Génération de textes artificiels pour l'expansion de requêtes

Vincent Claveau

Génération de textes artificiels pour l'expansion de requêtes

Vincent Claveau

IRISA, CNRS, Campus de Beaulieu 35042 Rennes

RÉSUMÉ. *Un moyen d'améliorer les performances de la recherche de documents consiste à étendre la requête de l'utilisateur. Plusieurs approches ont été proposées dans la littérature, et certaines d'entre elles obtiennent des résultats très compétitifs. Dans cet article, nous explorons l'utilisation de la génération de texte pour étendre automatiquement les requêtes. Nous nous appuyons sur un modèle génératif neuronal bien connu, GPT-2, qui est fourni avec des modèles pré-entraînés pour l'anglais mais qui peut également être affiné sur des corpus spécifiques. À travers différentes expériences, nous montrons que la génération de texte est un moyen très efficace d'améliorer les performances d'un système de RI, avec une marge importante (+10% de gains MAP), et qu'il surpasse des approches état-de-l'art reposant également sur l'expansion des requêtes (LM+RM3). Cette approche conceptuellement simple peut être facilement mise en œuvre sur n'importe quel système de RI grâce à la disponibilité du code et des modèles GPT.*

ABSTRACT. *A well-known way to improve the performance of document retrieval is to expand the user's query. Several approaches have been proposed in the literature, and some of them are considered as yielding state-of-the-art results. In this paper, we explore the use of text generation to automatically expand the queries. We rely on a well-known neural generative model, GPT-2, that comes with pre-trained models for English but can also be fine-tuned on specific corpora. Through different experiments, we show that text generation is a very effective way to improve the performance of an IR system, with a large margin (+10% MAP gains), and that it outperforms strong baselines also relying on query expansion (LM+RM3). This conceptually simple approach can easily be implemented on any IR system thanks to the availability of GPT code and models.*

MOTS-CLÉS : *Génération de textes, modèle de langue génératif, expansion de requête, GPT2, augmentation de données, recherche de documents.*

KEYWORDS: *Text generation, generative language model, query expansion, GPT2, data-augmentation, document retrieval.*

1. Introduction

Dans le cadre traditionnel de la Recherche d'Information (RI), un utilisateur exprime son besoin d'information à l'aide d'une requête. Cependant, il est parfois difficile de faire correspondre la requête avec les documents, par exemple parce que le vocabulaire de la requête peut différer de celui des documents. C'est notamment le cas lorsque la requête est courte : les performances du système sont généralement médiocres, car il est difficile de détecter l'objet précis du besoin d'information et l'importance relative des termes de la requête.

L'expansion de requête vise à résoudre ces problèmes en transformant la requête courte en un texte (ou un ensemble de mots) plus grand qui permet de faire correspondre plus facilement les documents de la collection. La principale difficulté de l'expansion de requête est évidemment d'ajouter uniquement les termes pertinents à la requête initiale. Plusieurs techniques ont été proposées dans la littérature, basées sur les ressources linguistiques (par exemple, des listes de synonymes) ou sur les documents eux-mêmes (par exemple, *pseudo-relevance feedback*).

Dans cet article, nous explorons l'utilisation de modèles de génération de texte récents pour étendre les requêtes. Nous démontrons expérimentalement que les récentes avancées en matière de génération de neurones peuvent améliorer considérablement la recherche ad hoc, même lorsqu'il s'agit de RI en domaines spécialisés. Plus précisément, nous démontrons par différentes expériences que :

1) des textes générés artificiellement à partir de la requête peuvent être utilisés pour de l'expansion de requête ;

2) cette approche ne permet pas seulement d'ajouter de nouveaux termes à la requête, mais permet aussi de mieux estimer leur importance (poids) ;

3) cette approche permet également de mieux estimer les poids des mots de la requête originale ;

4) l'approche fonctionne aussi sur des domaines spécialisées.

L'article est structuré de la façon suivante : après une présentation des travaux proches, la section 3 détaille les différents composants de notre approche. Plusieurs expériences visant à valider les affirmations exposées ci-dessus sont présentées en section 4. Quelques conclusions et pistes pour des travaux futurs sont donnés finalement en section 5.

2. Travaux connexes

L'expansion des requêtes est une technique bien établie pour essayer d'améliorer les performances d'un système IR. On attend généralement de l'ajout de nouveaux termes à la requête une amélioration du rappel, mais comme la requête résultante est, dans le meilleur des cas, mieux formulée et plus détaillée, elle peut également améliorer les résultats en haut de liste et être bénéfique à la précision. On peut classer

les approches automatiques existantes en fonction de l'origine de la ressource utilisée pour étendre la requête.

2.1. *Ressources externes*

Une façon évidente d'étendre une requête est d'y ajouter des termes sémantiquement liés (synonymes ou partageant d'autres relations sémantiques comme les hyponymes, quasi-synonymes, méronymes...). Les ressources lexicales existantes peuvent être utilisées pour ajouter, pour chaque terme de la requête, une liste de termes sémantiquement liés ; cependant, il faut faire face à différents problèmes : existence de ressources lexicales pour la langue de la collection, ou pour le domaine spécifique de la collection, choix de la pertinence de certaines relations, besoin de désambiguïsation du sens pour les mots polysémiques... WordNet (Miller, 1990) est une des ressources les plus connues pour l'anglais (pour la langue générale) et a été utilisé avec des résultats décevants lors des premières expériences (Voorhees, 1994), mais qui s'est finalement révélé efficace (Claveau et Kijak, 2016, inter alia).

