TECHNIQUES D'IDENTIFICATION D'ENTITÉS NOMMÉES ET DE CLASSIFICATION NON-SUPERVISÉE POUR DES REQUÊTES DE RECHERCHE WEB À L'AIDE D'INFORMATIONS CONTENUES DANS LES PAGES WEB VISITÉES

par

Sylvain Goulet

Mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES

UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, 19 juillet 2014

Sommaire

Le web est maintenant devenu une importante source d'information et de divertissement pour un grand nombre de personnes et les techniques pour accéder au contenu désiré ne cessent d'évoluer. Par exemple, en plus de la liste de pages web habituelle, certains moteurs de recherche présentent maintenant directement, lorsque possible, l'information recherchée par l'usager. Dans ce contexte, l'étude des requêtes soumises à ce type de moteur de recherche devient un outil pouvant aider à perfectionner ce genre de système et ainsi améliorer l'expérience d'utilisation de ses usagers.

Dans cette optique, le présent document présentera certaines techniques qui ont été développées pour faire l'étude des requêtes de recherche web soumises à un moteur de recherche. En particulier, le travail présenté ici s'intéresse à deux problèmes distincts. Le premier porte sur la classification non-supervisée d'un ensemble de requêtes de recherche web dans le but de parvenir à regrouper ensemble les requêtes traitant d'un même sujet. Le deuxième problème porte quant à lui sur la détection non-supervisée des entités nommées contenues dans un ensemble de requêtes qui ont été soumises à un moteur de recherche. Les deux techniques proposées utilisent l'information supplémentaire apportée par la connaissance des pages web qui ont été visitées par les utilisateurs ayant émis les requêtes étudiées.

Mots-clés: classification non-supervisée ; requête de recherche web ; détection d'entités nommées ; *topic modeling* ; fouille du web (*web mining*)

Remerciements

Je tiens ici à adresser mes remerciements aux personnes m'ayant apporté leur aide tout au long de la maîtrise et lors de l'élaboration de ce mémoire.

Tout d'abord, je voudrais remercier mon directeur de recherche, Monsieur Shengrui Wang, professeur titulaire à l'Université de Sherbrooke, qui s'est toujours montré disponible et intéressé pour me conseiller et discuter au sujet du travail que j'ai effectué au courant de cette maîtrise.

Je voudrais également remercier Monsieur Matthieu Hébert pour l'intérêt qu'il a porté à mon travail, ainsi que pour sa disponibilité, son esprit critique et ses conseils éclairés.

Finalement, je voudrais remercier Madame Joumana Ghosn, Monsieur Francis Pieraut et Monsieur Frederic Ratle pour avoir partagé leurs connaissances avec moi, m'aidant ainsi à élargir mes horizons.

Abréviations

LDA *Latent Dirichlet Allocation*

TISK-LDA *Topic-In-Set Knowledge Latent Dirichlet Allocation*

VSM *Vector Space Model*

TF-IDF *Term Frequency-Inverse Document Frequency*

RI Recherche d'Information

SCE Somme du Carré des Erreurs

CGS *Collapsed Gibbs Sampling*

GS *Gibbs Sampling*

MCCM Monte-Carlo par Chaînes de Markov

SSGS *Systematic Scan Gibbs Sampler*

RSGS *Random Scan Gibbs Sampler*

URL *Uniform Resource Locator*

ENS Entité Nommée Solitaire

NIM Nom d'Item MicroData

HTML *HyperText Markup Language*

MRI Moteur de Recherche Intelligent

NERC *Named Entity Recognition and Classification*

ENP Entité Nommée Potentielle

FTI Fragment de Texte Important

TALN Traitement Automatique du Langage Naturel

Abréviations

MMC Modèle de Markov Caché

MEM *Maximum Entropy Model*

CRF *Conditional Random Field*

Table des matières

Liste des figures

Liste des tableaux

Introduction

De nos jours, le web est devenu une importante source d'information et de divertissement pour un grand nombre de personnes. Étant donnée l'énorme quantité de pages web existantes, un grand nombre de personnes utilisent les services fournis par les nombreux moteurs de recherche web disponibles afin d'accéder à l'information ou au contenu qu'elles désirent consulter. En effet, les plus populaires de ces moteurs de recherche peuvent recevoir quelques milliards de requêtes par mois [1].

Vu cette grande popularité, les compagnies qui développent ces moteurs de recherche web cherchent constamment de nouvelles façons d'améliorer l'expérience vécue par leurs utilisateurs. Dans cette optique, certains d'entre eux présentent maintenant directement, lorsque possible, l'information recherchée par ses utilisateurs en plus de la liste de pages web habituellement présentée. Des exemples de ce type d'amélioration peuvent être donnés par les systèmes Google Knowledge Graph [2] et Bing Snapshot [3].

Note Dans le restant du présent document, le terme Moteur de Recherche Intelligent (MRI) sera utilisé pour faire référence à ce type de moteur de recherche amélioré.

On suppose que pour pouvoir identifier correctement l'information qu'il doit présenter à l'utilisateur, ce type de système nécessite généralement l'intervention d'un

1. http://www.comscore.com/Insights/Press_Releases/2013/11/comScore_Releases_October_2013_US_Search_Engine_Rankings
2. http://www.google.ca/insidesearch/features/search/knowledge.html
3. http://www.bing.com/blogs/site_blogs/b/search/archive/2012/06/01/do-more-with-snapshot.aspx

processus de Traitement Automatique du Langage Naturel (TALN) afin d'identifier l'intention derrière la requête de l'usager et de détecter les entités nommées présentes dans cette dernière. Comme ce genre de système a pour but de bien répondre à des besoins d'information spécifiques, il peut être souhaitable de faire l'analyse des requêtes qui lui ont été soumises afin de déceler certains problèmes potentiels ou encore de découvrir de nouveaux besoins qui pourraient être pris en charge par le système. Dans cette optique, le travail s'attardera à la résolution de deux problèmes distincts qui sont la classification non-supervisée de requêtes de recherche web selon leurs domaines et la détection non-supervisée d'entités nommées à l'intérieur de requêtes de recherche web. Une description plus détaillée de ces problèmes sera présentée dans le chapitre 1.

Chapitre 1

Comment tirer profit de l'information contenue dans les *logs* d'un moteur de recherche

Le travail qui sera ici présenté a été inspiré par la question suivante :

«Comment tirer profit de l'information contenue dans les *logs* d'un MRI pour générer des informations qui seront utiles pour améliorer le système en place, et ce de manière non-supervisée ?»

Ce premier chapitre introduira de quelle manière le travail présenté propose de répondre à cette question. La section 1.1 présentera plus précisément la problématique étudiée et les sous-problèmes qui s'y rattachent. La section 1.2 présentera quant à elle les données qui seront analysées. La section 1.3 présentera certaines techniques existantes possédant des objectifs similaires à ceux qui seront fixés dans la section 1.1. Finalement, la section 1.4 décrira les techniques proposées pour répondre aux problèmes soulevés et présentera les avantages et désavantages de ces dernières.

1.1 Définition de la problématique

Le travail présenté dans ce document poursuit deux objectifs **distincts et indépendants** qui se rattachent tous deux au même but général de fournir des informations utiles pour l'amélioration des services offerts par un MRI. Ces deux objectifs seront présentés dans cette section. Mais tout d'abord, il est important de noter que le travail dont il est ici question s'attardera à l'étude d'un certain type de requête en particulier. En effet, comme il est ici question de requêtes provenant des *logs* d'un MRI, il est possible de savoir, pour chaque requête, si le MRI a présenté de l'information supplémentaire visant à répondre directement au besoin d'information spécifié par la requête ou non. Pour le présent travail, il a été décidé d'étudier les requêtes pour lesquelles un MRI n'a pas présenté d'information supplémentaire à la liste de pages web habituellement présentée. De plus, les requêtes considérées seront celles pour lesquelles une page web a été consultée par l'usager suite à la soumission de sa requête. [1]

1.1.1 Classification non-supervisée

Le premier objectif est de pouvoir identifier, parmi un ensemble de requêtes donné, des groupes de requêtes issues du même domaine, [2] et ce sans avoir de connaissance a priori sur la nature du domaine de chacun des groupes qui seront formés. Ce genre de problème est habituellement référé par le terme de classification non-supervisée. Les résultats fournis par ce genre d'étude pourraient servir à l'amélioration du système d'un MRI de deux façons différentes :

1. Ils rendraient possible la détection de domaines de recherche pour lesquels le système actuel du MRI n'est pas entraîné à fournir directement les informations demandées, mais pour lesquels il existe un nombre significatif de requêtes traitant de ces derniers qui ont été soumises au système.
2. Comme les requêtes étudiées sont des requêtes pour lesquelles le système du MRI

1. Les adresses de pages web visitées sont habituellement des informations qui sont contenues dans les *logs* de moteur de recherche.

2. Tout au long de ce document, le «domaine» d'une requête fera référence au sujet dont elle traite. Des exemples de ce qu'on entend ici par domaine sont : musique, cinéma, santé ou automobile.

n'a pas présenté d'information supplémentaire à l'usager, il serait alors possible de détecter des groupes de requêtes issues de domaines qui sont supposés être pris en charge par le système du MRI, mais pour lesquelles ce dernier n'a pas fourni d'information additionnelle. L'étude de ces groupes de requêtes pourrait alors fournir des pistes sur des aspects du système qui pourraient être améliorés.

1.1.2 Détection d'entités nommées

Comme il a été mentionné précédemment, certains moteurs de recherche, qui sont ici référés par le terme MRI, tentent maintenant de fournir directement, pour les requêtes auxquelles ce genre de procédé s'applique, l'information demandée par l'usager par le biais de sa requête. Pour être capable de bien répondre à ce besoin, on suppose que ce genre de système doit, entre autres, être capable de découvrir les entités nommées contenues dans la requête émise par l'usager et d'en identifier leurs types. Cette tâche est habituellement référée par le terme *Named Entity Recognition and Classification* (NERC). En général, les types d'entités nommées à découvrir et à classifier diffèrent d'un problème à l'autre selon les besoins du système. Dans le contexte de la recherche web, on pourrait vouloir d'un système de NERC qu'il soit capable d'identifier des mentions d'entités nommées telles que, par exemple, des noms d'artistes, d'albums, de chansons, d'athlètes, de restaurants, de compagnies diverses ou de villes et villages.

De façon générale, deux grands types de technique peuvent être utilisés pour la conception un système de NERC. Le premier type regroupe les techniques basées sur des ensembles de règles grammaticales et syntaxiques qui ont été construites manuellement pour chaque type d'entité nommée considéré. Si un segment de texte respecte une des règles construites, alors il sera marqué comme correspondant à une entité nommée du type associé à la règle en question. Un exemple d'un système utilisant ce genre de technique peut être trouvé dans le travail de Sekine et al. [22]. Le deuxième type regroupe les techniques basées sur des modèles statistiques (e.g. Modèle de Markov Caché (MMC)[5], *Maximum Entropy Model* (MEM)[8] ou encore *Conditional Random Field* (CRF)[18]) qui seront entraînés avec un ensemble de textes dans lesquels les entités nommées à détecter ont déjà été identifiées et classées. Après avoir été

entraînés sur ces données, les modèles statistiques peuvent être utilisés pour identifier et classifier les entités nommées présentes dans un segment de texte donné. Il existe également des systèmes hybrides qui utilisent à la fois un ensemble de règles grammaticales et syntaxiques et un ou des modèle(s) statistique(s) pour effectuer cette tâche [25]. De plus, peu importe la manière de procéder, il n'est pas rare de voir ce genre de système utiliser un dictionnaire contenant une liste d'entités nommées associées à leurs types afin d'identifier directement les mentions de ces entités nommées dans les textes considérés.

Il est important de souligner que ce qui est ici proposé n'est pas un système de NERC, mais une technique d'analyse des *logs* d'un MRI muni d'un système de NERC qui aura pour but de découvrir de l'information pouvant être utilisée pour améliorer le système de NERC en question. Plus précisément, la technique proposée vise à effectuer la détection *a posteriori* des entités nommées qui sont contenues dans une requête et qui auraient pu ne pas être détectées par le système de NERC du MRI lors de l'émission de la requête. Afin de pouvoir détecter ces entités nommées, la technique proposée fera l'utilisation du texte de la page web visitée par l'usager suite à l'émission de sa requête. En supposant que le système de NERC utilisé par le MRI est basé sur un ou des modèle(s) statistique(s) nécessitant des données d'entraînement et qu'il fait également appel à un dictionnaire d'entités nommées, les entités nommées détectées par la technique ici proposée pourraient être utilisées pour améliorer le système de NERC en place de deux façons :

1. Ajouter de nouvelles entités nommées au dictionnaire d'entités nommées présentement utilisé par le système.
2. Effectuer l'annotation partielle de nouvelles données d'entraînement pour le ou les modèle(s) statistique(s) utilisé(s) par le système.

1.2 Présentation des données

Le jeu de données qui sera étudié comporte 29 618 requêtes distinctes qui ont été soumises à un certain moteur de recherche, et ce de manière vocale, impliquant ainsi un système de reconnaissance de la voix. De plus, étant donnée une requête, le jeu

Nombre de mots	Fréquence	Pourcentage	Cumulatif
2	6942	23,44%	23,44%
3	6238	21,06%	44,50%
4	4801	16,21%	60,71%
5	3751	12,66%	73,37%
6	2285	7,71%	81,09%
1	1895	6,40%	87,49%
7	1469	4,96%	92,45%
8	893	3,02%	95,46%
9	555	1,87%	97,34%
10-30	789	2,66%	100,0%
Total	29618		

Tableau 1.1 – Distribution du nombre de mots par requête

de données associe également à cette requête un ensemble contenant les URLs des pages web qui ont été consultées par les utilisateurs ayant soumis cette requête au moteur de recherche. Une brève description de la composition de ces requêtes sera présentée dans la section 1.2.1. Puis, une description des URLs contenus dans la base de données et de leurs interactions avec les requêtes sera présentée à la section 1.2.2.

