrase positionnée dans un contexte : phrases adjacentes



ParagraphVector



Le, Q. and Mikolov, T.2014. Distributed Representations of Sentences and Documents.

Sans oublier tous les travaux récents de modèle de langue

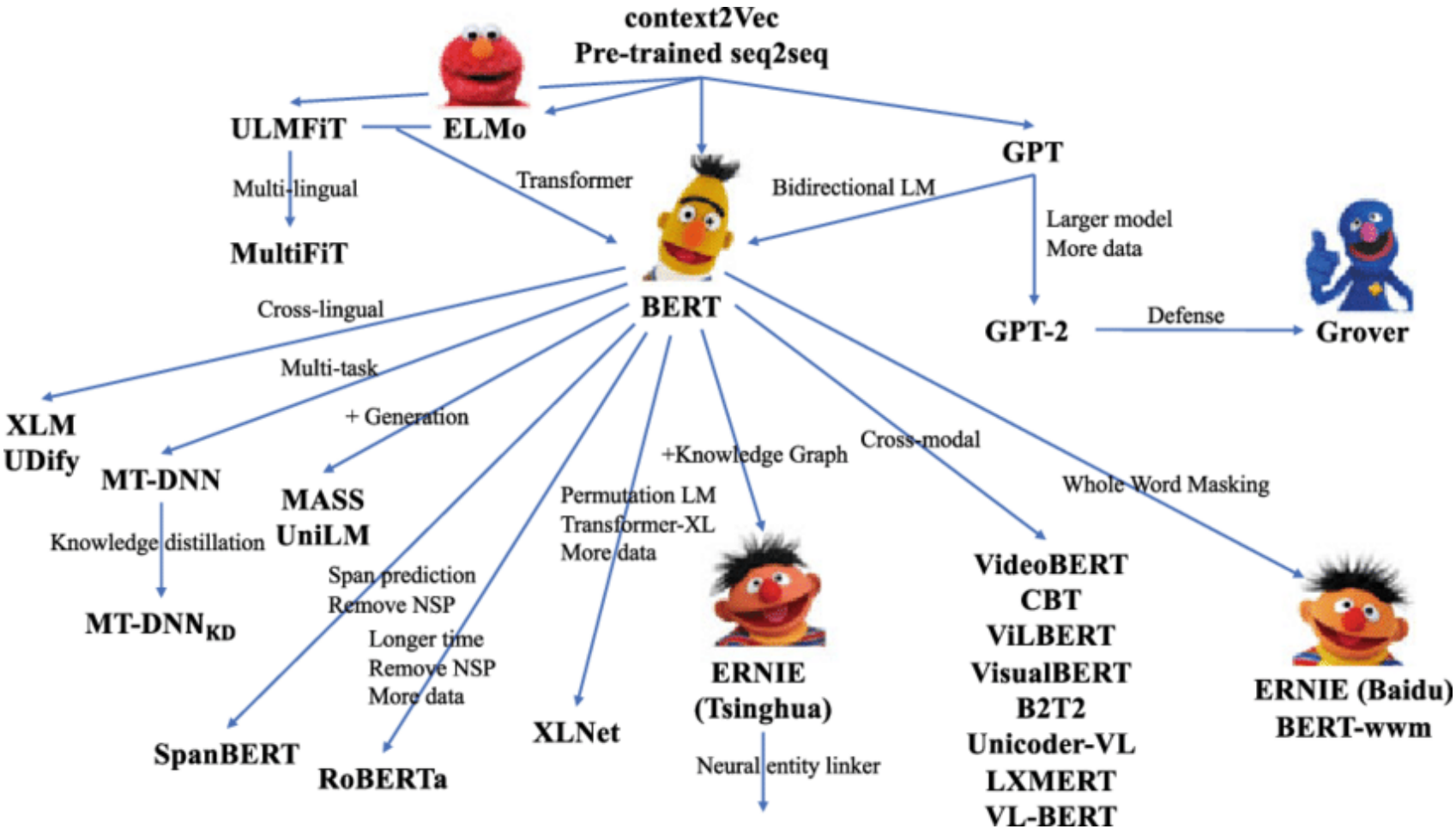
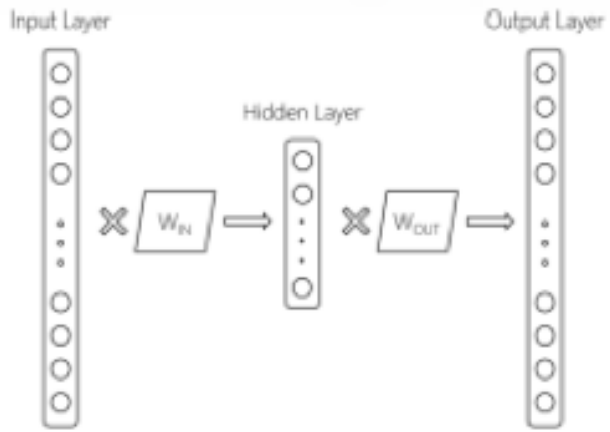


Figure 2: Source: https://www.researchgate.net/figure/The-Pre-trained-language-model-family_fig4_342684048

Comment cela fonctionne en RI ?

- Deux approches principales
  - Injecter les représentations dans les modèles traditionnels de RI
  - Apprendre de façon "end-to-end" les représentations ainsi que la fonction d'ordonnancement

Dual Embedding Space Model (DESM) [Nalisnick et al., 2016]



- Word2Vec optimises le produit des vecteurs en entrée et sortie pour capturer des co-occurrences statistiques sur le texte
- Et si on utilisait ces deux représentations (IN vs. OUT) ?
  - Les similitudes cosinus IN-IN et OUT-OUT sont élevées pour les mots qui sont similaires (sens commun)
  - Les similitudes cosinus IN-OUT sont élevées entre des mots qui se retrouvent souvent dans la même requête et le même document (thématique).

	yale	
IN-IN	OUT-OUT	IN-OUT
yale	yale	yale
harvard	uconn	faculty
nyu	harvard	alumni
cornell	tulane	orientation
tulane	nyu	haven
tufts	tufts	graduate

	seahawks	
IN-IN	OUT-OUT	IN-OUT
seahawks	seahawks	seahawks
49ers	broncos	highlights
broncos	49ers	jerseys
packers	nfl	tshirts
nfl	packers	seattle
steelers	steelers	hats

	eminem	
IN-IN	OUT-OUT	IN-OUT
eminem	eminem	eminem
rihanna	rihanna	rap
ludacris	dre	featuring
kanye	kanye	tracklist
beyonce	beyonce	diss
2pac	tupac	performs

Dual Embedding Space Model (DESM) [Nalisnick et al., 2016]

- Calcul de la similarité
  - Un document : vecteur centroïde des vecteurs OUT

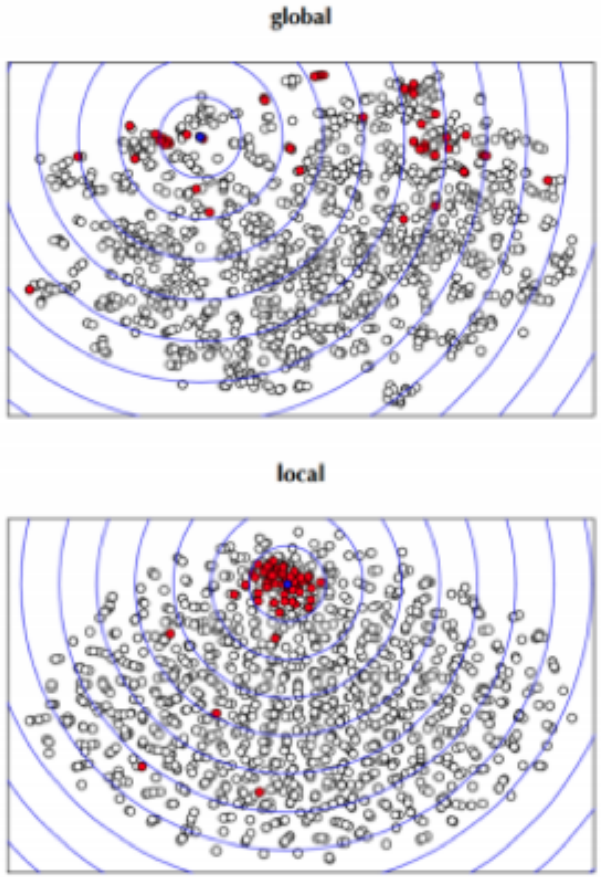
$$\vec{v}_{d,\text{OUT}} = \frac{1}{|d|} \sum_{t_d, \in d} \frac{\vec{v}_{t_{d,\text{OUT}}}}{\left\| \vec{v}_{t_{d,\text{OUT}}} \right\|} \tag{3}$$