2.2. *Ressources fondées sur la collection*

Des thésaurus distributionnels ont également été exploité pour enrichir les requêtes. Comme ils peuvent être construits à partir de la collection de documents (ou d'un grand corpus aux caractéristiques similaires), ils sont adaptés à la langue, au domaine, au vocabulaire... Les techniques traditionnelles de construction de ces thésaurus ont obtenu de bons résultats pour l'expansion des requêtes (Claveau et Kijak, 2016). Les approches neuronales, c'est-à-dire les approches de plongements de mots (*word embedding*), sont maintenant largement utilisées pour construire de telles ressources sémantiques. Ces dernières années, des plongements statiques (word2vec (Mikolov *et al.*, 2013), Glove (Pennington *et al.*, 2014) ou FastText (Bojanowski *et al.*, 2016) pour n'en citer que quelques-uns) ont ainsi été utilisés en RI, notamment pour enrichir la requête. En effet, ces représentations denses et entraînables permettent de trouver facilement des mots sémantiquement proches des mots de la requête.

Plus récemment encore, des représentations dynamiques de mots obtenues avec des architectures basées sur des *transformers*, telles que BERT (Devlin *et al.*, 2019) ou GPT (Radford *et al.*, 2019), ont été proposées. Elles construisent une représentation pour chaque mot en fonction de son contexte, et cette capacité a été exploitée pour obtenir des résultats compétitifs dans les tâches RI (Dai et Callan, 2019 ; Khattab et Zaharia, 2020a, inter alia). BERT a également été utilisé pour l'expansion des requêtes dans le cadre d'un système IR neuronal (Khattab et Zaharia, 2020b ; Zheng *et al.*, 2020 ; Naseri *et al.*, 2021), permettant par exemple le reclassement (Padaki *et al.*, 2020)

2.3. *Pseudo-relevance feedback*

Une dernière catégorie d'études ne prend en compte qu'un petit ensemble de documents pour aider à étendre, reformuler ou re-pondérer la requête. Pour être automatiques, elles remplacent la rétro-action de pertinence de l'utilisateur (*relevance feedback*) par l'hypothèse que les documents les mieux classés, récupérés avec la requête originale, sont pertinents et peuvent contenir des informations sémantiques utiles (Ruthven et Lalmas, 2003). Il est intéressant de noter que dans ce cas, non seulement des termes sémantiquement pertinents sont extraits, mais aussi des informations statistiques sur la distribution de ces termes et de ceux de la requête originale. Dans cette catégorie, l'approche Rocchio, développé dans les années 60 pour le modèle vectoriel, a été parmi les premiers à être popularisé (Manning *et al.*, 2008). L'une des approches les plus connues actuellement est RM3, qui a été développée dans le cadre des systèmes de RI fondés sur des modèles de langage (Abdul-jaleel *et al.*, 2004). On rapporte souvent qu'elle donne les meilleurs résultats dans les tâches d'extraction ad hoc, même par rapport aux modèles neuronaux récents (Lin, 2018). Notons enfin que des approches neuronales ont également été proposées pour intégrer des informations de pseudo-relevance feedback (Li *et al.*, 2018), mais, comme le rapportent les auteurs, les résultats sont encore inférieurs à ceux des modèles traditionnels avec expansion de la requête.

2.4. *Positionnement par rapport à l'état de l'art*

Dans cet article, nous proposons d'utiliser la génération de texte (contrainte) pour étendre les requêtes. Dans cette approche, la requête originale est utilisée comme une amorce donnée à un modèle de génération qui produira des textes qui sont, nous l'espérons, liés à la requête. Si la génération de texte et les modèles de langue génératifs ne sont pas nouveaux, les performances des modèles neuronaux récents basés sur les transformers (Vaswani *et al.*, 2017) rendent cette tâche réaliste. Dans cet article, nous utilisons les modèles de type *Generative Pre-trained Transformers* (GPT). Ils sont construits à partir de transformers empilés (précisément, des décodeurs) entraînés sur un grand corpus par auto-régression, c'est-à-dire entraînés sans supervision à prédire le prochain mot (ou plus précisément *token*) sachant les mots précédents. La deuxième version, GPT-2 (Radford *et al.*, 2019), contient 1,5 milliard de paramètres pour son plus grand modèle pré-entraîné sur plus de 8M de documents issus de Reddit (c'est-à-dire des documents principalement en anglais et de langue générale, comme des discussions sur les articles de presse). Une version plus récente, GPT-3, a été annoncée en juillet 2020 ; elle est beaucoup plus grande (175 milliards de paramètres) et surpasse GPT-2 dans toutes les tâches testées. Pour autant, GPT-3 n'est pas librement accessible et il n'est pas possible de ré-entraîner ces très gros modèles. Les expériences rapportées ci-dessous ont donc été réalisées avec GPT-2.