1.2.1 Requêtes

Une brève description de la composition des 29 618 requêtes étudiées sera présentée dans cette sous-section afin de permettre une meilleure compréhension et une analyse plus éclairée des résultats qui seront présentés dans le chapitre 3.

Nombre de mots par requête : Les requêtes contenues dans le jeu de données étudié sont en général plutôt courtes. En effet, la moyenne du nombre de mots par requête pour l'ensemble de 29 618 requêtes étudié est de $3,8$ et son écart-type est de $2,3$. Pour un aperçu un peu plus détaillé, le Tableau 1.1 présente la distribution du nombre de mots des 29 618 requêtes.

Taille du vocabulaire : Les 29 618 requêtes étudiées forment un vocabulaire de 19 583 mots. De plus, il peut être intéressant de remarquer qu'un peu plus de 90% de

ces 19 583 mots sont des mots qui ne se retrouvent que dans neuf requêtes ou moins. Ceci indique une grande diversité des requêtes dans l'ensemble étudié.

Note : Le fait que le nombre de mots par requête soit en moyenne relativement bas et que la taille du vocabulaire engendré par l'ensemble des 29 618 requêtes soit relativement grande mène à penser qu'une technique de classification non-supervisée ne se basant que sur les mots composant les requêtes pour effectuer sa classification pourrait ne pas donner de bons résultats. Des résultats renforçant cette intuition seront présentés à la section 3.2.

1.2.2 Pages web consultées

Au minimum un URL de page web est attaché à chaque requête contenue dans le jeu de données étudié. Il est cependant possible qu'une requête se voie associer à plus d'un URL ou encore se voie associer plus d'une fois au même URL. Pour donner une idée de la fréquence où ceci se produit, la distribution du nombre de URLs associés à chaque requête est présentée dans le Tableau 1.2. L'information importante à retenir ici est que la majorité, 97% des requêtes étudiées ne sont associées qu'à un seul URL. Il est également intéressant de mentionner que le nombre de URLs uniques contenus dans le jeu de données étudié est de 29 935.

Note importante : L'information supplémentaire apportée par les URLs des pages web consultées sera utilisée dans la plupart des expériences présentées dans ce document. Une hypothèse importante qui est faite lorsque cette information est utilisée est que la page web consultée par l'utilisateur est cohérente avec l'intention de la requête formulée par ce dernier. Un humain a testé la véracité de cette hypothèse sur un sous-ensemble de cent paires (requête, URL). Quatre-vingt-huit de ces cent requêtes ont été identifiées comme étant pertinentes avec au moins un des URLs leur correspondant. Il faut donc garder en tête que l'information supplémentaire considérée est bruitée et que ce bruit aura un certain impact sur les résultats obtenus par les techniques qui utilisent cette information.

Nombre de URLs	Fréquence	Pourcentage	Cumulatif
1	28731	97,01%	97,01%
2	643	2,17%	99,18%
3	133	0,45%	99,63%
4	55	0,19%	99,81%
5	14	0,05%	99,86%
7	8	0,03%	99,89%
8	8	0,03%	99,91%
6	6	0,02%	99,93%
13	3	0,01%	99,94%
9	2	0,01%	99,95%
10	2	0,01%	99,96%
11	2	0,01%	99,96%
17	2	0,01%	99,97%
21	2	0,01%	99,98%
15	1	0,00%	99,98%
19	1	0,00%	99,98%
23	1	0,00%	99,99%
35	1	0,00%	99,99%
99	1	0,00%	99,99%
130	1	0,00%	99,99%
194	1	0,00%	100,00%
Total	29618		

Tableau 1.2 – Distribution du nombre de URLs par requête

1.3 Techniques existantes

Comme mentionné précédemment, le travail présenté dans ce document poursuit deux objectifs différents et indépendants qui ont tous deux comme but d'extraire des informations qui pourraient servir à l'amélioration du système de TALN du MRI à partir des *logs* de ce dernier. Le premier objectif vise à effectuer la classification non-supervisée des requêtes soumises à un MRI en se servant de la connaissance des pages web visitées par les utilisateurs et le deuxième objectif vise à effectuer la détection de certaines des entités nommées présentes dans les requêtes contenues dans les *logs* d'un MRI, et ce toujours en utilisant la connaissance des pages web visitées par les utilisateurs. Les sections 1.3.1 et 1.3.2 discuteront certaines techniques qui ont déjà été développées pour atteindre des objectifs similaires.

1.3.1 Classification non-supervisée

Le problème de classification non-supervisée de requêtes de recherche web a déjà retenu l'attention de plusieurs chercheurs. Parmi eux, certains ont également songé à développer des méthodes utilisant les pages web visitées par les utilisateurs pour améliorer leurs résultats. Cette section fera une brève description de certaines de ces méthodes et soulèvera quelques points faibles et points forts de ces dernières.

Beeferman et Berger [4] présentent une technique de classification non-supervisée qui ne se sert ni du contenu de la requête ni de celui de la page web visitée, mais plutôt de la façon dont les requêtes et les URLs sont associés entre eux. Cet algorithme sera présenté plus en détail à l'annexe A de ce document. Un des avantages de cette technique est qu'elle ne nécessite pas l'extraction du texte des pages web qui ont été visitées, ce qui peut être une tâche fastidieuse. Il est cependant nécessaire, pour que cette technique mène à des résultats intéressants, qu'une grande quantité des URLs contenus dans l'ensemble de données étudié soient associés à plusieurs requêtes. Dans le cas contraire, le résultat produit par cet algorithme contiendra une très grande quantité de petits *clusters* qui pourraient, dans le pire des cas, ne contenir qu'une seule requête.

Wen et Zhang [30] présentent et comparent plusieurs fonctions pour calculer la similarité entre deux requêtes étant données les pages web visitées par l'utilisateur lors de l'émission de ces requêtes. En particulier, ils mentionnent l'idée d'utiliser une technique standard de classification non-supervisée pour effectuer la classification des pages web visitées et d'utiliser ensuite le résultat obtenu pour calculer la similarité entre deux requêtes. Cependant, aucune expérimentation utilisant cette idée n'est présentée dans leur article. Ils présentent cependant des résultats de techniques de classification non-supervisée utilisant l'adresse des pages web visitées et une catégorisation déjà préétablie de ces dernières pour effectuer la classification des requêtes. Leurs résultats semblent indiquer que la considération des pages web visitées apporte un certain gain dans la qualité des groupes de requêtes obtenus. Par contre, tout comme pour la technique présentée par Beeferman et Berger [4], il est nécessaire qu'une grande quantité des URLs contenus dans l'ensemble de données étudié soient associés à plusieurs requêtes pour que cette technique mène à des résultats intéressants.

1.3.2 Détection d'entités nommées

L'identification non-supervisée et l'identification semi-supervisée d'entités nommées dans des requêtes de recherche web en utilisant les *logs* d'un moteur de recherche sont des sujets d'intérêt sur lesquels certains chercheurs se sont penchés.

Pasca [20] présente une technique permettant, en utilisant une liste d'entités nommées dont on connaît déjà le type, d'extraire, d'un ensemble de requêtes, d'autres entités nommées du même type. Selon les résultats qu'il présente, cette méthode semble donner de bons résultats. Elle permet également de découvrir certains patrons de mots qui entourent les types d'entités nommées d'intérêt. Cependant, étant donné le fonctionnement de l'algorithme, la méthode semble nécessiter un ensemble de requêtes d'assez grande taille. De plus, cette méthode ne permet pas d'identifier des entités nommées qui seraient seules (sans contexte) dans une requête[3]. Un

3. Ce type d'entité nommée sera référé par le terme Entité Nommée Solitaire (ENS) dans le reste de ce document.

exemple d'une telle requête pourrait être «Ariane Moffatt», ou encore «Canadiens de Montréal».

Xu et al. [31] proposent une technique qui permet, en utilisant une liste d'entités nommées dont les types sont connus, d'extraire, d'un ensemble de requêtes, d'autres entités nommées ayant le même type que celles contenues dans la liste fournie. Cette technique se base sur un modèle génératif basé sur l'algorithme LDA qui prend en compte à la fois le contexte des entités nommées et l'adresse de la page web visitée par l'émetteur de la requête contenant l'entité nommée. Étant donné la nécessité de fournir en entrée une liste d'entités nommées dont le type est déjà connu, cet algorithme fait également partie de la classe des algorithmes semi-supervisées. L'avantage qu'il présente face au travail de Pasca [20] est qu'il permet de mieux gérer l'ambiguïté face aux types de certaines entités nommées. Par contre, tout comme la méthode présentée par Pasca [20], cette méthode ne semble pas pouvoir identifier des ENSs.

1.4 Description des méthodes proposées

Cette section décrira brièvement les méthodes proposées dans le présent travail pour résoudre les deux problèmes distincts adressés dans ce dernier, soit la classification non-supervisée de pages web et la détection non-supervisée d'entités nommées dans des requêtes de recherche web. Les sections 1.4.1 et 1.4.2 présenteront respectivement les informations relatives aux techniques proposées pour effectuer la classification et la détection d'entités nommées incluant les avantages et désavantages des techniques proposées. Les détails du fonctionnement de ces techniques seront présentés dans le chapitre 2.

1.4.1 Classification non-supervisée

Comme il a été mentionné précédemment, la technique de classification non-supervisée proposée tire avantage de la connaissance des pages web qui ont été visitées par les émetteurs des requêtes à classifier. Cependant, contrairement au travail de Beeferman et Berger [4], la technique de classification non-supervisée proposée dans ce mémoire

n'utilisera pas directement les URLs des pages visitées, mais fera plutôt usage du texte qui est contenu dans ces dernières. Le processus partant d'un ensemble de requêtes, chacune d'entre elles jumelée à un ou plusieurs URL(s) correspondant aux pages web qui ont été visitées par leurs émetteurs, pour produire finalement un partitionnement des requêtes de l'ensemble étudié peut être décrit par un enchaînement de trois étapes distinctes :

1. L'extraction du texte des pages web associées aux URLs contenus dans l'ensemble de données considéré.
2. La classification non-supervisée des URLs considérés basée sur le texte des pages web qu'ils représentent.
3. La classification non-supervisée des requêtes basée sur le partitionnement des URLs qui leur sont associés obtenu à l'étape précédente.

La Figure 1.1 représente l'enchaînement de ces trois étapes en spécifiant les entrées et sorties de chacune d'elles. Les détails de chacune de ces étapes sont présentés dans les sections 2.4.1, 2.4.2 et 2.4.3

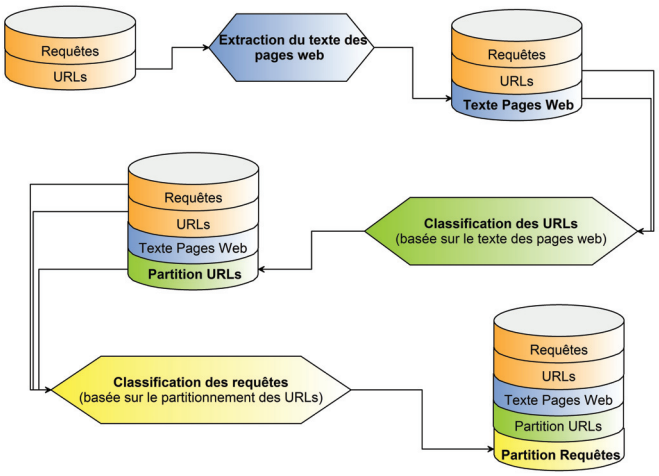


Figure 1.1 – Diagramme du processus de classification non-supervisée des requêtes de recherche web qui est proposé dans ce mémoire.

Les avantages apportés par la technique ici proposée sont les suivants :

- Elle crée des *clusters* de requêtes qui sont sémantiquement similaires sans toutefois être syntaxiquement similaires.
- Elle ne nécessite pas qu'une grande quantité des URLs contenus dans l'ensemble de données étudié soient associés à plusieurs requêtes pour fournir des résultats intéressants.
- Elle crée automatiquement des noms de *cluster* qui peuvent être facilement interprétés par un humain.
- Elle permet, moyennant une légère intervention humaine, de cibler certains domaines de recherche (e.g. musique, cinéma, automobile) qui sont particulièrement intéressants pour son utilisateur.

Cette technique comprend également certains désavantages dont :

- La pureté des *clusters* produits n'est pas toujours très élevée.
- Il est nécessaire de faire l'extraction du texte des pages web visitées pour appliquer cette technique. Ce processus peut être fastidieux.
- Cette technique semble permettre de regrouper un certain nombre requêtes selon leurs domaines. Cependant, elle ne semble pas permettre de, à l'intérieur d'un même domaine, faire des sous-groupes de requêtes qui varieraient selon leurs intentions. Par exemple, il sera difficile pour cette technique de faire la distinction entre un groupe de requêtes visant à trouver l'emplacement d'un restaurant et un autre visant à connaître les évaluations d'anciens clients pour un certain restaurant.

1.4.2 Détection d'entités nommées

Le processus de détection d'entités nommées décrit dans ce document comporte deux techniques. Ces deux techniques sont basées sur une comparaison des mots contenus dans le texte des pages web avec les mots contenus dans les requêtes qui leur sont associées. Les avantages de ces deux techniques sont qu'elles permettent la détection de ENSs et qu'elles sont complètement non-supervisées (elles ne nécessitent pas de listes d'entités nommées en entrée). De plus, une des deux techniques, qui sera présentée à la section 2.6.2, permet également d'identifier le type des entités nommées détectées, et ce avec une grande précision. Cependant cette technique ne

détecte que très peu d'entités nommées parmi celles présentes dans l'ensemble de requêtes considéré. La deuxième technique, présentée à la section 2.6.3, détecte pour sa part un plus grand nombre d'entités nommées, mais elle le fait au coût d'une baisse significative de précision. De plus, elle ne permet pas d'identifier le type des entités nommées détectées.