  - Les similarités document-requête sont estimée à partir des similarités des vecteurs OUT des documents et IN de requêtes

$$RSV(q,d) = \frac{1}{q} \sum_{t_q \in q} \frac{\vec{v}_{t_{d,\text{OUT}}} * \vec{v}_{t_{q,\text{IN}}}}{\left\| \vec{v}_{t_{d,\text{OUT}}} \right\| \left\| \vec{v}_{t_{q,\text{IN}}} \right\|} \tag{4}$$

Expansion de requêtes [Roy et al., 2016]

- Exploitation de l'espace sémantique pour identifier les termes les plus proches
  - Pré-ordonnancement : plus proches voisins dans l'espace
  - Post-ordonnancement : restreindre le vocabulaire aux mots inclus dans les top N documents retournés par un système
- Exploitation de l'espace local vs. global



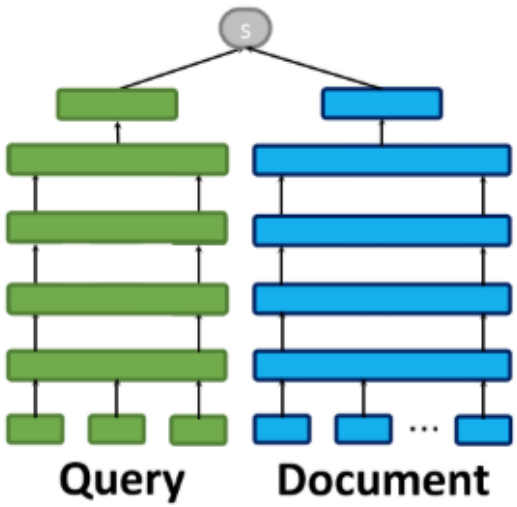
Vers des méthodes end-to-end

- Les représentations de documents/requêtes sont généralement apprises indépendamment de la tâche de récupération.
  → Bien conçu pour l'appariement sémantique
  → N'est pas spécialement conçu pour le calcul de la pertinence

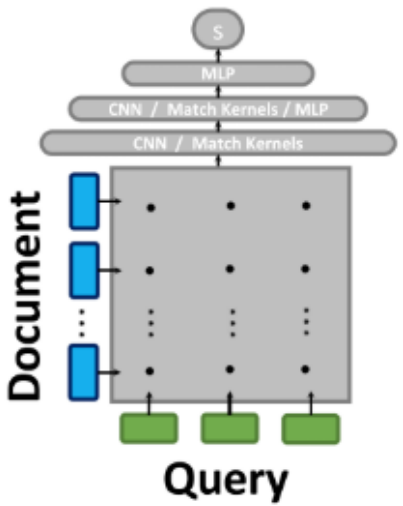
- Une solution consiste à guider la représentation en fonction de la tâche de récupération.
  → Objectif de la tâche de récupération : classement.
  → Besoin de construire un réseau neuronal capable de classer les documents (apprendre à classer).

Vers des méthodes end-to-end

- Modèles siamois (à gauche) : on apprend la représentation de la requête et du document en même temps qu'on apprend la fonction de score de similarité.
- Modèles basés sur l'interaction (à droite) : on pré-calcule une matrice d'interaction ayant pour but de calculer la similarités des termes de la requête et des documents (paires à paires) et de donner cette matrice en entrée au réseau de neurones pour apprendre le score de similarité.

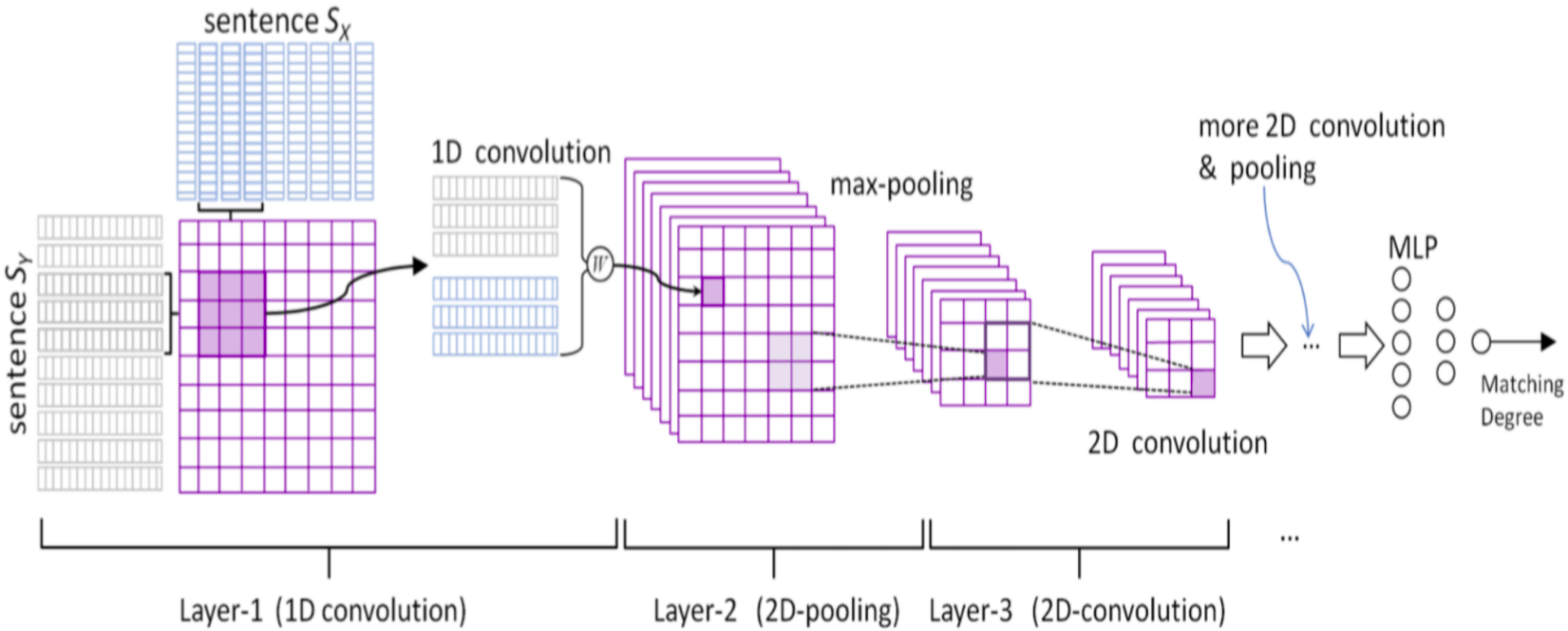


(a) Representation-based Similarity
(e.g., DSSM, SNRM)



(b) Query-Document Interaction
(e.g., DRMM, KNRM, Conv-KNRM)