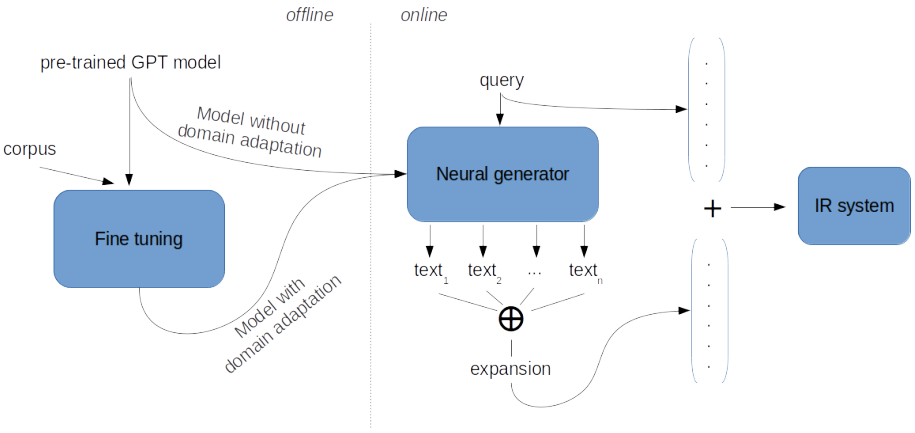

Figure 1 : Schéma global de notre approche d'expansion de requête par génération de textes artificiels.

3. Expansion artificielle de requêtes

3.1. *Aperçu de l'approche*

Comme cela a été expliqué précédemment, notre approche est simple de conception et de mise en œuvre car elle s'appuie sur des outils et des techniques existants. À partir d'une requête, plusieurs textes sont générés par un modèle GPT-2 en utilisant la requête comme armorce. Notez que le processus de génération n'est pas déterministe, et donc, même avec la même amorce, les textes générés peuvent être différents. Dans les expériences rapportées ci-dessous, plusieurs textes par requête sont générés. La génération d'un grand nombre de textes permet ainsi d'avoir un vocabulaire large et une bonne estimation de l'importance relative des mots par leur fréquence dans les textes générés. Ces textes sont concaténés et serviront, comme pour le *relevance feedback*, à étendre la requête. La figure 1 détaille l'ensemble du processus, dont les détails sont donnés dans les sous-sections suivantes.

Un exemple de texte généré à partir d'une requête est présenté dans la figure 2. Comme on peut le voir, le texte généré, bien que complètement fictif (notez les prix du baril), est pertinent pour la requête. Il contient de nombreux termes, absents de la requête originale, qui sont plus ou moins étroitement liés au besoin d'information :

– des variantes orthographiques (`United States` vs. `U.S.`);

– des méronymes-métonymes (`barrel` vs. `oil`);

– des hyperonymes (`energy` vs. `oil`);

– plus généralement toute relation paradigmatique (`consumer`, `producer` vs. `industry`);

– et des relations syntagmatiques (`production` pour `oil`).

U.S. oil production has been declining steadily for decades and it is not expected to reverse. In fact, some argue that it may even get worse. The long-term trend is for oil production to decline at a rate of about 1 percent per year. With production of about 8 million barrels per day now, there is no way the United States can replace its current output.
The U.S. oil boom was a result of an energy revolution in the 1970s that led to increased oil production, and a significant change in the global oil market. The U.S. now produces about 2.3 million barrels of oil per day, the highest it has been in over 30 years.
The United States is now the world's largest oil producer and the fourth largest oil exporter.
What happened ?
When oil prices peaked in the 1970s, the United States was the world's largest oil producer. But over the next several decades, the United States' oil production began to decline. The decline was most pronounced in the 1980s, when the United States began to fall behind other oil producing countries.
The oil price decline in the 1970s was not entirely voluntary. The United States was producing less oil and exporting more oil than it was consuming. The Federal Reserve controlled the amount of dollars in the Federal Reserve's reserves, so the United States was not exporting as much oil as it was producing. The decline in U.S. oil production was a result of the declining price of oil.
The price of oil had declined from $8 per barrel in 1973 to $2.50 per barrel in 1977. In 1979, the price of oil reached a high of $15.75 per barrel. By 1983, the price of oil had fallen to $4.65 per barrel. By 1986, the price of oil had fallen to $1.86 per barrel. By the end of the 1980s, the price of oil had fallen to $1.24 per barrel.
The decline in oil prices was a direct result of the energy revolution in the 1970s. The United States was the world's largest oil producer, but the United States was also the world's largest consumer of oil. When oil prices fell, so did the cost of producing oil.

Figure 2 : Exemple de document généré avec GPT-2 et son plus grand modèle pré-entraîné 1.5 milliards de paramètres) à partir du texte amorce "U.S. oil industry history" (requête 701 de la collection GOV2).

Il est à noter que ces textes donnent également une information précieuse sur la fréquence relative de chaque terme (contrairement aux thésaurus ou aux plongements).