Il est également important de mentionner que les deux méthodes présentées permettent de détecter des entités nommées qui pourraient contenir des erreurs d'orthographe et de fournir les versions correctement orthographiées de ces entités nommées. Ce genre de détection d'erreurs peut être particulièrement utile dans le cas où les requêtes sont données vocalement par l'utilisateur et donc traduites à l'écrit par un système de reconnaissance vocale. En effet, ces techniques pourraient permettre d'identifier des erreurs fréquemment commises par le système de reconnaissance vocale et donc donner des pistes d'améliorations pour le système en place.

Chapitre 2

Cadre théorique

Ce chapitre présentera les détails des techniques utilisées dans le présent travail pour atteindre les deux objectifs fixés. Les techniques et les concepts reliés à la classification non-supervisée des requêtes seront présentés dans les sections 2.1, 2.2, 2.3, 2.4 et 2.5. Les techniques relatives à la détection non-supervisée d'entités nommées seront présentées à la section 2.6.

2.1 *Vector Space Model*

Dans certains algorithmes de classifications non-supervisées il est nécessaire de pouvoir représenter les objets à classifier par des vecteurs de nombres réels. Lorsque, comme c'est le cas dans ce travail, les données étudiées sont des documents textes (requêtes ou pages web), il est courant de voir ces derniers représentés à l'aide d'un modèle général appelé *Vector Space Model* (VSM). Ce modèle sera présenté dans cette section.

Pour le restant de cette section, le symbole $\mathcal{D}$ représentera le corpus étudié. Ce corpus contiendra M documents qui engendreront un vocabulaire $\mathcal{V}$ de N mots.

Le VSM représente chaque document du corpus $\mathcal{D}$ par un vecteur dans $\mathbb{R}_+^N$ et associe chaque mot du vocabulaire $\mathcal{V}$ à une des N dimensions de $\mathbb{R}_+^N$. Pour un document

donné, la valeur de la $n^{\text{ième}}$ composante du vecteur le représentant correspond à un poids associé au mot relatif à la dimension n. La définition de ce poids dépend du modèle choisi. Un modèle simple pourrait restreindre les valeurs possibles à l'ensemble $\{0, 1\}$, un poids de 1 caractérisant la présence d'un mot et un poids de 0 caractérisant son absence. Un modèle plus complexe pourrait étendre les valeurs possibles à $\mathbb{R}^+$ et définir une certaine fonction d'importance pour un mot par rapport à un document.

Avec une telle représentation, le corpus de document $\mathcal{D}$ peut être représenté par une matrice $M \times N$ où chaque ligne correspond à la représentation vectorielle d'un des documents du corpus. Il est important de noter qu'une des conséquences de l'utilisation de cette représentation vectorielle est la perte de l'ordre d'apparition des mots dans les documents.

2.1.1 *Term Frequency-Inverse Document Frequency*

Un modèle souvent utilisé dans des applications de Recherche d'Information (RI) est le modèle *Term Frequency-Inverse Document Frequency* (TF-IDF). Le poids que ce modèle associe à un certain mot pour un certain document est défini comme le produit de deux valeurs, dont l'une donne au mot considéré un certain poids relatif à sa fréquence d'apparition dans le document considéré (c.-à-d. poids local) et l'autre donne au mot considéré un certain poids relatif au nombre de documents du corpus dans lesquels on retrouve ce mot (c.-à-d. poids global).

Considérons la fonction

$$tf_d : \mathcal{V} \longrightarrow \mathbb{N}$$

donnant pour chaque mot du vocabulaire $\mathcal{V}$, sa fréquence d'apparition dans le document $d \in \mathcal{D}$ («term-frequency») et la fonction

$$df : \mathcal{V} \longrightarrow \mathbb{N}$$

donnant pour chaque mot du vocabulaire $\mathcal{V}$, le nombre de document du corpus $\mathcal{D}$ le contenant («document-frequency»). Ces deux fonctions seront utiles dans la définition du poids local et du poids global du modèle TF-IDF.

Poids local Soit un certain mot $w \in \mathcal{V}$ et un certain document $d \in \mathcal{D}$, le poids local associé à w pour d sera fonction de la fréquence d'apparition de w dans d. La définition de cette fonction dépendra du contexte dans lequel la représentation TF-IDF est utilisée. Des exemples communs sont :

$$poidsLocal(w, d) = \begin{cases} 1 & \text{si } tf_d(w) > 0 \\ 0 & \text{sinon} \end{cases} \tag{2.1}$$

$$poidsLocal(w, d) = tf_d(w) \tag{2.2}$$

Cependant, afin de diminuer le poids d'un mot qui apparaîtrait très fréquemment dans le document considéré, la définition suivante est souvent utilisée :

$$poidsLocal(w, d) = \log(tf_d(w) + 1) \tag{2.3}$$

Comme la fonction logarithme est croissante, mais croît moins rapidement que la fonction identité sur l'intervalle $[1, \inf[$, on obtient l'effet d'amortissement désiré. Notons qu'il est nécessaire de prendre le logarithme de $tf_{d_i}(w_j) + 1$ et non de $tf_{d_i}(w_j)$ uniquement puisque $tf_{d_i}(w_j)$ prend ses valeurs dans l'intervalle $[0, \inf[$ et qu'il est nécessaire de prendre le logarithme de valeurs se situant dans l'intervalle $[1, \inf[$ afin que la valeur du poids local soit positive.

Poids global Le poids global est relatif à tout le corpus et doit donner un poids plus élevé aux mots qui apparaissent dans peu de documents qu'à ceux qui apparaissent dans beaucoup de documents. Ce poids est référé comme étant la fréquence inverse de document et est défini comme :

$$poidsGlobal(w) = \log\left(\frac{M}{df(w)}\right) \tag{2.4}$$

Poids total Peut importe la définition des fonctions de poids local et de poids global, la fonction du poids total d'un mot w pour un document d sera donnée par :

$$tfidf(w, d) = poidsLocal(w, d) * poidsGlobal(w)$$

2.2 *K-Means* et quelques variantes

Cette section présente l'algorithme de classification non-supervisée *K-Means* et ses variantes *K-Means++* et *Sperical K-Means*, des algorithmes simples, mais fréquemment utilisés. Il est important de mentionner que la méthode de classification non-supervisée proposée dans ce travail n'utilise pas l'algorithme *K-Means*. Cependant, des résultats d'expérimentations utilisant cet algorithme et ces variantes sur les requêtes étudiées seront présentés dans le chapitre 3 afin d'illustrer certains avantages de la technique proposée. Pour cette raison, les algorithmes *K-Means*, *K-Means++* et *Spherical K-Means* sont présentés dans cette section.

Avant de débuter l'explication du fonctionnement de l'algorithme *K-Means* de base et de ses variantes, il faut mentionner que ce type d'algorithme est applicable sur des objets pouvant être représentés sous forme de vecteurs de nombres réels. Ainsi, lorsque ce type d'algorithme est utilisé pour classifier un ensemble de segments de textes, ces derniers sont souvent représentés par le biais d'un modèle TF-IDF tel que présenté dans la section 2.1.

2.2.1 *K-Means* de base

Pour un nombre K donné, *K-Means* vise à séparer en K groupes disjoints les objets d'un ensemble de données de telle sorte que les objets appartenant à un même groupe ont tendance à être plus similaires les uns aux autres que les objets appartenant à des groupes différents.

Plus précisément, *K-Means* est un algorithme itératif conçu pour trouver une partition $\mathcal{C} = \{\mathcal{C}_1,\ \mathcal{C}_2,\ \ldots\ \mathcal{C}_\mathrm{K}\}$ d'un ensemble $\mathcal{X} = \{\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots\ \boldsymbol{x}_\mathrm{M}\}$ de points dans $\mathbb{R}^N$

qui minimise la Somme du Carré des Erreurs (SCE) donnée par :

$$\sum_{k=1}^{K} \sum_{\boldsymbol{x} \in \mathcal{C}_k} ||\boldsymbol{x} - \boldsymbol{c_k}||_2^2 \tag{2.5}$$

où $\boldsymbol{c_k} \in \mathbb{R}^N$ est le représentant de l'élément $\mathcal{C}_k$ de la partition $\mathcal{C}$. Pour ce faire, l'algorithme procédera d'abord à une initialisation des K représentants $\boldsymbol{c}_1, \boldsymbol{c}_2, \dots \boldsymbol{c}_K$ en choisissant aléatoirement [1] K points appartenant à $\mathcal{X}$. Il alternera ensuite entre deux séquences d'opérations, le partitionnement et le calcul des représentants, jusqu'à ce qu'un certain critère d'arrêt indiquant la convergence de l'algorithme soit satisfait.

Partitionnement Cette étape crée une nouvelle partition de $\mathcal{X}$ en assignant chaque point $\boldsymbol{x} \in \mathcal{X}$ au sous-ensemble $\mathcal{C}_k$ dont il est le plus près (c.-à-d. au sous-ensemble $\mathcal{C}_k$, où $k = \underset{l \in \{1,2,\dots,K\}}{\arg\min} ||\boldsymbol{x} - \boldsymbol{c_l}||_2$). Notons que cette façon de partitionner assure qu'après chaque étape de partitionnement, la SCE aura diminuée ou sera restée la même.

Calcul des représentants Cette étape effectue le calcul des nouveaux représentants des éléments de la partition nouvellement créée. Lorsque la distance euclidienne est utilisée, la valeur du représentant d'un sous-ensemble $\mathcal{C}_k$ est donnée par la moyenne des points de $\mathcal{X}$ que ce sous-ensemble contient. Autrement dit :

$$\boldsymbol{c}_k = \frac{1}{|\mathcal{C}_k|} \sum_{\boldsymbol{x} \in \mathcal{C}_k} \boldsymbol{x}$$

La justification pour ce choix est la suivante : en considérant l'expression (2.5) comme une fonction de $\boldsymbol{c}_1, \boldsymbol{c}_2, \dots \boldsymbol{c}_K$ (c.-à-d. la partition de $\mathcal{X}$ est fixée), alors il est possible de prouver de façon analytique [26, p.513] que les valeurs de $\boldsymbol{c}_k$ minimisant cette fonction sont données par $\boldsymbol{c}_k = \frac{1}{|\mathcal{C}_k|} \sum_{\boldsymbol{x} \in \mathcal{C}_k} \boldsymbol{x}$. Ainsi, après cette étape de calcul des représentants, la SCE aura diminuée ou sera restée la même.

Critère d'arrêt Différents critères d'arrêt peuvent être utilisés :

1. L'algorithme de base propose une initialisation aléatoire, mais certaines techniques d'initialisation plus sophistiquées ont été développées, dont une qui sera présentée à la section 2.2.2.

- aucun ou peu de changement entre la partition de l'itération courante et celle de l'itération précédente
- aucun ou peu de variation entre la valeur de la SCE de l'itération courante et celle de l'itération précédente
- un nombre maximal d'itérations a été atteint

Convergence Comme il existe un nombre fini de partitions différentes de $\mathcal{X}$ et que l'algorithme décrit ici fait décroître la SCE de façon monotone après chaque opération, la convergence de l'algorithme vers une solution après un nombre fini d'opérations est assurée de se produire. Si il n'y a pas de nombre maximum d'itérations et que l'algorithme stoppe lorsqu'il n'y a plus aucun changement dans le partitionnement ou dans la valeur de la SCE d'une itération à l'autre, cette solution, sauf dans de rares cas,[2] sera un minimum local de la SCE. La preuve de convergence vers un minimum local est présentée dans [23].

2.2.2 *K-Means++*

L'algorithme *K-Means++*[3] est identique à l'algorithme *K-Means* de base présenté à la section 2.2.1 à l'exception de la méthode d'initialisation des représentants des K *clusters*. Cette méthode consiste à choisir successivement les K représentants de façon aléatoire parmi les éléments de l'ensemble à partitionner, et ce selon une distribution de probabilité favorisant les objets les plus éloignés des représentants déjà sélectionnés. Dans leur papier, Arthur et Vassilvitskii [3] testent cette nouvelle méthode d'initialisation sur plusieurs jeux de données et obtiennent de meilleurs résultats en terme de précision et de temps d'exécution de l'algorithme. L'algorithme 1 présente les détails

2. Si les points de $\mathcal{X}$ sont colinéaires et que le nombre de points constituant $\mathcal{X}$ est pair, la solution n'est pas assurément un minimum local.

de cette technique.

Algorithme 1 : Algorithme d'initialisation de K-Means++

Entrées :

– $\mathcal{X}$ un ensemble de données de taille N

– K le nombre de groupes désirés

Sorties :

– $\mathcal{S}$ un ensemble de K représentant initiaux

Initialiser $\mathcal{S} = \{\}$

Choisir $c_1 \in \mathcal{X}$ selon $\mathbb{P}(c_1 = x) = \frac{1}{N}$

Ajouter c_1 à $\mathcal{S}$

pour *i=2* **à** K **faire**

  Choisir $c_i \in \mathcal{X}$ selon $\mathbb{P}(c_i = x) \propto \min_{c \in \mathcal{S}} ||c - x||_2$

  Ajouter c_i à $\mathcal{S}$

fin

2.2.3 *Spherical K-Means*

Sperical K-Means est un algorithme itératif conçu pour trouver une partition $\mathcal{C} = \{\mathcal{C}_1,\ \mathcal{C}_2,\ \ldots\ \mathcal{C}_\mathrm{K}\}$ d'un ensemble $\mathcal{X} = \{\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots\ \boldsymbol{x}_\mathrm{M}\}$ de points dans $\mathbb{R}^N$ qui minimise la quantité suivante :

$$\sum_{k=1}^{K} \sum_{\boldsymbol{x} \in \mathcal{C}_k} 1 - cosine(\boldsymbol{x}, \boldsymbol{c_k}) \tag{2.6}$$

où $\boldsymbol{c_k} \in \mathbb{R}^N$ est le représentant de l'élément $\mathcal{C}_k$ de la partition $\mathcal{C}$ et $cosine(\cdot, \cdot)$ est la similarité cosinus définie pour deux vecteurs $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^N$ comme

$$cosine(\boldsymbol{u}, \boldsymbol{v}) = \frac{\langle \boldsymbol{u}, \boldsymbol{v} \rangle}{||\boldsymbol{u}||_2 \, ||\boldsymbol{v}||_2}$$

En fait, cet algorithme possède la même structure que l'algorithme *K-Means* à l'exception de ces deux détails :

– La fonction objective est de la même forme que celle de *K-Means*, mais une mesure de dissemblance basée sur la similarité cosinus est utilisée au lieu de la distance euclidienne.