Modèles siamois

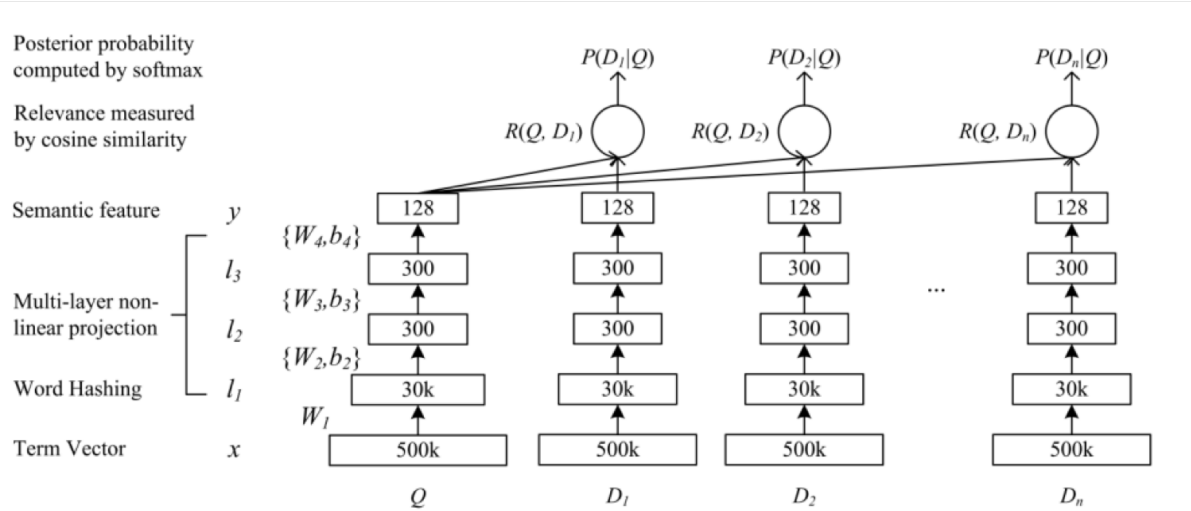


Matrice d'interaction *paramétrique*

Hu, B. et al. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. Advances in Neural Information Processing Systems 27.

Modèles siamois

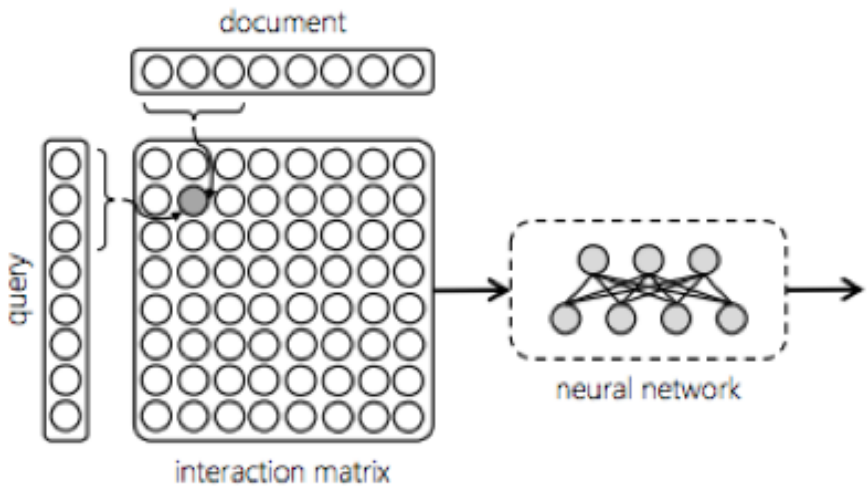
- Deep Semantic Matching Model (Huang et al, 2014)



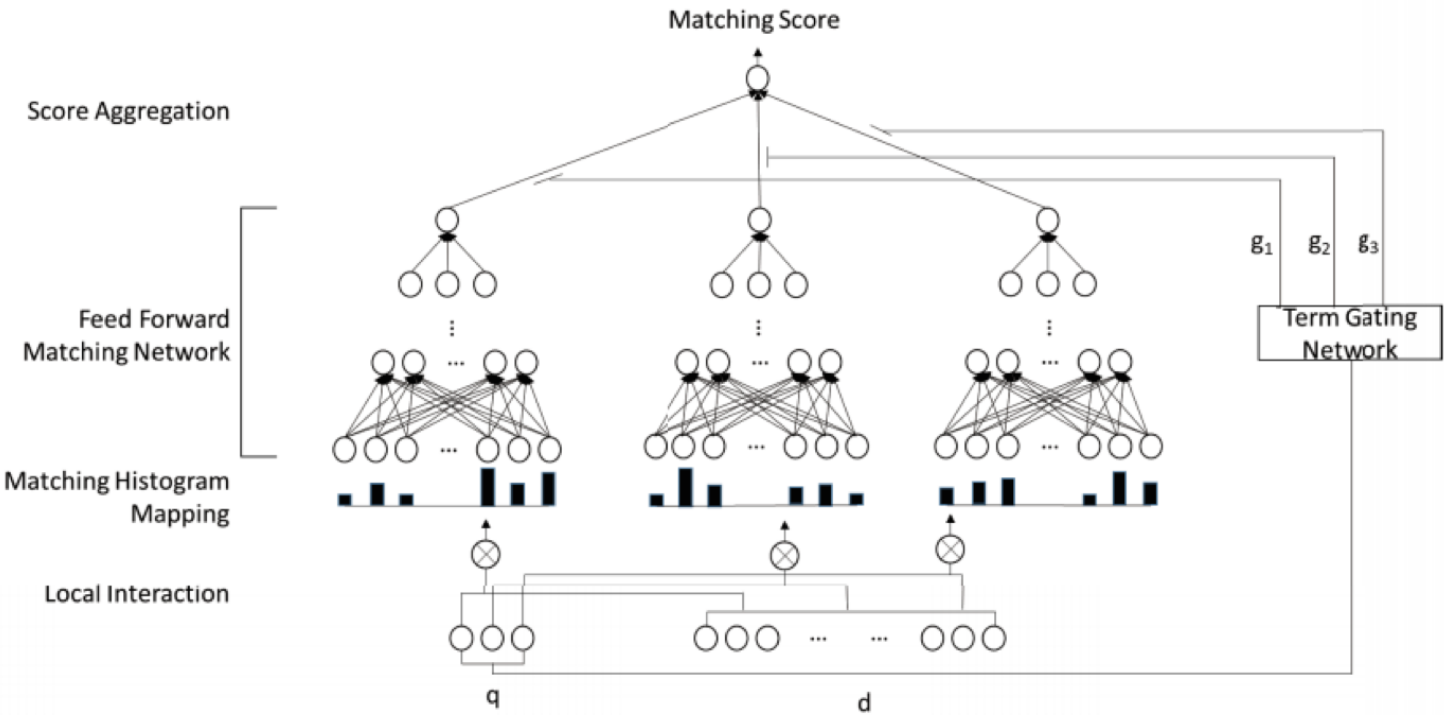
- On va calculer le score pour plusieurs paires de requête-document (la même requête):
  - 1 paire document pertinent-requête
  - N-1 paires de documents non pertinents-requête
- On va faire en sorte que le score du document pertinent soit supérieur à celui des documents non pertinents.
- Pour cela, les paires ont un facteur commun : la requête.
- Cela permet de capture les facteurs de similarité/pertinence en opposant des exemples négatifs.

Vers des méthodes end-to-end : les modèles basés sur l'interaction

- Modèles basés sur l'interaction (à droite) : on pré-calcule une matrice d'interaction ayant pour but de calculer la similarités des termes de la requête et des documents (paires à paires) et de donner cette matrice en entrée au réseau de neurones pour apprendre le score de similarité.