3.2. *Modèles pré-entraînés, ré-entraînement et paramètres*

GPT-2 est disponible avec plusieurs modèles pré-entraînés, ayant des tailles différentes en termes de paramètres (de 124 millions à 1,5 milliards). Comme il a été dit précédemment, leurs données d'entraînement étaient des textes de la langue générale axés sur les actualités. Le plus grand modèle (1,5 milliards de paramètres) a été utilisé pour deux des collections testées (voir ci-dessous). Ces modèles polyvalents

conviennent aux collections de RI dont les documents sont en anglais non spécialisé, ils peuvent ne pas convenir aux collections de RI spécifiques à un domaine. Dans les expériences rapportées dans la section suivante, nous utilisons entre autre la collection OHSUMED, constituée de documents médicaux en anglais. Pour cette collection, nous avons affiné (*fine-tuning*) le modèle GPT-2 de 355M paramètres sur les documents de la collection afin d'adapter le modèle de langue à la syntaxe et au vocabulaire spécifique du domaine médical. Le fine-tuning a été arrêté après le traitement de 250 000 échantillons du corpus OHSUMED (ce nombre d'échantillons contrôle indirectement le sous/sur-apprentissage au corpus spécialisé) et les autres paramètres (taille du batch, optimiseur, taux d'apprentissage...) ont été laissés à leurs valeurs par défaut. Bien qu'un ensemble plus important de documents médicaux puisse être utilisé (de Pubmed par exemple), ce petit modèle finement ajusté devrait être plus adapté (qu'un modèle généraliste même plus large) pour générer des documents utiles pour enrichir la requête.

Concernant la génération de documents, à des fins de reproductibilité, voici les principaux paramètres GPT-2 utilisés (cf. documentation GPT-2[1]) : longueur=512, température=0,5, top_p=0,95, top_k = 40. La génération pouvant se faire en parallèle, on produit, pour une même amorce, 40 textes en même temps (c'est le maximum possible que nous avons constaté sur une carte GPU NVidia V100 avec 32GB VRAM pour le modèle de 355 paramètres), ce qui est fait en environ 10 secondes avec nos paramètres.

3.3. *Systèmes de RI*

Dans les expériences rapportées dans la section suivante, nous utilisons deux modèles de RI. Le premier est BM25+ (Lv et Zhai, 2011), une variante du BM25 (Robertson *et al.*, 1998). Les paramètres k_1, k_3, b et δ ont été conservés à leurs valeurs usuelles (resp. 1.2, 1000, 0.75, 1). Il est implémenté sous la forme d'une modification personnalisée de la bilbiothèque GENSIM (Řehůřek et Sojka, 2010). Le second modèle de RI est un modèle de langue avec lissage Dirichlet (Zhai et Lafferty, 2001) tel qu'implémenté dans Indri (Metzler et Croft, 2004 ; Strohman *et al.*, 2005). Le paramètre de lissage μ est fixé à 2 500. Les deux modèles sont considérés comme offrant des performances état-de-l'art pour la représentation en sacs-de-mots (Lin, 2018). Leur fonction RSV peut être écrite :

$$RSV(q,d) = \sum_{t \in q} w_q(t) \cdot w_d(t)$$

avec $w_q(t)$ le poids du terme t dans la requête q et $w_d(t)$ le poids dans le document d, comme décliné dans le tableau 1 (d'après (Lv et Zhai, 2011)). Pour l'expansion avec l'approche RM3, nous nous appuyons également sur l'implémentation d'Indri ; les résultats présentés dans la section suivante correspondent aux paramètres les plus

1. `https://github.com/openai/gpt-2`

modèle de RI	pondérations
BM25+ $w_d(t)$	$\left(\frac{(k1+1)c(t,d)}{k1(1-b+b\cdot dl(d)/avdl)+c(t,d)} + \delta\right) \cdot \log \frac{N+1}{df(t)}$
BM25+ $w_q(t)$	$\frac{(k3+1)c(t,q)}{k3+c(t,q)}$
	avec $k1, k_3, b$ et δ des paramètres fixés
LM $w_d(t)$	$\log\left(\frac{\mu}{dl(d)+\mu} + \frac{c(t,d)}{(dl(d)+\mu)p(t\|C)}\right)$
LM $w_q(t)$	$c(t,q)$
	$\mu > 0$ un paramètre de lissage

Tableau 1 : Modèle de RI (fonction de pondérations des termes dans la requête et le document) pour BM25+ (Robertson *et al.*, 1998 ; Lv et Zhai, 2011) et modèle de langue avec lissage de Dirichlet (Zhai et Lafferty, 2001)

performants testés pour chaque collection (nombre de documents pris en compte pour le retour de pseudo-pertinence et nombre de termes conservés).