– Le calcul du représentant $\boldsymbol{c}_k$ d'un élément de la partition est donné par

$$\boldsymbol{c}_k = \frac{\sum\limits_{\boldsymbol{x}\in\mathcal{C}_k} \boldsymbol{x}/\left|\left|\boldsymbol{x}\right|\right|_2}{\left|\left|\sum\limits_{\boldsymbol{x}\in\mathcal{C}_k} \boldsymbol{x}/\left|\left|\boldsymbol{x}\right|\right|_2\right|\right|_2} \tag{2.7}$$

Ce nouveau calcul de représentant est directement relié à la définition de la fonction objective. En effet, tout comme pour l'algorithme *K-Means*, pour une partition $\mathcal{C}$ de $\mathcal{X}$ donnée, la valeur des représentants de chaque élément de la partition sera définie de façon à minimiser la fonction objective. Dans ce cas, les valeurs des K représentants minimisant la fonction objective sont données par l'équation (2.7). Liu et al. [16] font la preuve que cette définition des K représentants minimise bien la fonction objective (2.6) pour une partition de $\mathcal{X}$ fixée.

Cet algorithme est particulièrement utile lorsque la similarité cosinus modélise bien la similarité entre deux données de l'ensemble étudié. Basé sur cette similarité, il est alors possible de définir la dissemblance cosinus entre deux vecteurs $u, v \in \mathbb{R}^N$ par :

$$1 - cosine(u, v) = 1 - \frac{\langle \boldsymbol{u}, \boldsymbol{v} \rangle}{\left|\left|\boldsymbol{u}\right|\right|_2 \left|\left|\boldsymbol{v}\right|\right|_2}$$

où $\langle \cdot, \cdot \rangle$ désigne le produit scalaire et $\left|\left|\cdot\right|\right|_2$ désigne la norme euclidienne. Notons que cette mesure est bornée supérieurement par 1 et inférieurement par 0. Elle est entre autres souvent utilisée pour calculer la dissemblance entre deux documents représentés par un VSM comme le modèle TF-IDF. En effet, lorsque le modèle TF-IDF est utilisé pour représenter un corpus de documents, les documents seront représentés par des vecteurs qui seront la plupart du temps très creux (c.-à-d. comportent beaucoup de composantes nulles) puisqu'en général, la taille du vocabulaire engendré par le corpus dépassera de beaucoup le nombre de mots utilisés dans un document quelconque du corpus. Ainsi, la mesure de dissemblance cosinus semble mieux adaptée à cette situation qu'une distance de Minkowski comme la distance euclidienne.

Voici deux exemples qui appuie la préférence de la dissemblance cosinus à une distance de Minkowski pour le calcul de dissemblance entre deux documents représentés

par un VSM du type TF-IDF.

1. Si deux documents n'ont aucun mot en commun, la similarité cosinus entre ces dernier sera de 0 et donc une dissemblance cosinus de 1. Ceci semble cohérent, jusqu'à un certain point, avec le fait que les deux documents n'ont aucun mot en commun. Par contre, pour la même situation, il n'est pas possible de prédire une valeur de distance pour une distance de Minkowski donnée, puisque ce genre de distance n'est pas bornée supérieurement et qu'elle dépend de la fréquence des mots dans chaque document. Ainsi, dans ce cas particulier, la dissemblance cosinus semble plus cohérente qu'une distance de Minkowski quelconque.
2. En considérant une distance de Minkowski, deux documents ayant des mots en commun pourraient se voir attribuer la même distance que deux documents n'ayant aucun mot en commun. Ce comportement n'est pas observé lorsque la dissemblance cosinus est utilisée. En effet, il est impossible que deux documents ayant au moins un mot en commun se voient attribuer une similarité cosinus plus petite ou égale à 0, qui est la valeur de similarité entre deux documents qui n'ont aucun mot en commun.

2.3 *Latent Dirichlet Allocation*

La technique de classification non-supervisée proposée, qui sera présentée à la section 2.4, est majoritairement basée sur l'application de l'algorithme LDA et d'une de ses variantes nommée *Topic-In-Set Knowledge Latent Dirichlet Allocation* (TISK-LDA) [2] [1, Chapter 5] sur un ensemble de textes provenant de pages web. L'algorithme LDA est un algorithme de *topic modeling* qui a été développé en 2003 par Blei et al.[7]. Cette technique modélise la génération d'un ensemble de documents à l'aide d'un réseau bayésien. Dans ce dernier, chaque document est associé à une distribution de probabilité sur un ensemble fini de variables cachées nommées *topics* et chacun de ces *topics* est associé à une distribution de probabilité sur les mots du vocabulaire engendré par les documents du corpus. Cette section présentera la théorie et le fonctionnement de cet algorithme (LDA) et de la variante utilisée (TISK-LDA). Mais tout d'abord, la section 2.3.1 présentera une loi de probabilité, nommée loi de Dirichlet,

qui est largement utilisée dans la conception du modèle de l'algorithme LDA et la section 2.3.2 présentera la notation qui sera utilisée dans cette section.

2.3.1 Loi de Dirichlet

La loi de Dirichlet d'ordre K, notée $Dir(\boldsymbol{\alpha})$, est une loi de probabilité multivariée et continue qui est paramétrée par un vecteur de K composantes réelles et positives noté $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_K)$. Le support d'une loi de Dirichlet d'ordre K est

$$\mathcal{S}_K = \{\boldsymbol{x} \in \mathbb{R}^K | \sum_{i=1}^{K} x_i = 1 \text{ et } x_i \geq 0 \text{ pour } i = 1, 2, \ldots, K\}$$

et sa densité de probabilité est donnée par :

$$f(\boldsymbol{x}) = \begin{cases} \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} & \text{si } \boldsymbol{x} \in \mathcal{S}_{\mathcal{K}} \\ 0 & \text{sinon} \end{cases} \tag{2.8}$$

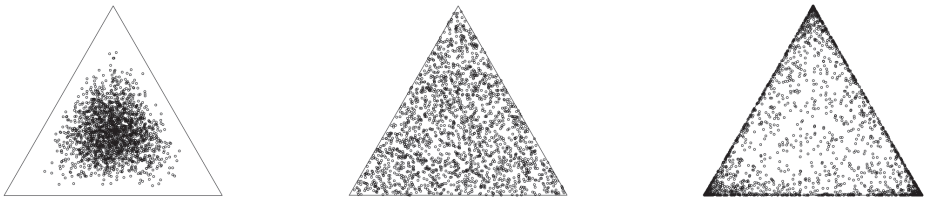
Le support $\mathcal{S}_K$ est en fait un $K-1$ simplex régulier. Pour $K = 3$, ce simplex peut être représenté par un triangle équilatéral où les trois sommets correspondent aux points $(1, 0, 0)$, $(0, 1, 0)$ et $(0, 0, 1)$ et le centre de gravité correspond au point $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. La Figure 2.1 représente des échantillons de 2000 points obtenus à partir de lois de Dirichlet d'ordre 3 pour différentes valeurs du paramètre $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$.

Lois marginales Il est utile de noter que les lois marginales des composantes individuelles d'une loi de Dirichlet sont des lois Bêta. Plus précisément, Soit

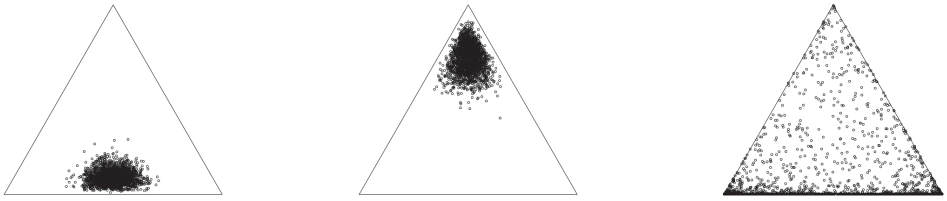
$$\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K) \sim Dir(\alpha_1, \alpha_2, \ldots, \alpha_K)$$

Alors

$$\mathbf{X}_i \sim Beta(\alpha_i, \left(\sum_{k=1}^{K} \alpha_k\right) - \alpha_i)$$



(a) $\boldsymbol{\alpha} = (5, 5, 5)$ (b) $\boldsymbol{\alpha} = (1, 1, 1)$ (c) $\boldsymbol{\alpha} = (0.3, 0.3, 0.3)$



(d) $\boldsymbol{\alpha} = (25, 25, 5)$ (e) $\boldsymbol{\alpha} = (5, 5, 25)$ (f) $\boldsymbol{\alpha} = (0.5, 0.5, 0.1)$

Figure 2.1 – Visualisation d'échantillons de 2000 points obtenus à partir de lois de Dirichlet d'ordre trois pour différentes valeurs du paramètre $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$

De plus, comme $\mathbf{X}_i \sim Beta(\alpha_i, \left(\sum_{k=1}^{K} \alpha_k\right) - \alpha_i)$, on a que

$$\mathrm{E}\left[\mathbf{X}_\mathrm{i}\right] = \frac{\alpha_i}{\sum_{k=1}^{K} \alpha_k} \tag{2.9}$$

2.3.2 Notation

Dans le restant de cette section, la notation suivante sera utilisée :

- $\mathcal{D} = \{\mathrm{d}_1,\ \mathrm{d}_2,\ \ldots\ \mathrm{d}_\mathrm{M}\}$: le corpus étudié
- $\mathcal{V} = \{\mathrm{v}_1,\ \mathrm{v}_2,\ \ldots\ \mathrm{v}_\mathrm{N}\}$: le vocabulaire engendré par le corpus $\mathcal{D}$
- $\mathcal{Z} = \{1, 2, \ldots K\}$: K identifiants faisant référence aux K *topics* du modèle.
- $\boldsymbol{w}_d = (w_{d,1}, w_{d,2}, \ldots, w_{d,N_d})$ les N_d mots formant le document d
- $\boldsymbol{z}_d = (z_{d,1}, z_{d,2}, \ldots, z_{d,N_d})$ les N_d identifiants des *topics* générant chacun des mots du document d

Dans cette section, chaque document $d \in \mathcal{D}$ sera associé à un identifiant unique donné par un entier entre 1 et M et chaque mot $v \in \mathcal{V}$ sera associé à un identifiant unique donné par un entier entre 1 et N. Lorsque les symboles d ou v sont utilisés comme indice pour faire référence à un document ou à un mot, ils feront alors référence à l'entier utilisé comme identifiant unique pour ces derniers.

De plus, il sera souvent question de distributions de probabilité sur les mots du vocabulaire $\mathcal{V}$ ainsi que sur les K *topics* du modèle. Les distributions de probabilité sur les mots du vocabulaire seront représentées par des vecteurs de N composantes et les distributions de probabilité sur les K *topics* du modèle seront représentées par des vecteurs de K composantes. Dans les deux cas, la $i^{\text{ième}}$ composante d'un vecteur donne la probabilité, selon la distribution représentée par le vecteur, du mot ou du *topic* associé à l'identifiant unique i.

2.3.3 Définition du modèle génératif

Étant donné :

- K : un nombre de *topics*
- $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ un vecteur de nombres réels positifs
- $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_K$: K distributions de probabilités sur les mots de vocabulaire $\mathcal{V}$

Le modèle tel que décrit par Blei dans le papier à l'origine de LDA [7] suppose le processus génératif suivant pour un document $d \in \mathcal{D}$:

1. Choix de la longueur du document, $N_d \sim Poisson(\xi)$
2. Choix d'une distribution sur les *topics*, $\boldsymbol{\theta}_d \sim Dirichlet(\boldsymbol{\alpha})$
3. Pour chacun des N_d mots du document :
   (a) Choix d'un *topic*, $z_{d,n} \sim Categorical(\boldsymbol{\theta}_d)$
   (b) Choix d'un mot, $w_{d,n} \sim Categorical(\boldsymbol{\beta}_{z_{d,n}})$

Notons ici que dans le processus génératif décrit ci-haut, il est supposé que la variable N_d suive une loi de Poisson de paramètre ξ, mais toute loi de probabilité discrète faisant un certain sens pour modéliser le nombre de mots contenu dans un document pourrait être utilisée. De plus, selon le processus génératif donné, cette variable est

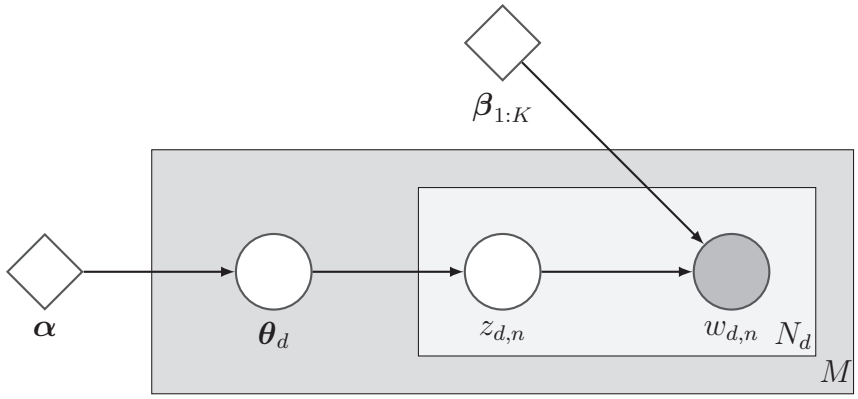


Figure 2.2 – Représentation graphique des relations entre les différentes variables du modèle *Latent Dirichlet Allocation* de base

complètement indépendante des autres variables du modèle et donc l'estimation du ou des paramètres associé(s) à la loi de probabilité choisie pour modéliser cette variable peut ce faire complètement indépendamment de l'estimation des autres paramètres du modèle. Dans le cas de la loi de Poisson, un estimateur du maximum de vraisemblance pourrait être utilisé et ainsi la moyenne des longueurs des documents du corpus étudié serait l'estimation du paramètre ξ. Comme cette partie du modèle est complètement indépendante du reste, son caractère aléatoire sera ignoré pour le restant de cette section.