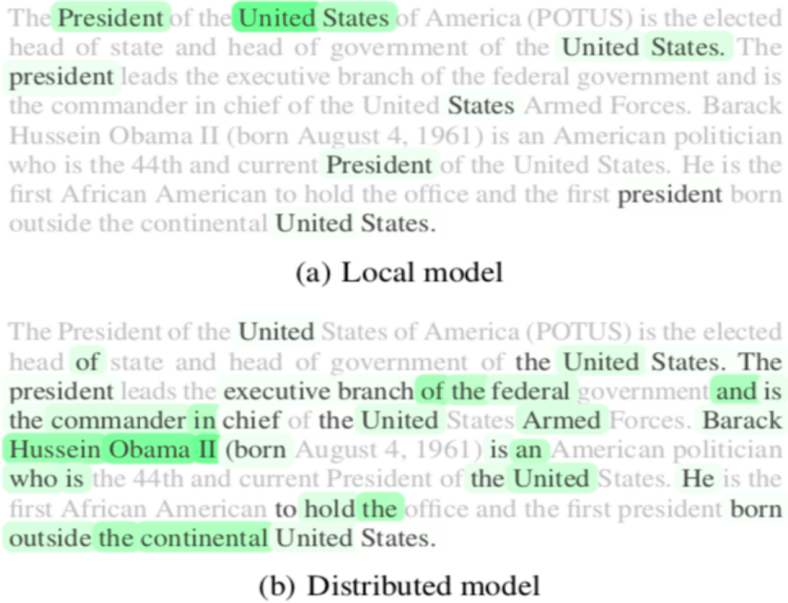


Deep Relevance Matching model [Guo et al., 2016]



- On transforme la matrice d'interaction en histogramme
- Les histogrammes correspondent à des vecteurs données en entrée du réseau

DUET - Combiner logal et global matching



Lissage thématique

Les modèles de langues

$$p(q_i|d) = p_{ML}(q_i|d) + p_{\text{topical}}(q_i|d)$$

où le second terme est plus "vague" (LDA, etc.)

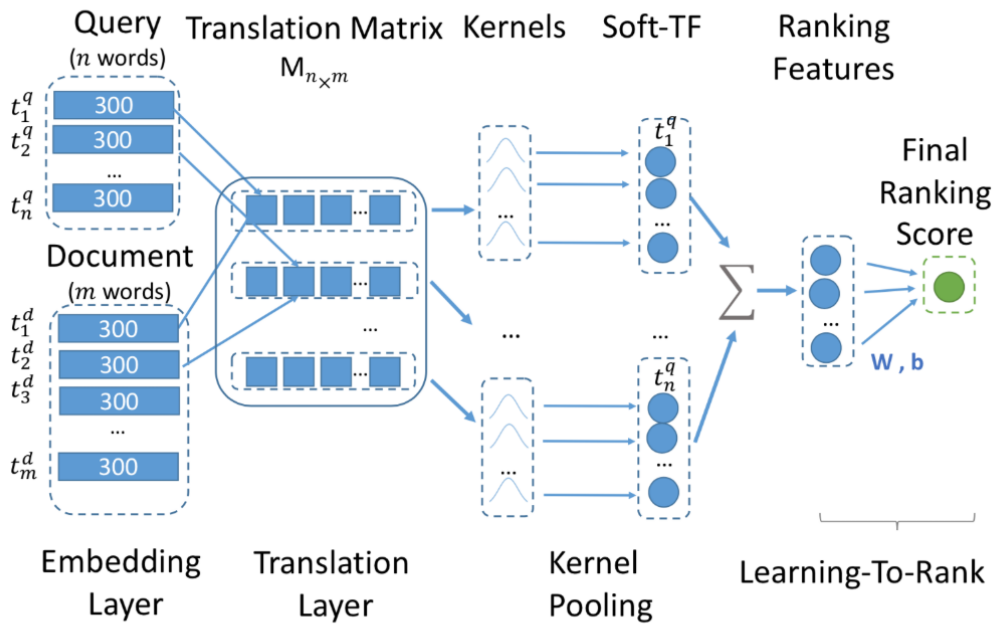
Combinaison de deux modèles :

1. (DSSM)
   Huang, P. et al. 2013. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data.
2. (MatchPyramid)
   Pang, L. et al. 2016. Text Matching as Image Recognition.

Mitra, B. et al. 2017. Learning to Match Using Local and Distributed Representations of Text for Web Search. Proceedings of the 26th International Conference on World Wide Web.

KNRM - Kernel Neural Relevance Model

☞ Un des meilleurs modèles d'interaction (avant BERT...)



Généralisation du DRMM

$$K_k(\underbrace{q_i \hat{d}}_{I_i}) = \exp\left(\frac{(I_{ij} - \mu_k)^2}{2\sigma_k^2}\right)$$

avec $\hat{d} \in \mathbb{R}^{n \times \ell}$ et $q_i \in \mathbb{R}^n$

Xiong, C. et al. 2017. End-to-End Neural Ad-Hoc Ranking with Kernel Pooling.

Et si on prenait un peu de recul...

- Résultats en RI

Data Set / Model	Robust04		$GOV2_{MQ2007}$		Sogou-Log	
	MAP	P@20	MAP	P@10	NDCG@1	NDCG@10
BM25[46] (1994)^{2,4}	0.255	0.370	0.450	0.366	0.142	0.287
QL[116] (1998)^{1,4}	0.253	0.369	\	\	0.126	0.282
RM3[117](2001)^{4}	0.287	0.377	\	\	\	\
RankSVM[118] (2002)^{2}	\	\	0.464	0.381	0.146	0.309
LambdaMart[97] (2010)^{2,4}	\	\	0.468	0.384	\	\
DSSM[13] (2013)$^{1,2}_{S/R/G}$	0.095	0.171	0.409	0.352	\	\
CDSSM[47] (2014)$^{1,2}_{S/R/G}$	0.067	0.125	0.364	0.291	0.144	0.333
ARC-I[17] (2014)$^{1,2}_{S/R/G}$	0.041	0.065	0.417	0.364	\	\
ARC-II[17] (2014)$^{1,2}_{S/I/G}$	0.067	0.128	0.421	0.366	\	\
MP[18] (2016)$^{1,2}_{S/I/G}$	0.189	0.290	0.434	0.371	0.218	0.380
Match-SRNN[69] (2016)$^{2}_{S/H/G}$	\	\	0.456	0.384	\	\
DRMM[21] (2016)$^{1,2,4}_{A/I/G}$	0.279	0.382	0.467	0.388	0.137	0.315
Duet[23] (2017)$^{3}_{A/H/G}$	\	\	0.474	0.398	\	\
DeepRank[33] (2017)$^{2}_{A/I/G}$	\	\	0.497	0.412	\	\
K-NRM[85] (2017)$^{4}_{A/I/G}$	\	\	\	\	0.264	0.428
SNRM[28] (2018)$^{5}_{S/R/G}$	0.286	0.377	\	\	\	\
SNRM+PRF[28] (2018)$^{5}_{S/R/G}$	0.297	0.395	\	\	\	\
CONV-KNRM[84] (2018)$^{4}_{A/I/M}$	\	\	\	\	0.336	0.481
HiNT[34] (2018)$^{3}_{A/I/G}$	\	\	0.502	0.418	\	\

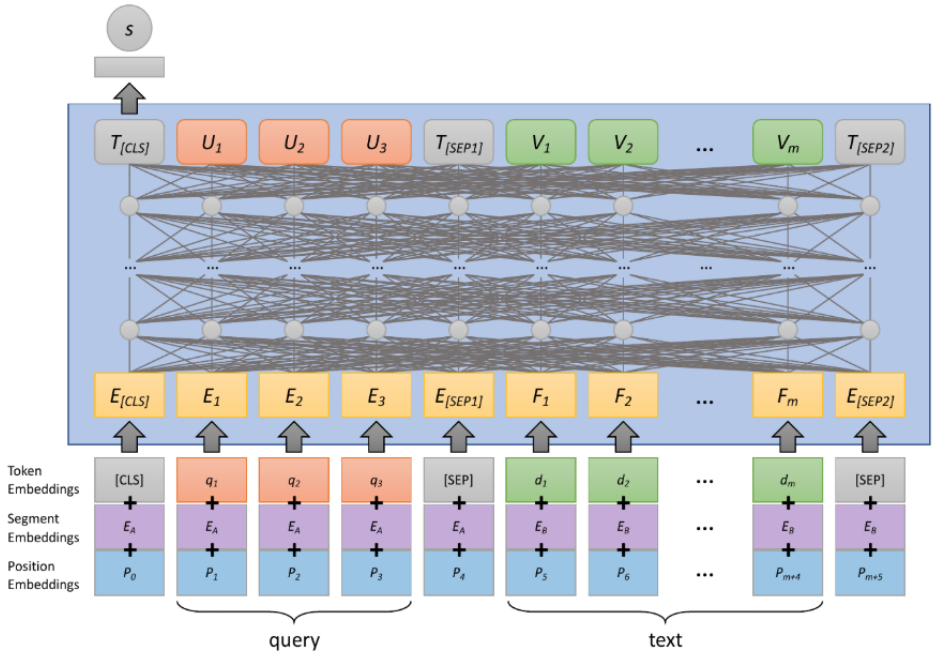
- BM25 fonctionne bien!
- Plus de données = c'est mieux
- On travaille beaucoup sur les textes courts (titres)