3.4. *Construction de la requête étendue*

Dans nos expériences, 100 textes (sauf indication contraire) sont générés pour chaque requête, avec une longueur maximale de 512 tokens (voir section 4.5 pour une étude de l'influence du nombre de textes générés). Ces textes sont concaténés et constitue l'extension de requête. Cette requête étendue, très grande, est ensuite transmise à un simple système de RI BM25+ (Lv et Zhai, 2011) dans nos expériences, mais elle pourrait évidemment être utilisée dans n'importe quel autre système de RI, y compris des systèmes neuronaux. En pratique, pour notre système vectoriel, nous utilisons cette expansion pour construire un vecteur contenant la fréquence des termes dans les textes générés. Pour produire le vecteur requête final, il est additionné au vecteur requête initial puis le tout est pondéré selon le schéma de pondération choisi. Le calcul vectoriel pour trouver les documents les plus proches n'étant pas dépendant de la sparsité des vecteurs, la taille de la requête étendue n'a aucune influence sur le temps de réponse du moteur de recherche.

4. Expériences

4.1. *Cadre expérimental*

Trois collections de RI sont utilisées dans nos expériences : Tipster, GOV2 et OHSUMED. Quelques statistiques de base sur ces collections sont données dans le tableau 2.

	Tipster	GOV2	OHSUMED
nb de documents	170 000	25M	350 000
nb de requêtes	50	150	106
taille moyenne des requêtes	6,74	3.15	7,24
langue	En	En	En
nb moyen de doc pertinent par requête	849	179	21

Tableau 2 : Statistiques sur les collections de RI utilisées

Tipster a été utilisé dans TREC-2. Les documents sont des articles de journaux, de brevets et de la presse spécialisée (informatique) en anglais. Les requêtes sont composées de plusieurs parties, dont la requête elle-même et un texte détaillant les critères de pertinence ; dans les expériences rapportées ci-dessous, seule la partie de la requête proprement dite est utilisée.

GOV2 est une vaste collection de pages Web explorées à partir du domaine .gov et utilisées dans plusieurs tâches de TREC. Dans les expériences rapportées ci-dessous, 150 requêtes provenant des tâches de recherche ad hoc de TREC 2004-2006 sont utilisées ; comme pour Tipster, seule la partie requête à proprement parlé est utilisée (c'est-à-dire que les champs de description et de narration ne sont pas inclus dans la requête).

OHSUMED (Hersh *et al.*, 1994) contient des notices bibliographiques de Medline et des requêtes de la tâche de filtrage TREC-9. Son intérêt pour nos expériences est qu'elle traite d'un domaine spécialisé, elle contient donc un vocabulaire spécifique.

Les performances sont évaluées à l'aide des mesures usuelles : Précision à différents seuils (P@x), R-précision (R-prec), MAP. Si nécessaire, un t-test pairé avec $p = 0,05$ est effectué pour évaluer la significativité statistique de la différence entre les systèmes.

4.2. *Langue générale*

Les tableaux 3 et 4 présentent respectivement les résultats sur les collections en langue générale Tipster et GOV2. A titre de comparaison, nous indiquons les résultats de BM25+ sans expansion, du modèle de langue d'Indri (LM) avec et sans expansion RM3. Le réglage le plus performant pour RM3 sur Tipster est de 100 termes pour les 20 premiers documents, et de 100 termes pour les 10 premiers documents GOV2. La significativité statistique est calculée en comparant avec la baseline LM+RM3.

Sur les deux collections, et sur chaque mesure de performance, l'expansion des requêtes avec les textes générés apporte des gains importants par rapport au système sans expansion. En outre, notre approche surpasse l'expansion de RM3 dans presque

	MAP	R-Prec	P@5	P@10	P@20	P@100
BM25+	25,06	32,16	95,60	92,60	89,70	73,64
LM	24,48	31,48	92,40	89,00	85,40	70,70
LM + RM3	31,01	36,38	94,40	93,20	90,60	81,22
BM25+ et expansion	**35,22***	**39,87***	**99,60***	**98,40***	**98,20***	**87,84***

Tableau 3 : Performances (%) de l'expansion de requête sur la collection Tipster ; les meilleurs résultats sont en gras, la significativité statistique par rapport à LM+RM3 est notée avec $*$

	MAP	R-Prec	P@5	P@10	P@20	P@100
BM25+	25,66	31,25	52,92	49,97	46,52	34,63
LM	27,96	33,01	56,08	55,20	51,59	37,32
LM + RM3	30,22	34,20	55,00	56,08	53,67	**45,86**
BM25+ et expansion	**34,54***	**37,76**	**67,91***	**63,88***	**57,94**	44,30

Tableau 4 : Performances (%) de l'expansion de requête sur la collection GOV2 ; les meilleurs résultats sont en gras, la significativité statistique par rapport à LM+RM3 est notée avec $*$

toutes les situations, et avec une grande marge en terme de MAP, R-prec et précision sur les documents en tête de liste (P@5, P@10).

4.3. *Langue de spécialité*

Le même cadre expérimental que précédemment est utilisé avec la collection OHSUMED. Pour ce jeu de données de RI orienté sur le domaine médical, nous rapportons les résultats de deux versions de notre approche : l'une repose sur les modèles de génération pré-entraînés comme auparavant, l'autre s'appuie sur un modèle adapté (*fine-tuning*) sur les documents de la collection. Le meilleur paramétrage pour RM3 est ici de 80 termes issus des 10 premiers documents. Les résultats sont rapportés dans le tableau 5.