La Figure 2.2 représente le processus génératif tel que décrit ci-haut, pour l'ensemble des documents du corpus. Cette figure met en évidence les différents niveaux du processus décrit par LDA. Le premier niveau est relatif au corpus en entier et contient les paramètres $\boldsymbol{\alpha}$ et $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_K$. Le deuxième niveau est relatif aux documents et contient les variables cachées $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_M$ qui seront prélevées pour chaque document du corpus. Le troisième niveau est relatif aux mots des documents et contient les variables cachées $\{z_{d,n}\}_{d\in\mathcal{D},n=1,2,\ldots,N_d}$ et les variables observées $\{w_{d,n}\}_{d\in\mathcal{D},n=1,2,\ldots,N_d}$ qui seront prélevées pour chaque mot de chaque document du corpus.

Blei propose également une version modifiée du modèle de base représenté dans la Figure 2.2. Ce modèle modifié ajoute une distribution *apriori* sur les distributions de probabilité $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_K$ du modèle. Cette modification a pour but de lisser les

distributions $\boldsymbol{\beta}_k$ afin qu'un modèle n'assigne pas une probabilité nulle à un document contenant des mots ne se retrouvant pas dans les documents utilisés pour l'entraînement du modèle. La distribution a priori choisie est une distribution de Dirichlet symétrique de paramètre η.[3] Et donc les $\boldsymbol{\beta}_k$ passent du statut de paramètres à celui de variables cachées et un nouveau paramètre η est ajouté au modèle. Avec l'ajout de ces distributions apriori sur les $\boldsymbol{\beta_k}$, la définition du modèle génératif, en omettant l'étape du choix de la longueur des documents pour la raison mentionnée plutôt dans cette section, devient

1. Pour $k = 1, 2, \ldots K$:
   (a) Choix d'une distribution sur les mots du vocabulaire, $\boldsymbol{\beta}_k \sim Dirichlet(\eta)$
2. Pour $d = 1, 2, \ldots M$:
   (a) Choix d'une distribution sur les *topics*, $\boldsymbol{\theta}_d \sim Dirichlet(\boldsymbol{\alpha})$
   (b) Pour chacun des N_d mots du document :
      i. Choix d'un *topic*, $z_{d,n} \sim Categorical(\boldsymbol{\theta}_d)$
      ii. Choix d'un mot, $w_{d,n} \sim Categorical(\boldsymbol{\beta}_{z_{d,n}})$

La Figure 2.3 présente le diagramme de ce modèle modifié.

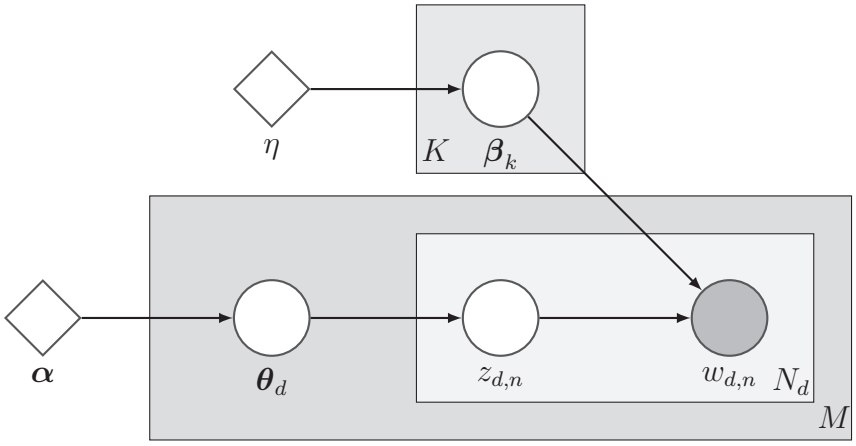


Figure 2.3 – Représentation graphique des relations entre les différentes variables du modèle *Latent Dirichlet Allocation*

3. Dans ce document, une loi de Dirichlet symétrique est une loi de Dirichlet dont les composantes du vecteur de paramètres ont toutes la même valeur.

2.3.4 Inférence

Afin de pouvoir représenter les documents comme une mixture de *topics* et de connaître, pour chaque *topic*, l'importance de chacun des mots du vocabulaire, la distribution *a posteriori* des variables cachées du modèle, $\mathbb{P}(\boldsymbol{\theta}_{1:M}, \boldsymbol{z}_{1:M}, \boldsymbol{\beta}_{1:K}|\boldsymbol{w}_{1:M}, \boldsymbol{\alpha}, \eta)$, doit être calculée.

En supposant la connaissance de cette distribution, le choix pourrait être fait de représenter la mixture de *topics* d'un document d par $\mathrm{E}[\boldsymbol{\theta}_{\mathrm{d}}]$ selon la distribution de probabilité marginale $\mathbb{P}(\boldsymbol{\theta}_d|\boldsymbol{w}_{1:M}, \boldsymbol{\alpha}, \eta)$ et pour un *topic* k donné, de représenter l'importance de chacun des mots du vocabulaire par $\mathrm{E}[\boldsymbol{\beta}_{\mathrm{k}}]$ selon la distribution de probabilité marginale $\mathbb{P}(\boldsymbol{\beta}_k|\boldsymbol{w}_{1:M}, \boldsymbol{\alpha}, \eta)$.[6]

Il est cependant impossible de calculer exactement la distribution *a posteriori*

$$\mathbb{P}(\boldsymbol{\theta}_{1:M}, \boldsymbol{z}_{1:M}, \boldsymbol{\beta}_{1:K}|\boldsymbol{w}_{1:M}, \boldsymbol{\alpha}, \eta) = \frac{\mathbb{P}(\boldsymbol{\theta}_{1:M}, \boldsymbol{z}_{1:M}, \boldsymbol{\beta}_{1:K}, \boldsymbol{w}_{1:M}|\boldsymbol{\alpha}, \eta)}{\mathbb{P}(\boldsymbol{w}_{1:M}|\boldsymbol{\alpha}, \eta)} \tag{2.10}$$

En effet, le calcul du facteur de normalisation $\mathbb{P}(\boldsymbol{w}_{1:M}|\boldsymbol{\alpha}, \eta)$ de la distribution *a posteriori* est donné par l'équation (2.11), qui est intraitable (*intractable*) [7].

$$\mathbb{P}(\boldsymbol{w}_{1:M}|\boldsymbol{\alpha}, \eta) = \prod_{d=1}^{M} \int_{\boldsymbol{\theta}_d} \int_{\boldsymbol{\beta}_1} \dots \int_{\boldsymbol{\beta}_K} \sum_{\boldsymbol{z} \in \mathcal{Z}^{N_d}} \mathbb{P}(\boldsymbol{\theta}_d, \boldsymbol{z}_d, \boldsymbol{\beta}_{1:K}, \boldsymbol{w}_d|\boldsymbol{\alpha}, \eta) d\boldsymbol{\beta}_K \dots d\boldsymbol{\beta}_1 d\boldsymbol{\theta}_{\boldsymbol{d}} \tag{2.11}$$

Blei présente, dans le papier original de LDA [7], une méthode d'inférence variationnelle (*varionational inference*) permettant d'approximer, en simplifiant le modèle de LDA, la distribution *a posteriori* des variables cachées $\mathbb{P}(\boldsymbol{\theta}_{1:M}, \boldsymbol{z}_{1:M}, \boldsymbol{\beta}_{1:K}|\boldsymbol{w}_{1:M}, \boldsymbol{\alpha}, \eta)$. Différentes techniques d'inférence ont également été développées par d'autres chercheurs, par exemple Teh et al. [27] et Griffiths et al. [14]. La technique d'inférence utilisée dans ce travail est celle proposée par Griffiths et al. [14] et sera décrite dans la section 2.3.5.

La section 2.3.8 présentera les techniques d'optimisation utilisées pour déterminer les valeurs de $\boldsymbol{\alpha}$ et η maximisant $\mathcal{L}(\boldsymbol{\alpha}, \eta) = log(\mathbb{P}(\boldsymbol{w_{1:M}}|\boldsymbol{\alpha}, \eta))$.

2.3.5 Inférence : *Collapsed Gibbs Sampling*

Dans le travail présenté dans ce document, une technique d'inférence différente de celle présentée par Blei [7] est utilisée. Il s'agit de la technique de *Collapsed Gibbs Sampling* (CGS) présentée par Griffiths et al. [14]. Cette section présentera le fonctionnement de cette technique.

Cette approche ne s'intéressera pas directement au problème d'approximation de la distribution *a posteriori* de toutes les variables cachées du modèle $(\boldsymbol{\theta}_{1:M}, \boldsymbol{z}_{1:M}, \boldsymbol{\beta}_{1:K})$ comme le fait la technique d'inférence variationnelle présentée par Blei [7]. En effet, l'idée générale de cette technique est plutôt de tirer un échantillon des variables $\boldsymbol{z}_{1:M}$ selon la loi de probabilité $\mathbb{P}(\boldsymbol{z}_{1:M}|\boldsymbol{w}_{1:M}, \boldsymbol{\alpha}, \eta)$ pour ensuite représenter la mixture de *topics* d'un document d par un estimateur ponctuel de $\boldsymbol{\theta}_d$ selon la distribution de probabilité $\mathbb{P}(\boldsymbol{\theta}_d|\boldsymbol{z}_{1:M}, \boldsymbol{\alpha}, \eta)$ et représenter l'importance de chacun des mots d'un *topic* k par un estimateur ponctuel de $\boldsymbol{\beta}_k$ selon la distribution de probabilité $\mathbb{P}(\boldsymbol{\beta}_k|\boldsymbol{z}_{1:M}, \boldsymbol{w}_{1:M}, \boldsymbol{\alpha}, \eta)$.

Le problème de tirer un échantillon des variables $\boldsymbol{z}_{1:M}$ selon la loi de probabilité $\mathbb{P}(\boldsymbol{z}_{1:M}|\boldsymbol{w}_{1:M}, \boldsymbol{\alpha}, \eta)$ sera abordé dans la section 2.3.6 et les lois de probabilité $\mathbb{P}(\boldsymbol{\theta}_d|\boldsymbol{z}_{1:M}, \boldsymbol{\alpha}, \eta)$ et $\mathbb{P}(\boldsymbol{\beta}_k|\boldsymbol{z}_{1:M}, \boldsymbol{w}_{1:M}, \boldsymbol{\alpha}, \eta)$ seront ensuite décrites dans la section 2.3.7.

2.3.6 *Collapsed Gibbs Sampling* : Tirage des variables d'assignation de *topic*

La notation de cette section diffèrera quelque peu de celle utilisée précédemment dans la présentation du modèle génératif qu'est LDA. Le symbole L représentera ici le nombre total de mots contenus dans le corpus.

$$L = \sum_{d=1}^{M} N_d$$

Notons que L n'est pas égal à la taille du vocabulaire ($L \neq N$). De plus, les *topics* et les mots des M documents du corpus ne seront pas représentées par M vecteurs $\boldsymbol{z_1}, \boldsymbol{z_2} \dots \boldsymbol{z_M}$ et M vecteurs $\boldsymbol{w_1}, \boldsymbol{w_2} \dots \boldsymbol{w_M}$ comme précédemment. Deux vecteurs de

L composantes $\boldsymbol{z} = (z_1, z_2, \dots z_L)$ et $\boldsymbol{w} = (w_1, w_2, \dots w_L)$ seront plutôt utilisés pour représenter les L *topics* et les L mots de tous les documents du corpus. Avec cette nouvelle notation, d_i représentera le document du corpus qui est relatif aux éléments à la position i dans les vecteurs $\boldsymbol{w}$ et $\boldsymbol{z}$. Il est ainsi possible de voir les vecteurs $\boldsymbol{w}$ et $\boldsymbol{z}$ comme étant une concaténation des vecteurs $\boldsymbol{w_1}, \boldsymbol{w_2} \dots \boldsymbol{w}_M$ et $\boldsymbol{z_1}, \boldsymbol{z_2} \dots \boldsymbol{z}_M$ respectivement.

La distribution de probabilité des variables $\boldsymbol{z}$ est donnée par

$$\mathbb{P}(\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\alpha}, \eta) = \frac{\mathbb{P}(\boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\alpha}, \eta)}{\sum\limits_{\boldsymbol{z} \in \mathcal{Z}^L} \mathbb{P}(\boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\alpha}, \eta)}$$

Pour tirer un échantillon selon cette distribution de probabilité, il faudrait évaluer et garder en mémoire la probabilité des K^L valeurs que $\boldsymbol{z}$ peut prendre, ce qui est impossible dû à la grandeur des valeurs que peut prendre K^L. Il sera donc nécessaire d'utiliser une technique permettant de tirer un échantillon selon une distribution de probabilité qui est assez près de $\mathbb{P}(\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\alpha}, \eta)$.