MonoBERT (2019)

SCIENCES SORBONNE UNIVERSITÉ

On utilise BERT en concaténant question et document cible

$$rsv(q, d) = x^{[\text{CLS}]}([\text{CLS}]\ \text{q}\ [\text{SEP}]\ \text{d})$$



Nogueira, R. and Cho, K. 2019. Passage Re-ranking with BERT.

- De très bons résultats
- Travaux récents permettent de réduire le temps de traitement (modèle EPIC de Macavaney et al. 2020 : expansion de document offline et ensuite cosine similarity entre les embeddings offline).

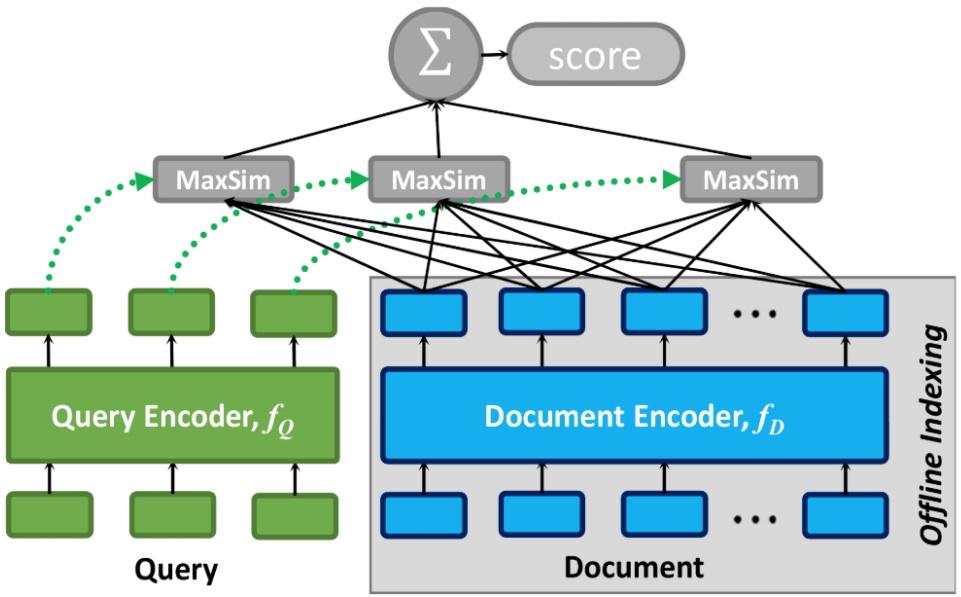
Et les résultats ?

	Method	TREC 2019 DL Passage		
		nDCG@10	MAP	Recall@1k
(3a)	BM25 (Anserini, $k = 1000$)	0.5058	0.3773	0.7389
(3b)	+ monoBERT$_{\text{Large}}$	0.7383	0.5058	0.7389
(4a)	BM25 + RM3 (Anserini, $k = 1000$)	0.5180	0.4270	0.7882
(4b)	+ monoBERT$_{\text{Large}}$	0.7421	0.5291	0.7882

ColBERT (2020)

ColBERT

- Encore plus simpole = Modèle d'interaction basé sur une aggrégation (maximum)
- permet de construire un index !



Khattab, O. and Zaharia, M. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT.

Comment apprendre ?

- But: mesurer la différence entre l'objectif et le comportement
- On calcule sur l'ensemble de la base d'apprentissage
  $\mathscr{D} = \sum_{x \in \mathscr{X}} \Delta(x, f_\theta(x))$
  où Δ mesure l'erreur par rapport à la sortie désirée en fonction de la tâche.
  et $\mathscr{X}$ = ensemble d'apprentissage

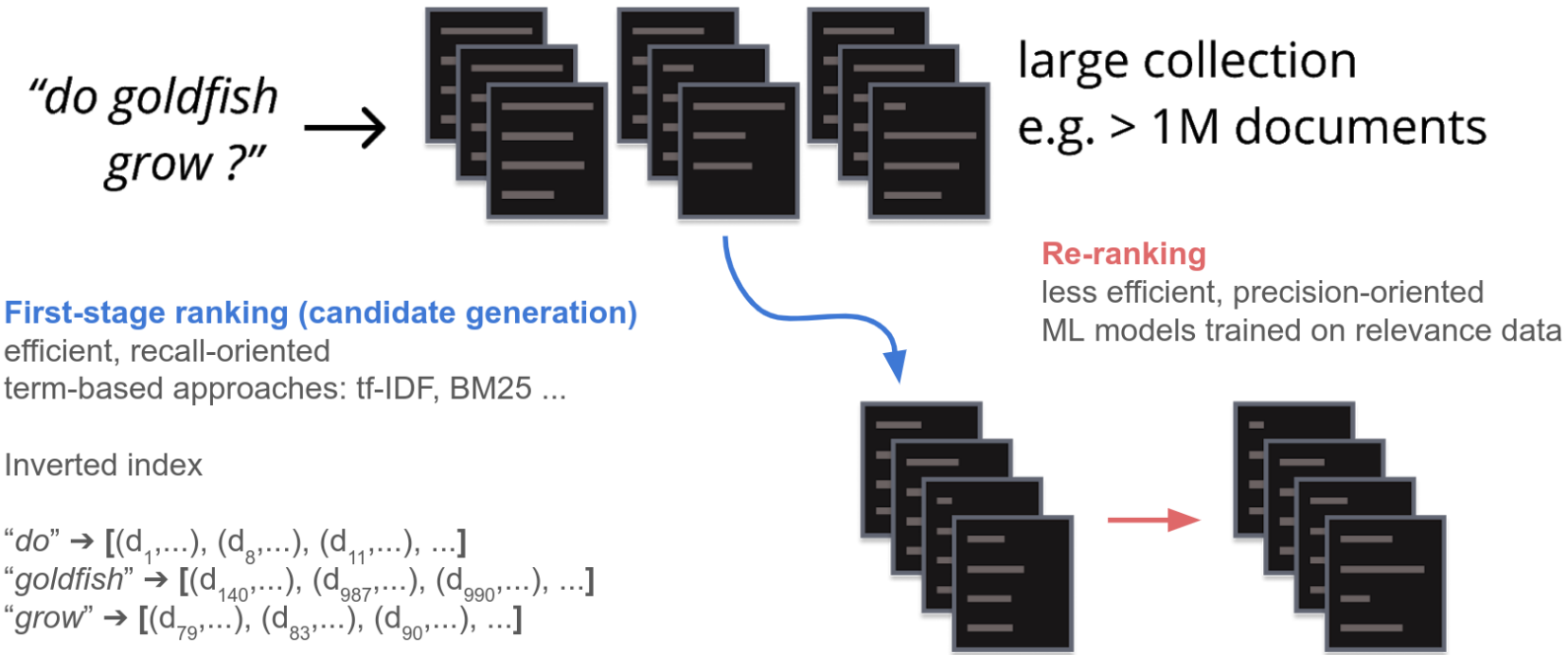
Les différentes fonctions de coût

- Regression : $\Delta(f_\theta(d), y') = \|y - y'\|^2$
- Classification : coût cross-entropique