On constate de nouveau que l'expansion effectuée à partir de GPT améliore largement les résultats du système de RI, et dépasse les performances de l'expansion avec RM3. Cependant, les gains constatés sont de moindre ampleur que pour les deux collections précédentes. Cela peut s'expliquer par les facteurs suivants :

1) les requêtes d'OHSUMED sont plus longues, plus complexes et plus spécifiques que celles de GOV2 ou Tipster (voir tableau 2), et peu de documents sont jugés pertinents ;

2) le modèle de génération de textes n'est pas suffisamment adapté aux documents.

	MAP	R-Prec	P@5	P@10	P@20	P@100
BM25+	18,27	19,94	31,88	26,04	20,50	9,48
LM	17,61	20,35	29,31	24,06	19,21	9,28
LM + RM3	20,80	22,54	30,89	26,83	22,18	10,51
BM25+ et expansion (sans fine-tuning)	21,60	23,75	33,47*	27,62	22,92	11,16
BM25+ et expansion (avec fine-tuning)	**23,07***	**24,65***	**34,65***	**29,41***	**24,31**	**11,42**

Tableau 5 : Performances (%) de l'expansion de requête sur la collection OHSUMED ; les meilleurs résultats sont en gras, la significativité statistique par rapport à LM+RM3 est notée avec $*$

Pour ce second facteur, on peut effectivement constater l'intérêt du fine-tuning du modèle génératif, mais de meilleurs résultats pourraient encore être obtenus en utilisant un plus grand jeu de documents du domaine, ou bien en adoptant des paramètres de fine-tuning différents (en particulier le nombre d'epochs ou d'exemples traités ; voir section 3.2). Malheureusement, connaître a priori ces paramètres optimaux n'est pas possible et le coût du fine-tuning rend impossible la recherche exhaustive du meilleur jeu de paramètres.

4.4. *Expansion de requêtes et information fréquentielle*

L'un des intérêts de la génération de textes complets est que nous pouvons recueillir des informations sur l'importance relative des mots, au contraire de l'élargissement des requêtes à l'aide d'un thésaurus. Pour observer l'impact du nombre d'occurrences des mots dans les textes générés, nous évaluons l'effet de conserver les termes les plus fréquents (k) des textes générés et de les pondérer soit par leur fréquence (comme le fait habituellement le modèle BM25), soit en leur donnant un poids fixe (défini à $1/k$). Les résultats pour différentes valeurs de k sont présentés dans la figure 3. On peut observer que l'ajout de termes à la requête avec un poids fixe améliore légèrement la MAP, mais la majeure partie du gain est en effet apportée par une pondération adéquate basée sur la fréquence du terme dans les documents générés.

Dans l'expérience suivante, nous examinons comment les textes générés peuvent aider à re-pondérer les termes de la requête initiale. Il n'y a pas d'expansion de la requête, puisque seuls les termes de la requête initiale sont conservés, mais leur fréquence dans les textes générés est utilisée dans le w_q de BM25+. Les résultats présentés dans le tableau 6 montrent qu'il y a effectivement une légère amélioration de la MAP, qui est plus perceptible à un rang élevé (cf. précision sur les 100 premiers documents). Ces deux expériences démontrent l'intérêt de manipuler des textes complets plutôt que des similarités mot à mot (comme avec les plongements et les thésaurus)

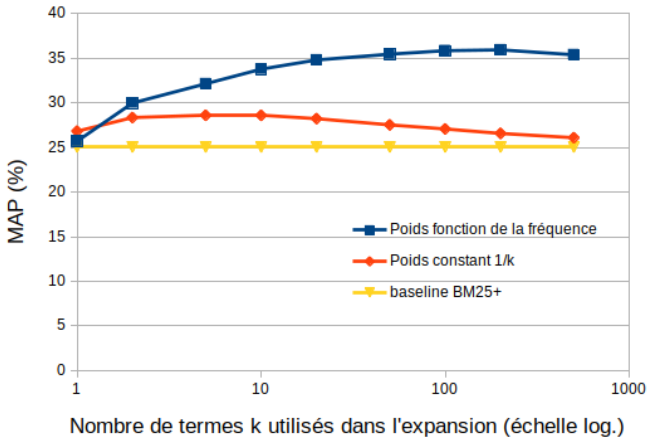

Figure 3 : MAP (%) selon la taille de l'expansion (k), termes avec un poids fixe ou un poids fonction de leur fréquence dans les documents générés ; collection Tipster

	MAP	R-Prec	P@5	P@10	P@20	P@100
BM25+	25,06	32,16	95,60	92,60	89,70	73,64
BM25+ avec re-pondération	27,12	33,22	96,80	94,20	92,70	77,12

Tableau 6 : Performances (%) de la re-pondération des termes de la requête ; collection Tipster

puisque l'on dispose alors d'informations fréquentielles en plus de nouvelles informations sémantiques.

4.5. *Influence du nombre de documents générés*

Comme la génération de textes peut être coûteuse, il est intéressant de voir combien de textes générés sont nécessaires. Dans la Fig. 4, le MAP obtenu en fonction du nombre de documents fictifs générés (jusqu'à un maximum de 100 documents) est présentée. On peut observer qu'un plateau est rapidement atteint (à environ 20 documents par requête). Bien entendu, la taille des documents générés (qui peut être définie comme un paramètre du processus de génération) est également à prendre en compte.