La technique d'échantillonnage utilisée dans ce travail est une technique de Monte-Carlo par Chaînes de Markov (MCCM) nommée *Gibbs Sampling* (GS). L'idée générale derrière les techniques de MCCM est de simuler une chaîne de Markov dans un espace d'états où chaque état correspond à une réalisation possible des variables aléatoires considérées, dans notre cas l'espace d'états est $\mathcal{Z}^L$, de telle sorte que la distribution stationnaire de cette chaîne soit la distribution selon laquelle l'échantillon désiré doit être tiré, dans notre cas $\mathbb{P}(\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\alpha}, \eta)$. Typiquement, dans ces méthodes, un état initial $\boldsymbol{z}^{(0)}$ est fixé de manière aléatoire ou selon un certain critère défini par l'utilisateur et par la suite, un processus itératif est amorcé et se poursuit jusqu'à ce qu'un certain critère d'arrêt soit atteint. Chaque itération de ce processus correspondra à une transition d'un état à un autre dans la chaîne de Markov simuler. Ce qui caractérise les différentes méthodes de MCCM est la façon de déterminer l'état qui sera visité à la prochaine itération selon l'état actuel. Pour la méthode de GS [4], en utilisant la

4. La méthode de GS utilisée dans ce travail est référée comme *Systematic Scan Gibbs Sampler* (SSGS), il existe cependant d'autres techniques, par exemple la technique de *Random Scan Gibbs*

notation du problème considéré dans ce travail, les coordonnées de l'état $\boldsymbol{z}^{(i)}$ visité à l'itération i sont choisies de manière séquentielle (pour $l = 1, 2, \dots L$) selon les lois de probabilité

$$\mathbb{P}(z_l^{(i)} | z_1^{(i)}, z_2^{(i)}, \dots z_{l-1}^{(i)}, z_{l+1}^{(i-1)}, \dots, z_L^{(i-1)}, \boldsymbol{w}, \boldsymbol{\alpha}, \eta)$$

Il est intéressant de remarquer ici que chaque itération s'effectue en un nombre d'opérations qui est de l'ordre de $K \cdot L$. Il peut être démontré, voir l'annexe B, que

$$\mathbb{P}(z_l | \boldsymbol{z}_{-l}, \boldsymbol{w}, \boldsymbol{\alpha}, \eta) \propto \left(F_{d,z_l}^{(-l)} + \alpha_{z_l}\right) \frac{C_{z_l,w_l}^{(-l)} + \eta}{C_{z_l}^{(-l)} + N\eta} \tag{2.12}$$

où

- $\boldsymbol{z}_{-l} = (z_1, z_2, \dots z_{l-1}, z_{l+1}, \dots, z_L)$
- d représente le document associé à la variable z_l
- $F^{(-l)}$ est une matrice $M \times K$ où l'entrée à la position (d, k) contient le nombre de fois que le *topic* k a été associé à un mot du document d et ce selon le vecteur $\boldsymbol{z}_{-l}$.
- $C^{(-l)}$ est une matrice $K \times N$ où l'entrée à la position (k, v) contient le nombre de fois que le mot v a été associé avec le *topic* k et ce selon le vecteur $\boldsymbol{z}_{-l}$.
- $C_k^{(-l)} = \sum_{v=1}^{N} C_{k,v}^{(-l)}$

2.3.7 Loi *a posteriori* des variables $\boldsymbol{\beta}_{1:K}$ et $\boldsymbol{\theta}_{1:M}$

Les variables d'intérêt pour la représentation du corpus selon les *topics* et pour la représentation des *topics* selon les mots du vocabulaire sont les variables cachées $\boldsymbol{\theta}_{1:M}$ et $\boldsymbol{\beta}_{1:K}$. Étant donné le vecteur d'assignation de *topics* $\boldsymbol{z}$ obtenu pour le processus de GS décrit dans la section 2.3.5, il peut être montré, voir annexe B.2, que

$$\boldsymbol{\theta}_d | \boldsymbol{z}_d, \boldsymbol{\alpha} \sim Dir(F_{d,1} + \alpha_1, F_{d,2} + \alpha_2, \dots F_{d,K} + \alpha_K)$$

et

$$\boldsymbol{\beta}_k | \boldsymbol{z}_{1:M}, \boldsymbol{w}_{1:M}, \eta \sim Dir(C_{k,1} + \eta, C_{k,2} + \eta, \dots C_{k,N} + \eta)$$

Sampler (RSGS)

où F et C sont les mêmes matrices que celles définies dans l'annexe B pour le calcul de $\mathbb{P}(\boldsymbol{z}|\boldsymbol{\alpha},\eta)$ et de $\mathbb{P}(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\alpha},\eta)$ respectivement.

Ainsi,

$$\theta_{d,i}|\boldsymbol{z}_d,\boldsymbol{\alpha} \sim Beta(F_{d,i}+\alpha_i, \left(\sum_{k=1}^{K} F_{d,k}+\alpha_k\right) - (F_{d,i}+\alpha_i))$$

et

$$\beta_{k,i}|\boldsymbol{z}_{1:M},\boldsymbol{w}_{1:M},\eta \sim Beta(C_{k,i}+\eta, (N-1)\eta + \left(\sum_{v=1}^{N} C_{k,v}\right) - C_{k,i})$$

les espérances de ces distributions de probabilité seront choisies comme estimateurs ponctuels $\hat{\beta}_{k,v}$ et $\hat{\theta}_{d,k}$, ainsi

$$\hat{\beta}_{k,v} = \frac{C_{k,v}+\eta}{N\eta + \sum_{i=1}^{N} C_{k,i}} \tag{2.13}$$

et

$$\hat{\theta}_{d,k} = \frac{F_{d,k}+\alpha_k}{N_d+\alpha_i} \tag{2.14}$$

Ce choix aura pour effet de minimiser les espérances des erreurs quadratiques $\mathrm{E}\left[(\hat{\beta}_{\mathrm{k,v}}-\beta_{\mathrm{k,v}})^2\right]$ et $\mathrm{E}\left[(\hat{\theta}_{\mathrm{d,k}}-\theta_{\mathrm{d,k}})^2\right]$ [17].

2.3.8 Optimisation des paramètres

Étant donné les variables $\boldsymbol{z}_{1:M}$ obtenues par la technique de CGS décrite plutôt, de nouvelles valeurs des paramètres du modèle, $\boldsymbol{\alpha}$ et η, peuvent être calculées afin d'optimiser la quantité $\mathcal{L}(\boldsymbol{\alpha},\eta) = \mathbb{P}(\boldsymbol{w}_{1:M},\boldsymbol{z}_{1:M}|\boldsymbol{\alpha},\eta)$.

Selon la règle de Bayes,

$$\mathbb{P}(\boldsymbol{w}_{1:M},\boldsymbol{z}_{1:M}|\boldsymbol{\alpha},\eta) = \mathbb{P}(\boldsymbol{w}_{1:M}|\boldsymbol{z}_{1:M},\boldsymbol{\alpha},\eta)\mathbb{P}(\boldsymbol{z}_{1:M}|\boldsymbol{\alpha},\eta)$$

De plus, selon le modèle spécifié par LDA

$$\mathbb{P}(\boldsymbol{w}_{1:M}|\boldsymbol{z}_{1:M},\boldsymbol{\alpha},\eta)\mathbb{P}(\boldsymbol{z}_{1:M}|\boldsymbol{\alpha},\eta) = \mathbb{P}(\boldsymbol{w}_{1:M}|\boldsymbol{z}_{1:M},\eta)\mathbb{P}(\boldsymbol{z}_{1:M}|\boldsymbol{\alpha})$$

Et donc,

$$\mathcal{L}(\boldsymbol{\alpha}, \eta) = \mathbb{P}(\boldsymbol{w}_{1:M}|\boldsymbol{z}_{1:M}, \eta)\mathbb{P}(\boldsymbol{z}_{1:M}|\boldsymbol{\alpha})$$

Comme $\mathbb{P}(\boldsymbol{w}_{1:M}|\boldsymbol{z}_{1:M}, \eta)$ ne dépend pas de $\boldsymbol{\alpha}$ et que $\mathbb{P}(\boldsymbol{z}_{1:M}|\boldsymbol{\alpha})$ ne dépend pas de η, trouver les valeurs de $\boldsymbol{\alpha}$ et η qui maximisent $\mathcal{L}(\boldsymbol{\alpha}, \eta)$ revient à trouver la valeur de $\boldsymbol{\alpha}$ qui maximise $\mathbb{P}(\boldsymbol{z}_{1:M}|\boldsymbol{\alpha})$ et la valeur de η qui maximise $\mathbb{P}(\boldsymbol{w}_{1:M}|\boldsymbol{z}_{1:M}, \eta)$.

Comme démontré dans l'annexe B,

$$\mathbb{P}(\boldsymbol{z}_{1:M}|\boldsymbol{\alpha}) = \prod_{d=1}^{M} \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \frac{\prod_{k=1}^{K} \Gamma\left(F_{d,k} + \alpha_k\right)}{\Gamma(N_d + \sum_{k=1}^{K} \alpha_k))} \tag{2.15}$$

et

$$\mathbb{P}(\boldsymbol{w}_{1:M}|\boldsymbol{z}, \eta) = \prod_{k=1}^{K} \frac{\Gamma\left(N\eta\right)}{\Gamma(\eta)^N} \frac{\prod_{n=1}^{N} \Gamma\left(C_{k,n} + \eta\right)}{\Gamma(C_k + N\eta))} \tag{2.16}$$

où F et C sont les mêmes matrices que celles définies dans l'annexe B pour le calcul de $\mathbb{P}(\boldsymbol{z}|\boldsymbol{\alpha}, \eta)$ et de $\mathbb{P}(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\alpha}, \eta)$ respectivement.

Wallach présente dans sa thèse [28, section 2.3] plusieurs méthodes pour optimiser ce genre de fonction. La méthode utilisée dans les expériences effectuées dans ce travail est la première méthode à point fixe présentée à la section 2.3.5 de la thèse de Wallach.

2.3.9 *Topic-In-Set Knowledge Latent Dirichlet Allocation*

David Andrzejewski propose une technique, nommée TISK-LDA [2], pour ajouter des connaissances de l'utilisateur sur la constitution des *topics* à l'algorithme LDA. Le but de cette technique est de permettre à l'utilisateur de spécifier, pour chaque variable z_l, un ensemble de *topics* $\mathcal{C}_l = \{c_1, c_2, \ldots c_{P_l}\}$ où $c_i \in \mathcal{Z}$ qu'il jugerait plus adéquat que les autres pour cette position l donnée.

De façon concrète, ces ensembles de *topics* $\mathcal{C}_l$ seront utilisés pour modifier la procédure de CGS de l'algorithme LDA. Durant la procédure standard de CGS de LDA, le tirage aléatoire de la valeur que prendra la variable z_l est fait selon une distribution de probabilité

$$\mathbb{P}(z_l|\boldsymbol{z}_{-l}, \boldsymbol{w}, \boldsymbol{\alpha}, \eta) = \frac{\left(F_{d,z_l}^{(-l)} + \alpha_{z_l}\right) \frac{C_{z_l,w_l}^{(-l)} + \eta}{C_{z_l}^{(-l)} + N\eta}}{\sum_{k=1}^{K} \left(F_{d,k}^{(-l)} + \alpha_k\right) \frac{C_{k,w_l}^{(-l)} + \eta}{C_k^{(-l)} + N\eta}} \tag{2.17}$$

Ce que Andrzejewski propose dans [2] est de modifier cette quantité de la façon suivante

$$\mathbb{P}(z_l|\boldsymbol{z}_{-l}, \boldsymbol{w}, \boldsymbol{\alpha}, \eta) = \frac{(\rho \cdot \delta(z_l \in \mathcal{C}_l) + 1 - \rho)\left[\left(F_{d,z_l}^{(-l)} + \alpha_{z_l}\right) \frac{C_{z_l,w_l}^{(-l)} + \eta}{C_{z_l}^{(-l)} + N\eta}\right]}{\sum_{k=1}^{K} (\rho \cdot \delta(k \in \mathcal{C}_l) + 1 - \rho)\left[\left(F_{d,k}^{(-l)} + \alpha_k\right) \frac{C_{k,w_l}^{(-l)} + \eta}{C_k^{(-l)} + N\eta}\right]} \tag{2.18}$$

où ρ est une valeur comprise entre 0 et 1 fixée par l'utilisateur qui spécifie à quel point les contraintes $\mathcal{C}_l$ sont «importantes» et $\delta(k \in \mathcal{C})$ prend la valeur 1 si $k \in \mathcal{C}$ et 0 sinon.

La technique qui a été utilisée dans ce travail pour construire les ensembles $\mathcal{C}_l$ est maintenant décrite. L'algorithme prend en entrée une série de J groupes, $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_J$ de mots, avec $0 < J < K$, définis par l'utilisateur. Chacun des J groupe constitue un ensemble de mots qui sont, selon les critères de l'utilisateur, reliés à un domaine d'intérêt. Un tel groupe de mots pourrait être par exemple {music, album, song, singer, pop, rap}. Chaque $\mathcal{C}_l$ sera alors construit de la manière suivante :

$$\mathcal{C}_l \quad = \quad \begin{cases} \{1, 2, \ldots, K\} & \text{si } \nexists j | w_l \in \mathcal{M}_j \\ \{j | w_l \in \mathcal{M}_j\} & \text{sinon} \end{cases}$$

2.4 Classification non-supervisée des requêtes basée sur les pages web consultées

Cette section présente la technique qui est proposée pour effectuer la classification non-supervisée d'un ensemble de requêtes de recherche web en se basant sur le texte contenu dans la ou les pages web qui ont été visitées par les émetteurs de ces requêtes. En utilisant ces textes de page web pour effectuer le partitionnement des requêtes, on espère obtenir un partitionnement de meilleure qualité qu'un partitionnement qui serait obtenu par une méthode qui ne considèrerait que le texte formant les requêtes.