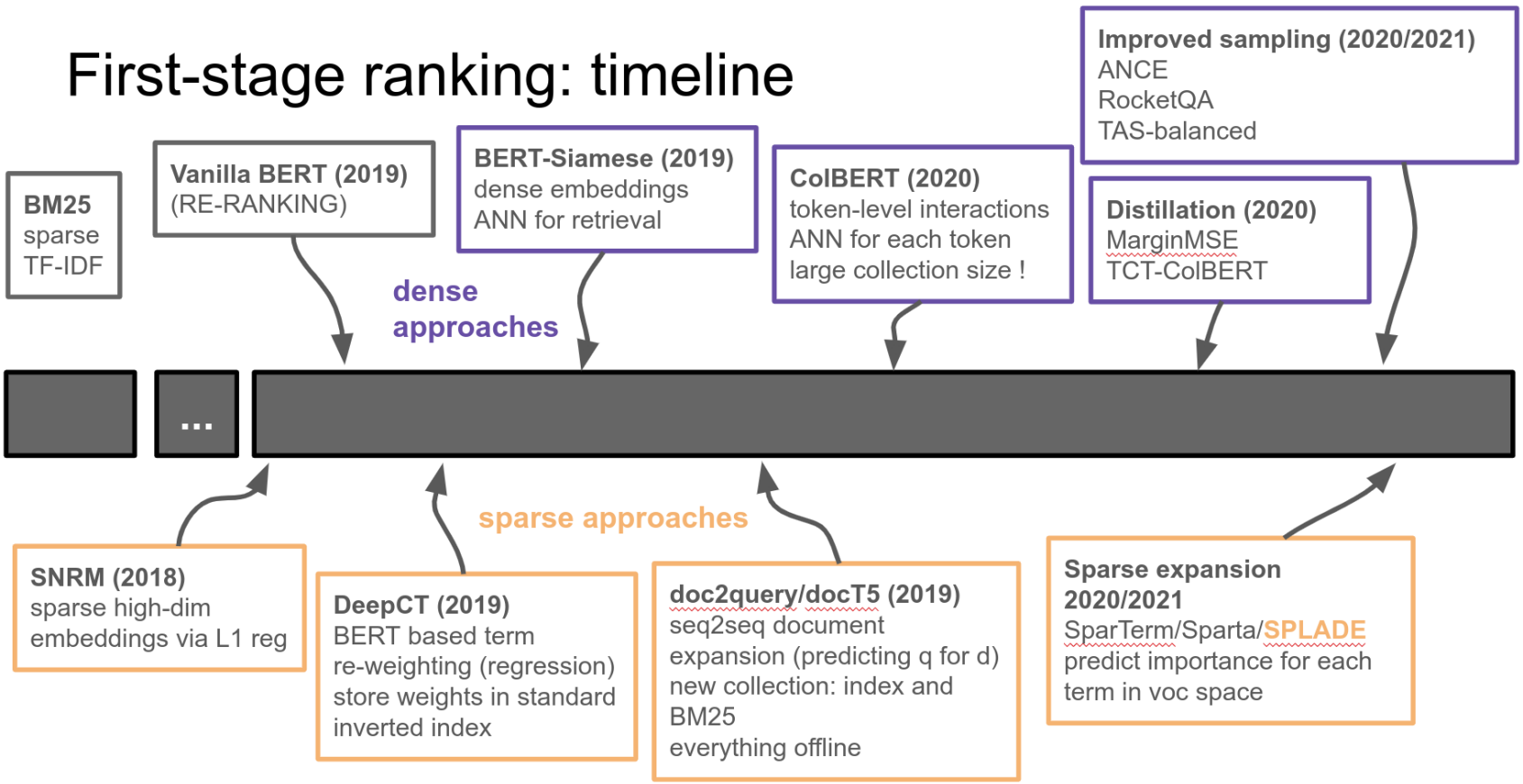
$$\Delta(f_\theta(x), c') = \log p(c = c'|x;\theta) = \sum_c \mathbf{1}[c = c'] \log (f_\theta(x))_{c'}$$

- Ordonnancement
  - Max-margin $\Delta(f_\theta(x_+), f_\theta(x_-)) = \max\{0, 1 - (f_\theta(x_+) - f_\theta(x_-))\}$
  - Orientée rang $\Delta(f, q, d_1, \ldots, d_n) = \sum_j R(q, d_j) \log \frac{\exp(f(q,d_j))}{\sum_k \exp(f(q,d_k))}$

Mais en fait... On fait du ré-ordonnancement



Mais en fait... ou pas...



Modèles parcimonieux pour éviter l'étape de pré-ordonnancement BM25

SCIENCES SORBONNE UNIVERSITÉ

Principe

- Les index sont efficaces (voir BM25)
- Comment concilier index parcimonieux avec représentations denses ?
- Quelles stratégies pour un index plus "dense" et donc un ordonnancement efficace (précision/rappel en une seule étape) ?
- Plus facile pour ordonnancer les documents (pas besoin de passer par une étape de détection de candidats)

Modèles parcimonieux: SNRM (Zamani et al., CIKM 2018)

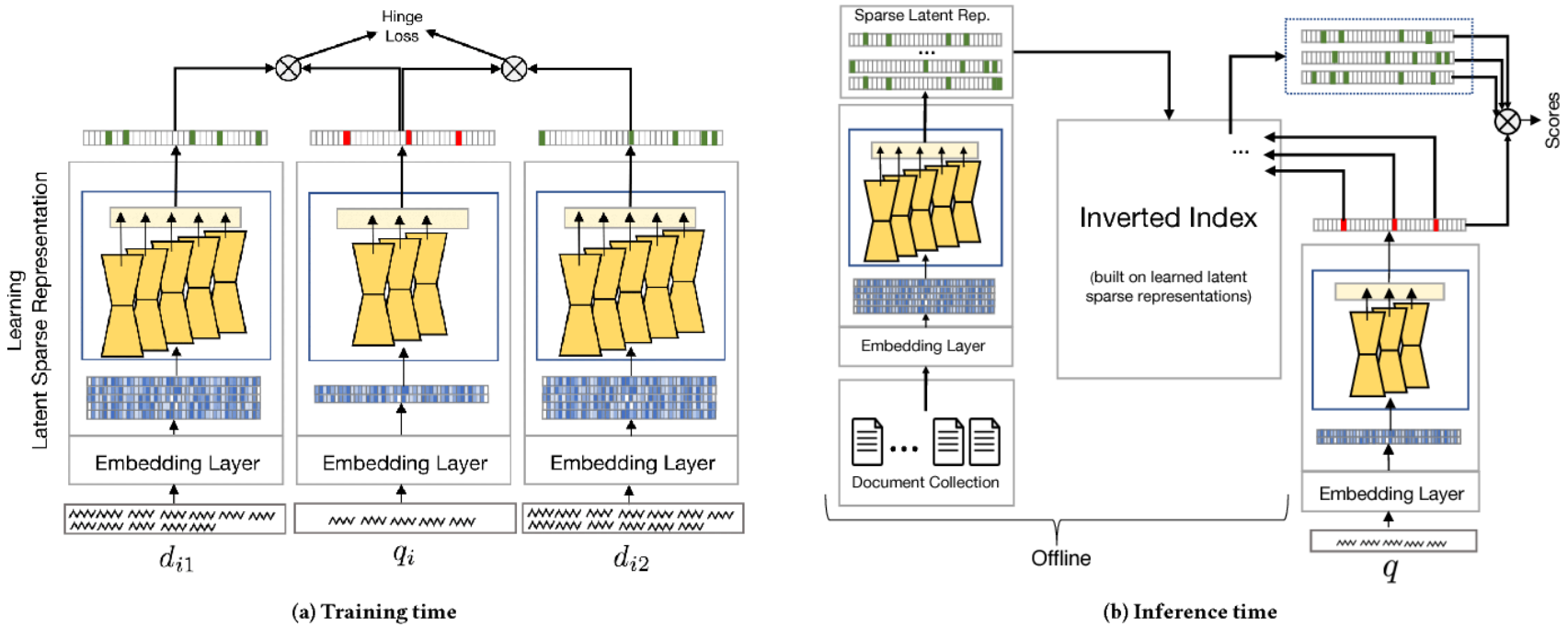


Figure 2: General schema of the proposed SNRM model.