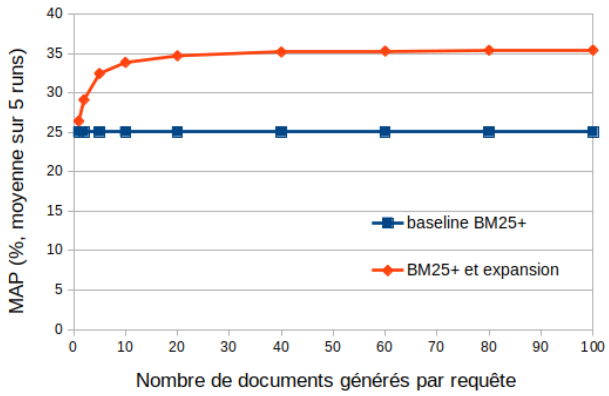

Figure 4 : MAP (%) selon le nombre de documents artificiels générés (moyennée sur 5 runs) ; collection Tipster

5. Conclusion et perspectives

Les approches neuronales sont de plus en plus utilisées en RI, avec des résultats mitigés, surtout si on les compare aux approches "traditionnelles" par sac de mots (Lin, 2018 ; Yang *et al.*, 2019). Ici, la partie neuronale est utilisée avec succès en dehors d'un système RI "traditionnel" (mais notez qu'elle pourrait être utilisée avec n'importe quel système de RI, y compris neuronal, puisqu'elle ne fait qu'enrichir la requête). L'approche d'expansion présentée dans cet article est simple et facile à mettre en œuvre (grâce à la disponibilité des modèles et du code GPT) tout en offrant des gains de performance très importants. Dans le travail rapporté ici, de nombreux paramètres sont encore optimisables, notamment du côté du modèle GPT (notamment les paramètres influençant la "créativité" de la génération de texte), et les possibilités du fine-tuning devraient également être explorées plus en profondeur (influence d'un corpus spécialisé plus important s'il est disponible, mélange précis entre le modèle généraliste et celui spécialisé, etc.). La disponibilité récente de GPT-3[2] ou d'autres modèles génératifs plus récents pourrait permettre d'obtenir des gains encore plus importants grâce à la qualité prétendument élevée des textes générés. Enfin, notons que le temps de génération des textes factices et la puissance nécessaire sont actuellement un frein pour une utilisation massive, mais pas complètement rédhibitoires puisque que comme nous l'avons écrit précédemment, pour une requête, sur une carte GPU, 40 textes sont générés en quelques secondes. Des techniques de réduction de modèle, comme la distillation (Sanh *et al.*, 2020), pourraient permettre de diminuer encore le temps de génération et son coût calculatoire.

2. `https://github.com/openai/gpt-3`

Toute notre approche ouvre également de nombreuses pistes de recherche : dans ce travail, nous avons utilisé la génération de texte comme moyen d'augmenter les données du côté de la requête, mais elle pourrait également être utilisée pour augmenter la représentation des documents (Nogueira *et al.*, 2019) même si en pratique, le coût est encore prohibitif sur les grandes collections. Toutes les approches d'apprentissage artificiel (neuronales ou non) basées sur du *pseudo-relevance feedback* pour entraîner leur modèle pourraient en revanche utiliser une génération de texte similaire, avec l'avantage de ne pas être limitées par le nombre de documents potentiellement pertinents dans la liste restreinte de top-k documents généralement considérés. Et bien sûr, une stratégie d'enrichissement des données similaire pourrait être utilisée pour d'autres tâches que la recherche de documents.

Plus fondamentalement, les récentes améliorations de la génération de texte remettent également en question la pertinence de la tâche de recherche de documents. En effet, il est possible d'envisager des systèmes qui seront capables de générer un document unique répondant au besoin d'information de l'utilisateur, se rapprochant ainsi des systèmes de questions-réponses. Si le modèle de génération est entraîné sur la collection de documents servant d'index, le document généré servira de résumé (qui est l'une des applications populaires des modèles GPT-x) des documents pertinents. Cependant, les limites actuelles des modèles testés dans cet article les rendent loin d'être adaptés à cette tâche ultime : les documents générés traitent effectivement du sujet de la requête, et utilisent donc un vocabulaire pertinent, mais ne fournissent pas d'informations précises et factuelles (comme le montre l'exemple de la figure 2 sur le prix des barils de pétrole).

6. Bibliographie

Abdul-jaleel N., Allan J., Croft W. B., Diaz O., Larkey L., Li X., Smucker M. D., Wade C., « UMass at TREC 2004 : Novelty and HARD », *In Proceedings of TREC-13*, 2004.

Bojanowski P., Grave E., Joulin A., Mikolov T., « Enriching Word Vectors with Subword Information », *arXiv preprint arXiv :1607.04606*, 2016.

Claveau V., Kijak E., « Direct vs. indirect evaluation of distributional thesauri », *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*, The COLING 2016 Organizing Committee, Osaka, Japan, p. 1837-1848, December, 2016.