Comme mentionnée dans la section 1.4.1, l'idée générale de cette technique peut se séparer en trois étapes. Tout d'abord il faut aller chercher le texte contenu dans chaque page web présente dans l'ensemble de données étudié de façon à créer un corpus de document et appliquer quelques techniques de prétraitement aux documents de ce corpus. Les détails de cette étape seront présentés à la section 2.4.1. Une technique de classification non-supervisée sera ensuite utilisée pour créer une partition de ce corpus en K groupes disjoints. Les détails de cette étape seront quant à eux présentés dans la section 2.4.2. Finalement, ce partitionnement des pages web sera utilisé pour créer une partition des requêtes reliées à ces pages web en K groupes disjoints. La section 2.4.3 détaillera cette dernière étape.

2.4.1 Acquisition et prétraitement du texte des pages web

Extraction du texte des pages web : Comme cette information n'est pas directement contenue dans les URLs présents dans l'ensemble de données, il est nécessaire d'aller extraire du web le code source de chaque URL de l'ensemble considéré. Une fois l'acquisition de ces codes sources accomplie, une technique devra être utilisée pour effectuer l'extraction du texte contenu dans chacun d'eux. Plusieurs librairies sont disponibles pour effectuer ce genre de tâche, dans le présent travail, les librairies BeautifulSoup[5] et Boilerpipe[6] ont été utilisées. Il faut souligner qu'il est préférable

5. http://www.crummy.com/software/BeautifulSoup/
6. https://code.google.com/p/boilerpipe/

d'effectuer cette extraction de texte le plus près possible de la date d'émission des requêtes considérées puisque le contenu de certaines pages web peut évoluer avec le temps et certaines d'entre elles peuvent également disparaître après une certaine période.

Prétraitement : Le processus de prétraitement suggéré dans ce travail consiste à appliquer pour chaque document représentant le texte d'une page web les étapes suivantes :

1. la normalisation du texte :
   - séparation du texte en *token* à l'aide du module python nltk [7] contenant la classe TreebankWordTokenizer [8] ;
   - séparation des *tokens* contenant les caractères «/» ou «.» en plus d'un *token* (pas effectué par TreebankWordTokenizer) ;
2. l'application d'une technique de *stemming* sur le texte (algorithme : *The English (Porter2) stemming algorithm* [9]) ;
3. le retranchement des mots contenus dans une liste de *stopwords* ;
4. le retranchement des mots n'apparaissant que dans un seul document.

2.4.2 Partitionnement des pages web

Cette section présentera la technique proposée pour effectuer le partitionnement des pages web associées aux requêtes étudiées. Cette technique est majoritairement basée sur l'algorithme LDA. Dans sa version la plus simple, la technique proposée est complètement non-supervisée. Cette version sera présentée dans les premiers paragraphes de cette section. Il sera ensuite question d'une légère «modification», nécessitant l'intervention d'un humain, pouvant être apportée à cette technique.

7. http://nltk.org/
8. http://nltk.googlecode.com/svn/trunk/doc/api/nltk.tokenize.treebank.TreebankWordTokenizer-class.html
9. http://snowball.tartarus.org/algorithms/english/stemmer.html

Version complètement non-supervisée

Étant donné le corpus des textes de pages web acquis dans l'étape d'extraction de texte décrite dans la section précédente, la première étape de ce processus de classification de pages web sera d'appliquer l'algorithme LDA sur ce corpus de textes. Cet algorithme donnera en sortie :

- Un ensemble de K *topics*, chacun représenté par une distribution de probabilité sur les mots du vocabulaire engendré par le corpus étudié. Pour un *topic* k donné, la probabilité du mot v est donnée par la valeur de $\hat{\beta}_{k,v}$ définie par l'équation 2.13.
- Pour chaque page web du corpus, un vecteur de K composantes donnant la proportion ou l'importance, pour cette page web, de chacun des K *topics* générés par l'algorithme. La somme des K composantes de ce vecteur sommant à $1,0$. Pour une page web d donnée, la proportion du *topic* k est donnée par la valeur de $\hat{\theta}_{d,k}$ définie par l'équation (2.14).

Une fois l'algorithme LDA appliqué, la technique de partitionnement de l'ensemble des pages web (URLs) est simple. Il s'agit de créer K groupes distincts qui sont chacun associés à l'un des K *topics* générés par LDA. Une page web (URL) se retrouvera alors dans le groupe associé au *topic* ayant la plus grande proportion ou importance selon la représentation donnée par LDA. Autrement dit, une page web d se retrouvera dans le *cluster* associé au *topic* k, où $k = \underset{k \in \{1,2,\ldots,K\}}{\arg\max}\ \hat{\theta}_{d,k}$. Il est ainsi possible de créer un partitionnement des pages web comportant K groupes distincts qui pourra être utilisé dans l'étape suivante du processus de classification des requêtes. Étape qui sera présentée à la section 2.4.3.

Modification : *Topic-In-Set Knowledge Latent Dirichlet Allocation*

Cette modification propose l'utilisation d'une version modifiée de LDA, nommée TISK-LDA, pour faire le *topic modeling*. Cette version permet à l'utilisateur de soumettre, pour un certain nombre de *topics*, de petits groupes de mots, chacun relié à un sujet qu'il voudrait voir apparaître comme étant un *topic* en sortie de LDA. Bien que cette modification demande une certaine supervision (les groupes de mots fournis

en entrée), la possibilité de pouvoir «diriger» les sujets de certains *topics* peut être utile dans le cas où les *topics* obtenus par LDA sont trop généraux ou trop spécifiques. Un exemple de cette situation sera contenu dans l'analyse des résultats présentée à la section 3.3.2. Cette technique peut également être utilisée pour s'assurer que certains sujets traités dans les textes (pages web) apparaîtront bien comme étant un *topic*.

2.4.3 Partitionnement des requêtes

Cette section présentera l'étape finale de la technique de classification des requêtes de recherche web proposée dans ce travail. Cette étape consiste uniquement à utiliser le partitionnement des pages web obtenu à l'étape précédente pour établir un partitionnement des requêtes qui leur sont associées. Dans cette section, le terme URL sera utilisé comme synonyme de page web.

Considérons la notation :

- $\mathcal{L}$: un ensemble de K étiquettes distinctes
- $\mathcal{U}$: l'ensemble des URLs contenus dans le jeu de données étudié
- $\mathcal{R}$: l'ensemble des requêtes contenues dans le jeu de données étudié
- $url(\cdot)$: une fonction allant de $\mathcal{R} \longrightarrow \mathcal{P}(\mathcal{U})$ donnant pour une requête r donnée l'ensemble des URLs qui lui sont associés [10]
- $P_U : \mathcal{U} \longrightarrow \mathcal{L}$ une fonction décrivant un partitionnement de l'ensemble $\mathcal{U}$

Avec cette notation, étant donné un partitionnement des URLs P_U, le partitionnement de l'ensemble des requêtes $\mathcal{R}$ sera donné par la fonction (2.19).

$$\begin{aligned} P_R : &\mathcal{R} \longrightarrow \mathcal{L} \\ &r \longmapsto \arg\max_{l \in \mathcal{L}} |\{u \in url(r) | P_U(u) = l\}| \end{aligned} \tag{2.19}$$

Dans le cas où plus d'un argument l mène à un maximum de $|\{u \in url(r) | P_U(u) = l\}|$, en notant $\mathcal{L}^M$ l'ensemble des l menant à ce maximum et en supposant qu'il est possible d'attribuer à chaque URL un certain degré d'appartenance à son *cluster* noté $score(u, l)$, on choisira, parmi les $l \in \mathcal{L}^M$, celui qui est tel que $\max_{u \in P_U^{-1}(l)} score(u, l)$ est

10. $\mathcal{P}(\mathcal{U})$ représente l'ensemble des parties de $\mathcal{U}$

maximale. Dans la technique ici proposée, ce score entre un URL et son *cluster* sera donné par l'importance du *topic* associé à ce *cluster* pour la page web représenté par le URL. Autrement dit, si le *cluster* est associé au *topic* k et que le URL correspond à la page web d, alors le score entre le URL et son *cluster* sera donné par $\hat{\theta}_{d,k}$.

2.5 Mesures d'évaluation utilisées

Le chapitre 3 présentera, entre autres, une analyse des résultats obtenus par certaines méthodes de classification non-supervisée appliquées sur l'ensemble de requêtes qui est étudié dans ce document. Ces analyses feront référence à certaines mesures d'évaluation servant à évaluer les *clusters* de requêtes obtenus. Les définitions des mesures d'évaluation qui seront utilisées sont présentées dans cette section.

Dans le contexte de classification non-supervisée, l'évaluation des résultats suppose qu'un certain étiquetage d'une partie ou de la totalité de l'ensemble de données considéré soit disponible et représente la classification qui devrait idéalement être obtenue par le processus de classification non-supervisée à évaluer. Pour les définitions qui suivent, la notation suivante sera utilisée :

- L : le nombre total d'étiquettes distinctes présentes dans les données étiquetées.
- K : le nombre de *clusters* qui ont été créés par le processus de classification non-supervisée à évaluer.
- n_{ij} : le nombre de données portant l'étiquette i contenues dans le *cluster* j.
- $n_{i\cdot}$: le nombre total de données portant l'étiquette i.
- $n_{\cdot j}$: le nombre total de données contenues dans le *cluster* j.
- n : le nombre total de données étiquetées.

Tout d'abord, il sera question de mesure pour évaluer la qualité de chaque *cluster* d'un ensemble de manière individuelle. Pour un *cluster* j donné, il est possible de calculer une valeur de précision et de rappel pour chacune des L étiquettes possibles. La **précision** de l'étiquette i pour le *cluster* j donnera la proportion des données portant l'étiquette i parmi toutes les données étiquetées contenues dans le *cluster* j.

Autrement dit,

$$précision(i,j) = \frac{n_{ij}}{n_{.j}}$$

Le **rappel** de l'étiquette i pour le *cluster* j donnera quant à lui la proportion des données portant l'étiquette i qui se retrouvent dans le *cluster* j. Autrement dit,

$$rappel(i,j) = \frac{n_{ij}}{n_{i.}}$$

Le terme **pureté** sera également utilisé dans ce document pour qualifier la qualité d'un *cluster*. En attribuant le terme **étiquette majoritaire** du *cluster* j à l'étiquette c qui est telle que $c = \underset{i=1,2,...,L}{\arg\max}\, n_{ij}$, la pureté du *cluster* j n'est rien d'autre que la précision qui est rattachée à l'étiquette majoritaire du *cluster* j. Autrement dit, la **pureté** d'un *cluster* j est donnée par

$$pureté(j) = \max_{i=1,2,...,L} \frac{n_{ij}}{n_{.j}}$$

Dans le but d'évaluer la qualité d'un ensemble de *clusters*, la notion de **pureté globale** sera utilisée. La pureté globale d'un ensemble de K *clusters* est définie comme étant la moyenne pondérée de la pureté de chacun des K *clusters*, où les poids utilisés sont proportionnels à la taille des *clusters*. Autrement dit, la pureté globale d'un ensemble de K *clusters* est donnée par $\sum_{j=1}^{K} \frac{n_{.j}}{n} * pureté(j)$.

2.6 Détection d'entités nommées

Cette section présentera deux méthodes non-supervisées pour détecter la présence de certaines entités nommées dans les requêtes étudiées en se servant du texte des pages web visitées. La première méthode présentée découvrira peu d'entités nommées, mais le fera avec une grande précision et fournira également le type des entités nommées découvertes. La deuxième méthode en découvre un plus grand nombre, mais le fait avec une moins grande précision. La section 2.6.2 décrira la première méthode et la section 2.6.3 décrira la seconde, mais tout d'abord, quelques étapes de prétraitement qui sont nécessaires aux deux méthodes seront présentées à la section 2.6.1.

2.6.1 Prétraitement

Normalisation

Chaque morceau de texte étudié, que ce soit une requête ou le texte d'une page web, sera normalisé de la façon suivante :

- séparation du texte en *token* à l'aide du module python nltk [11] contenant la classe TreebankWordTokenizer [12]
- séparation des *tokens* contenant les caractères «/» ou «.» en plus d'un *token* (pas effectué par TreebankWordTokenizer)
- transformation «digit to word» à l'aide du module python num2words (1 –> 'one', 22 –>'twenty two')
- transformation «&» –> «and»

Génération d'Entités Nommées Potentielles

Pour les deux méthodes présentées, la première étape du procédé est de générer des Entités Nommées Potentielles (ENPs) pour chaque requête étudiée. Ces ENPs seront ensuite évaluées par l'une ou l'autre des deux approches. Pour une requête donnée, les candidats générés sont tous les n-grams contenus dans la requête à l'exception de ceux qui ne sont formés que de mots qui sont contenus dans une *stopword list* donnée. À titre d'exemple, le Tableau 2.1 présente les ENPs qui seraient générés à partir de la requête «review of millard fillmore 's queens new york».