- Offline : Apprentissage de représentation parcimonieuse pour construire l'index inverse (MLP + RELU avec une hinge loss)
- Online : Représentation parcimonieuse de la requête pour sonder l'index inverse et calculer les scores d'appariement

DeepCT

Table 1: Two passages that mention 'stomach' twice. Only the first passage is about the topic 'stomach'. This paper proposes a method to weight terms by their roles in a specific text context, as shown by the heatmap over terms.

In some cases, an upset **stomach** is the result of an allergic reaction to a certain type of food. It also may be caused by an irritation. Sometimes this happens from consuming too much alcohol or caffeine. Eating too many fatty foods or too much food in general may also cause an upset **stomach**.
All parts of the body (muscles , brain, heart, and liver) need energy to work. This energy comes from the food we eat. Our bodies digest the food we eat by mixing it with fluids(acids and enzymes) in the **stomach**. When the **stomach** digests food, the carbohydrate (sugars and starches) in the food breaks down into another type of sugar, called glucose.

Figure 3: Dai and Callan, 2019. Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval

- Estimer la pertinence d'un terme avec la mesure de *query term recall*:

$$QTR(t,d) = \frac{|Q_{d,t}|}{|Q_d|} \tag{5}$$

  avec Q_d : le nombre de requêtes pour lesquelles le passage d est pertinent et $Q_{d,t}$ le sous ensemble de Q_d qui contiennent le terme t.
- Estimer le poids des termes selon une représentation contextualisée (BERT) avec une fonction linéaire
- Indexer seulement les termes les plus importants en fonction du poids estimé.
- Peut aussi être utilisé pour reformuler les requêtes longues

Sparta, SparTerm, Splade

- Prédire également le poids des termes... mais avec une distribution

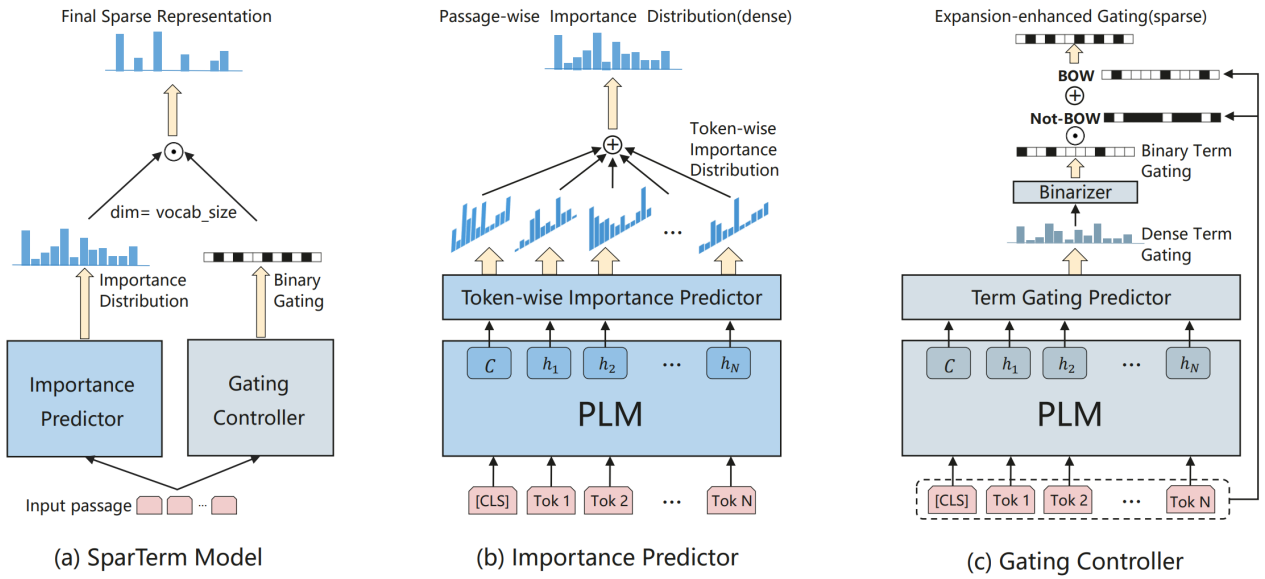


Figure 2: Model Architecture of SparTerm. Our overall architecture contains an importance predictor and a gating controller. The importance predictor generates a dense importance distribution with the dimension of vocabulary size, while the gating controller outputs a sparse and binary gating vector to control term activation for the final representation. These two modules cooperatively ensure the sparsity and flexibility of the final representation.

Figure 4: Bai et al., 2020. SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval

Sparta, SparTerm, Splade

- Importance predictor : distribution en fonction des différentes positions du mot
- Gating controller : Literal-only Gating (basé sur représentation sac de mots - activer seulement les mots dans le passage), Expansion-enhanced Gating (pour réduire le gap sémantique)
- Un pooling est ensuite appliqué pour n'obtenir qu'un poids par terme

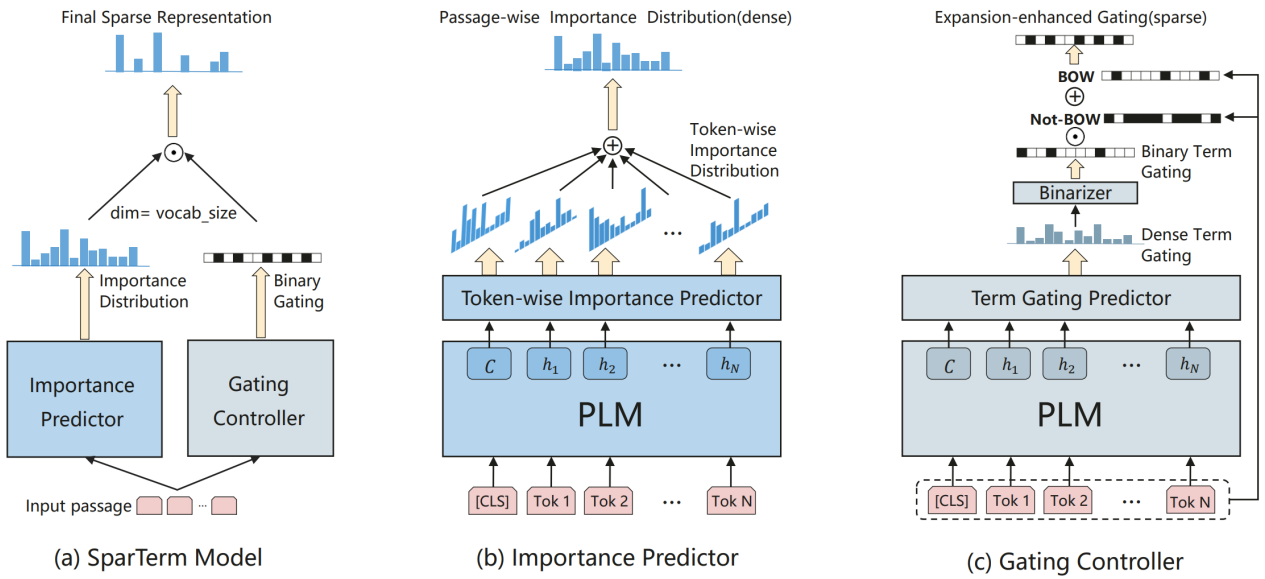


Figure 2: Model Architecture of SparTerm. Our overall architecture contains an importance predictor and a gating controller The importance predictor generates a dense importance distribution with the dimension of vocabulary size, while the gating controller outputs a sparse and binary gating vector to control term activation for the final representation. These two modules cooperatively ensure the sparsity and flexibility of the final representation.