Dai Z., Callan J., « Deeper Text Understanding for IR with Contextual Neural Language Modeling », *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul, 2019.

Devlin J., Chang M.-W., Lee K., Toutanova K., « BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding », 2019.

Hersh W., Buckley C., Leone T. J., Hickam D., « OHSUMED : An Interactive Retrieval Evaluation and New Large Test Collection for Research », *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, Springer-Verlag New York, Inc., New York, NY, USA, p. 192-201, 1994.

Khattab O., Zaharia M., « ColBERT : Efficient and Effective Passage Search via Contextualized Late Interaction over BERT », *in* J. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen, Y. Liu (eds), *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, ACM, p. 39-48, 2020a.

Khattab O., Zaharia M., « ColBERT : Efficient and Effective Passage Search via Contextualized Late Interaction over BERT », *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, Association for Computing Machinery, New York, NY, USA, p. 39–48, 2020b.

Li C., Sun Y., He B., Wang L., Hui K., Yates A., Sun L., Xu J., « NPRF : A Neural Pseudo Relevance Feedback Framework for Ad-hoc Information Retrieval », *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Brussels, Belgium, p. 4482-4491, October-November, 2018.

Lin J., « The Neural Hype and Comparisons Against Weak Baselines », *SIGIR Forum*, vol. 52, n° 2, p. 40-51, 2018.

Lv Y., Zhai C., « Lower-bounding Term Frequency Normalization », *Proc. of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, ACM, New York, NY, USA, p. 7-16, 2011.

Manning C. D., Raghavan P., Schütze H., *Introduction to Information Retrieval*, Cambridge University Press, USA, 2008.

Metzler D., Croft W., « Combining the Language Model and Inference Network Approaches to Retrieval », *Information Processing and Management Special Issue on Bayesian Networks and Information Retrieval*, vol. 40, n° 5, p. 735-750, 2004.

Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., « Distributed Representations of Words and Phrases and their Compositionality », *in* C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger (eds), *Advances in Neural Information Processing Systems 26 : 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, p. 3111-3119, 2013.

Miller G. A., « WordNet : An On-Line Lexical Database », *International Journal of Lexicography*, 1990.

Naseri S., Dalton J., Yates A., Allan J., « CEQE : Contextualized Embeddings for Query Expansion », *Proceedings of European Conference in Information Retrieval ECIR*, Lucca, IT (virtual event), March, 2021.

Nogueira R., Yang W., Lin J., Cho K., « Document Expansion by Query Prediction », 2019.

Padaki R., Dai Z., Callan J., « Rethinking Query Expansion for BERT Reranking », *in* J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, F. Martins (eds), *Advances in Information Retrieval*, Springer International Publishing, Cham, p. 297-304, 2020.

Pennington J., Socher R., Manning C. D., « GloVe : Global Vectors for Word Representation », *Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532-1543, 2014.

Radford A., Wu J., Child R., Luan D., Amodei D., Sutskever I., « Language Models are Unsupervised Multitask Learners », *OpenAI Blog*, 2019.

Řehůřek R., Sojka P., « Software Framework for Topic Modelling with Large Corpora », *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, ELRA, Valletta, Malta, p. 45-50, May, 2010. `http://is.muni.cz/publication/884893/en`.

Robertson S. E., Walker S., Hancock-Beaulieu M., « Okapi at TREC-7 : Automatic Ad Hoc, Filtering, VLC and Interactive », *Proc. of the 7th Text Retrieval Conference, TREC-7*, p. 199-210, 1998.

Ruthven I., Lalmas M., « A survey on the use of relevance feedback for information access systems. », *Knowledge Eng. Review*, vol. 18, nº 2, p. 95-145, 2003.

Sanh V., Debut L., Chaumond J., Wolf T., « DistilBERT, a distilled version of BERT : smaller, faster, cheaper and lighter », 2020.

Strohman T., Metzler D., Turtle H., Croft W., Indri : A language-model based search engine for complex queries (extended version), Technical report, CIIR, 2005.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L. u., Polosukhin I., « Attention is All you Need », *in* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds), *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., p. 5998-6008, 2017.

Voorhees E. M., « Query Expansion Using Lexical-semantic Relations », *Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, Springer-Verlag New York, Inc., New York, NY, USA, p. 61-69, 1994.

Yang W., Lu K., Yang P., Lin J., « Critically Examining the "Neural Hype" : Weak Baselines and the Additivity of Effectiveness Gains from Neural Ranking Models », *in* B. Piwowarski, M. Chevalier, É. Gaussier, Y. Maarek, J. Nie, F. Scholer (eds), *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, ACM, p. 1129-1132, 2019.

Zhai C., Lafferty J. D., « A study of smoothing methods for language models applied to ad hoc information retrieval », *Proc. of the SIGIR conference*, p. 334-342, 2001.

Zheng Z., Hui K., He B., Han X., Sun L., Yates A., « BERT-QE : Contextualized Query Expansion for Document Re-ranking », *Findings of the Association for Computational Linguistics : EMNLP 2020*, Association for Computational Linguistics, Online, p. 4718-4728, November, 2020.