Figure 5: Bai et al., 2020. SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval

La parcimonie détournée : étendre le document avec des modèles de langue

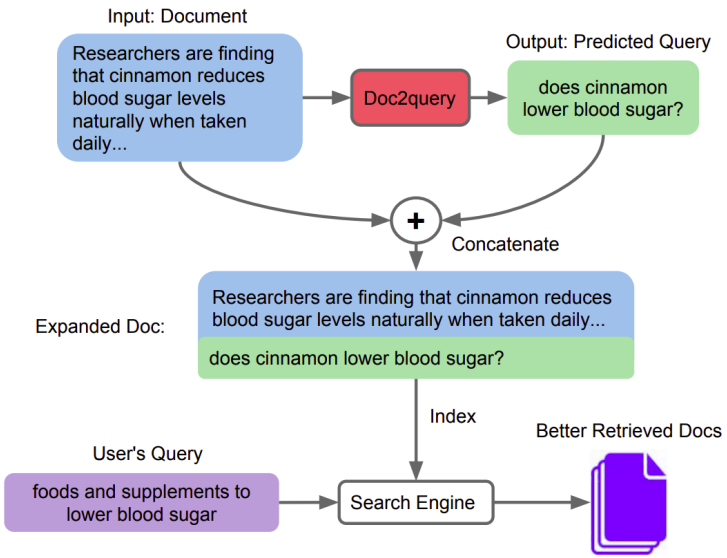


Figure 1: Given a document, our Doc2query model predicts a query, which is appended to the document. Expansion is applied to all documents in the corpus, which are then indexed and searched as before.

Figure 6: Document Expansion by Query Prediction. Nogueira et al., 2019

- Prédire un ensemble de requêtes associées au document.
- Extension du document avec un set de 10 requêtes
- Indexation et ordonnancement avec BM25
- Doc2TTTT, c'est juste avec T5 plutôt que BERT

Des modèles de distillation

- Constat :
  - Certaines données n'ont pas assez de jugement de pertinence (MSMarco)
  - les jugements de pertinence sont souvent binaires (grain pas assez fin)
- Mise en place d'architecture Teacher-Student :
  - Teacher : Modèle appris pour prédire des labels sur des documents (on a donc des scores pour tous les exemples - processus lent)
  - Student : Modèle qui s'appuie sur les labels pour apprendre l'ordonnancement

Exemple du principe Teacher-Student en RI

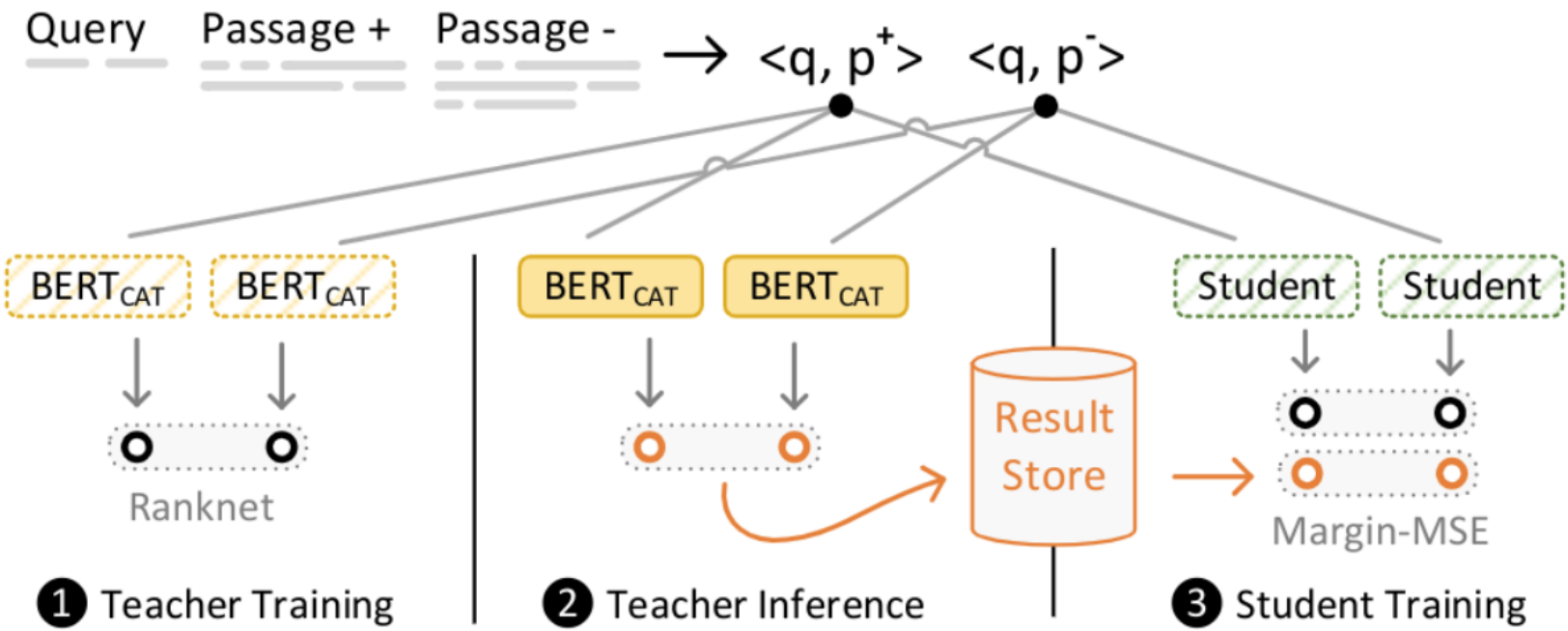


Figure 7: Hofstätter et al., 2020. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation

Quelques pointeurs sur les jeux de données

- MsMarco : le plus utilisé pour Deep learning - DEEP LEARNING TRACK à TREC
  - Articles Wikipédia, question en langage naturel (proche QA)
  - Un seul document pertinent (train)
  - Tâches : ordonnancement de documents, de passages, ...
  - Sert de base à de nombreux jeux de données : TREC CAST, conversational search, ...
- TREC COVID
  - Articles scientifiques : PubMed, OMS, Arxiv, ...
  - Requêtes : extraites de conversations sur le web (twitter, forums, ...), des recherches web sur les plateformes médicales, ...
- et d'autres : TREC CAR, Robust (plus petit), Gov, ...

Conclusion

Il reste encore beaucoup à faire...

- Modèles avec peu de données
  - Supervision
    - Des logs (clics utilisateurs, ...)
    - Des modèles de RI qui sont efficaces
  - Données non alignées
    - Apprendre le modèle de langue des requêtes et des documents
- Mieux prendre en compte l'utilisateur
  - Modèles d'interaction
  - Apprentissage par renforcement
  - Contexte et conversation (chatbot)
- Explicabilité : expliquer la prédiction

Quelques pointeurs pour vos lectures

- Conférences internationales RI : SIGIR, ECIR, ICTIR, CIKM, WSDM, WWW / nationale : CORIA
- Mais aussi les conférences NLP : EMNLP, ACL, COLING, ...
- Journaux internationaux : ACM TOIS, ACM Computing Surveys, IP&M, JIR, JASIST
- Campagnes d'évaluation : TREC, NIST, CLEF, FIRE
- Livres :
  – Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze, Introduction to Information Retrieval, Cambridge University Press. 2008.
  – Massih-Reza Amini, Eric Gaussier, Recherche d'information, Applications, modeles et algorithmes, Eyrolles 2013-2018
  – W. Bruce Croft, Donald Metzler, Trevor Strohman, Search Engines Information Retrieval in Practice, Addison Wesley, 2009
- Synthèses:
  - Sur la Recherche d'Information Neuronale (en particulier les loss) : Mitra, B. and Craswell, N. 2017. An Introduction to Neural Information Retrieval.
  - Sur la RI et les transformers (très complet) : Lin, J. et al. 2020. Pretrained Transformers for Text Ranking: BERT and Beyond.