2.6.2 Méthode 1 : Basée sur le format *microdata* et Schema.org

L'idée principale pour cette méthode est de tirer avantage de l'information supplémentaire qui est donnée par les balises *HyperText Markup Language* (HTML) augmentées d'attributs provenant de la spécification de type *microdata* [13] contenues dans le code source de la page web visitée. Pour des fins de clarté, avant de continuer

11. http://nltk.org/
12. http://nltk.googlecode.com/svn/trunk/doc/api/nltk.tokenize.treebank.TreebankWordTokenizer-class.html
13. http://www.w3.org/TR/2013/NOTE-microdata-20131029/

Requête	**review of millard fillmore's queens new york**
Unigrams	review ; millard ; fillmore ; queens ; new ; york
Bigrams	review of ; of millard ; millard fillmore ...
Trigrams	review of millard ; of millard fillmore ...
...	...
7-grams	review of millard fillmore 's queens new ; of millard fillmore 's queens new york
8-gram	review of millard fillmore 's queens new york

Tableau 2.1 – Exemple de génération des Entités Nommées Potentielles

plus loin dans l'explication de la technique ici proposée, une brève explication de la spécification *microdata* sera d'abord présentée.

Spécification *microdata*

Le terme *microdata* réfère à une spécification HTML [14] qui a été créée pour permettre aux concepteurs de pages web d'ajouter de l'information supplémentaire, pouvant être facilement collectée par des robots d'indexation (*web crawlers*), sur certains contenus de leurs pages web. Plus précisément, cette spécification permet, par l'ajout des attributs *itemscope*, *itemtype*, *itemprop*, *itemid* et *itemref* dans certaines balises HTML, d'identifier certains items présents dans une page web et de leur attribuer un type et certaines propriétés. À titre d'exemple, la spécification *microdata* pourrait être utilisée pour annoter le segment de code HTML suivant :

```
<div>
Primus est un groupe de rock qui a vu le jour en 1984.
Pork Soda est l'un de leurs meilleurs albums.
</div>
```

de la façon suivante :

```
<div itemscope itemtype=http://example.org/MusicGroup>
<span itemprop="name">Primus</span> est un groupe
```

14. http://www.w3.org/TR/2013/NOTE-microdata-20131029/

```
de <span itemprop="genre">rock</span> qui a vu le
jour en <span itemprop="foundingDate">1984</span>.
<span itemprop="album">Pork Soda</span> est l'un de
leurs meilleurs albums.
</div>
```

Ce segment de code, annoté à l'aide de la spécification *microdata*, permet l'identification de la présence d'un item de type *http ://example.org/MusicGroup* ayant les propriétés suivantes :

- *name* : «Primus»
- *genre* : «rock»
- *foundingDate* : «1984»
- *album* : «Pork Soda»

Il est important de noter que la spécification *microdata* ne spécifie pas une taxonomie particulière en ce qui concerne le type des items (valeur de l'attribut *itemtype*) et donc un créateur de pages web est libre d'inscrire ce qu'il désire comme valeur de cet attribut (la spécification demande uniquement qu'il s'agisse d'un URL). Ceci apporte l'avantage de laisser une grande liberté aux créateurs de page web quant au choix du type qu'ils veulent utiliser, mais apporte également le désavantage de créer une potentielle inconsistance entre les pages web crées par différentes personnes ou organisations. Ainsi, afin de rendre l'utilisation de cette spécification plus uniforme, le site Schema.org propose une taxonomie (ou vocabulaire) pour identifier le type des items présents dans une page web. De plus, pour chaque type défini dans la taxonomie proposée, Schema.org définit une série d'attributs spécifiques à la nature de ce type. En particulier, chaque type possède un attribut nommé *name* identifiant le nom de l'item dont il est question.

Méthode proposée

La méthode proposée dans cette section est basée sur le contenu des pages web considérées qui est annoté selon la spécification *microdata*. Plus précisément, cette méthode ne considèrera que la valeur de l'attribut *name* des items dont le type fait

partie de la taxonomie proposée par Schema.org. Pour le restant de cette section, le terme Nom d'Item MicroData (NIM) fera référence à cette valeur.

Cette méthode évaluera chaque ENP d'une requête en calculant une certaine distance entre la ENP et chacun des NIMs contenus dans la page associée à la requête. Il est important de noter que puisque les NIMs considérés sont associés à un type, l'utilisateur peut restreindre les NIMs considérés à ceux étant associés à un type pour lequel il possède un intérêt.

Comme mentionnée précédemment, pour chaque ENP d'une requête et chaque NIM issu de la page web associée à cette requête, une distance devra être calculée. La distance utilisée ici sera la distance de Levenshtein normalisée. Pour une paire (ENP, NIM) donnée, cette distance est définie comme :

$$d(\text{ENP}, \text{NIM}) = \frac{dist_{Lev}(\text{ENP}, \text{NIM})}{max\left[len(\text{ENP}), len(\text{NIM})\right]} \tag{2.20}$$

Ainsi, une ENP sera considérée comme étant une réelle entité nommée si sa distance avec l'un des NIMs est plus petite qu'un certain seuil $\delta \in [0, 1]$ fixé par l'utilisateur.

Pour les expérimentations effectuées dans ce travail, les coûts d'insertion, de retranchement et de substitution seront tous fixés à $1, 0$. Une version plus raffinée pourrait utiliser une matrice de coût spécifiant le coût de substitution pour chaque paire possible de lettres.

Note : Comme les requêtes étudiées dans ce travail ont été transmises par l'utilisateur au moyen de sa voix pour ensuite être transcrites en texte par un système de reconnaissance vocale, une transformation «grapheme-to-phoneme» (G2P) sera appliquée à la fois aux ENPs et aux NIMs considérés avant de calculer cette distance. Dans ce cas, chaque phonème sera considéré comme étant un symbole d'un alphabet et donc remplacera le rôle traditionnel d'une lettre dans la distance de Levenshtein présentée. La raison pour effectuer cette transformation est de permettre le jumelage de groupes de mots qui sont plus similaires sous une représentation phonétique qu'ils

le sont sous une représentation textuelle. Un exemple permettant de bien illustrer les bénéfices apportés par cette représentation en phonème suit.

En supposant qu'un usager ait soumis vocalement la requête «eye of the tiger» et que le système de reconnaissance vocale commet une erreur et traduit cette dernière comme étant «i of the tiger», alors si la page consultée par l'utilisateur contient les mots «eye of the tiger» identifiés par une balise *microdata*, la distance entre «i of the tiger» et «eye of the tiger» ne sera pas de 0, alors que les mots «i of the tiger» correspondent bel et bien aux mots «eye of the tiger». Cependant, en transformant les deux groupes de mots sous une représentation en phonème, les mots «i» et «eye» seront tout deux remplacés par le phonème «AY» et donc la distance entre «i of the tiger» et «eye of the tiger» sera de 0, tel que souhaité. De cette façon, la technique présentée permet d'identifier des entités nommées qui auraient pu être mal orthographiées par le système de reconnaissance vocale.

Note Pour une paire (ENP, NIM) donnée dont la distance entre l'ENP et le NIM est plus petite que le seuil spécifié et dont l'orthographe de l'ENP et du NIM diffère, en faisant l'hypothèse que le texte du NIM est correctement orthographié, la méthode décrite précédemment permet également de détecter des erreurs de frappe (dans le contexte de ce travail, de reconnaissance vocale) qui ont été commises par l'utilisateur (dans le contexte de ce travail, par le système de reconnaissance vocale).

2.6.3 Méthode 2 : Texte entier de la page web

Au lieu d'utiliser seulement les balises du type *microdata* pour effectuer l'évaluation des ENPs, comme discuté à la section 2.6.2, l'approche décrite dans cette section se servira du texte de la page web en entier. Dans cette section, pour une page web donnée, le terme Fragment de Texte Important (FTI) fera référence à tout n-gram apparaissant plus d'une fois dans le texte de la page web.

Association entre Entités Nommées Potentielles et Fragments de Texte Importants

Étant donné un ensemble de ENPs provenant d'une certaine requête et l'ensemble des FTIs, noté $\mathcal{T}$, provenant de la page web associée à la requête. La première étape de cette méthode sera de former des paires (ENP, FTI) pour chaque ENP provenant de la requête. On associera à un ENP donné, le FTI t qui est tel que $t = \arg\min_{t \in \mathcal{T}} [d(\text{ENP}, t)]$ où d est la fonction de distance 2.20 présentée précédemment.

Choisir entre deux Entités Nommées Potentielles se chevauchant

L'ensemble des ENPs provenant d'une requête contiendra généralement des ENPs qui se chevauchent (c.-à-d. partagent des mots en commun). Par exemple, étant donnée la requête «jennifer anniston movie», les ENPs `jennifer anniston` et `anniston movie` se chevauchent parce qu'ils partagent le mot «anniston». Lorsque ceci se produit, la méthode proposée se basera sur un critère de sélection pour trier l'ensemble des ENPs afin qu'il ne contienne finalement que des ENPs qui ne se chevauchent pas. Le critère de sélection qui a été utilisé dans ce travail sera défini dans le restant de cette section. Il faut noter que ce critère a été développé manuellement en testant plusieurs possibilités sur le jeu de données utilisé dans ce travail et qu'il serait intéressant, dans des travaux futurs, d'explorer d'autre façon, peut-être plus élégante, de définir ce critère.

Étant donné un ensemble de paires (ENP, FTI) où les ENPs se chevauchent et sont issues d'une même requête, la paire (ENP, FTI) fournissant le score le plus élevé pour une certaine fonction de score donnée sera gardée et les autres paires seront mises de côté. La fonction de score proposée ici est définie comme :

$$score(enp, fti) = sim(enp, fti) * imp(fti) * min(numWords(enp), numWords(fti))$$

La fonction sim donne la similarité entre la ENP et le FTI et est définie comme $sim(enp, fti) = 1.0 - d(enp, fti)$, où d est la fonction de distance (2.20). La fonction imp donne l'importance du FTI en se basant sur le texte de la page web selon la définition $imp(fti) = min(tf(fti), MAX\_FREQ)$, où $tf(fti)$ donne la fréquence

d'apparition du FTI dans le texte de la page web considérée et MAX_FREQ est une constante qui doit être fixée par l'utilisateur. La raison pour laquelle on prend le minimum entre $tf(fti)$ et MAX_FREQ et non simplement $tf(fti)$ est qu'on veut éviter de donner une importance trop grande aux FTIs très fréquents.

Finalement, le produit de la similarité et de l'importance est multiplié par le minimum entre le nombre de mots composant la ENP et le nombre de mots composant le FTI. Cette multiplication est effectuée pour donner un avantage aux paires contenant un plus grand nombre de mots. L'exemple suivant illustre l'utilité de cette multiplication. Considérons les paires (ENPs, FTIs) (John, John) et (John Trabolta, John Travolta). La paire (John, John) aura une similarité de 1.0, ce qui sera plus grand que la similarité de la paire (John Trabolta, John Travolta). Disons maintenant que John et John Travolta ont la même importance dans le texte de la page web considérée. Alors, si on ne multiplie pas la valeur du score par une certaine fonction du nombre de mots de la ENP et du FTI, la paire (John, John) aura un plus grand score que la paire (John Trabolta, John Travolta) et sera donc choisie comme paire restante. Ce qui, dans ce cas, serait une erreur.

Évaluation des Entités Nommées Potentielles

Étant données une requête et la page web qui lui est associée, les étapes décrites dans cette section produiront un ensemble de paires (ENP, FTI) où les différentes ENPs ne se chevauchent pas. À partir de ce moment, la façon d'identifier les réelles entités nommées se fera comme à la section 2.6.2. C'est-à-dire, qu'une ENP sera considérée comme une vraie entité nommée si la distance, selon la fonction (2.20), entre cette dernière et le FTI qui lui est associé est plus petite qu'un certain seuil fixé par l'utilisateur.

Note : Tout comme pour la première méthode décrite à la section 2.6.2, pour une paire (ENP, FTI) donnée où l'ENP a été identifiée comme une réelle entité nommée et où l'orthographe de l'ENP n'est pas la même que celle du FTI qui lui est associé, en faisant l'hypothèse que le texte de la page web est correctement orthographié, la méthode décrite précédemment permet également de détecter des erreurs de frappe

(dans le contexte de ce travail, de reconnaissance vocale) qui ont été commises par l'utilisateur (dans le contexte de ce travail, par le système de reconnaissance vocale).

Chapitre 3

Présentation et analyse des résultats

Ce chapitre présentera les résultats obtenus en appliquant les techniques proposées précédemment à l'ensemble de requêtes ici étudié, et ce pour les deux objectifs que sont la classification non-supervisée de requêtes de recherche web et la détection non-supervisée d'entités nommées à l'intérieur de requêtes de recherche web. Les sections 3.2 et 3.3 de ce chapitre sont relatives à la classification non-supervisée tandis que la section 3.4 est relative à la détection d'entités nommées.

Plus précisément, la section 3.1 présentera tout d'abord l'ensemble de données d'évaluation qui a été utilisé pour calculer certaines mesures servant à évaluer la qualité des résultats présentés dans ce chapitre. La section 3.2 présentera ensuite les résultats obtenus en appliquant les algorithmes de type K-Means présentés à la section 2.2 aux requêtes de l'ensemble étudié en ne considérant que les mots qui constituent ces dernières (c.-à-d. ignorant complètement les pages web qui leur sont associées). Ces résultats sont présentés pour appuyer le fait que de ne considérer que les mots composant les requêtes pour effectuer leur classification de façon non-supervisé ne mène pas à des résultats très satisfaisants. La section 3.3 présentera ensuite les résultats obtenus par la méthode de classification non-supervisée proposée. Finalement, la section 3.4 présentera les résultats obtenus en appliquant les deux techniques propo-

sées à la section 2.6 pour effectuer la détection non-supervisée d'entités nommées au sein de requêtes de recherche web.

Note : Certains tableaux contenus dans cette section présenteront des requêtes issues de l'ensemble étudié. Lorsque des noms d'individus qui ne sont pas du domaine public apparaitront dans ces derniers, ils seront remplacés par une répétition du caractère «*».

3.1 Données d'évaluation

Afin de pouvoir calculer des mesures d'évaluation supervisées sur les résultats obtenus par les diverses techniques de *clustering* et de détection d'entités nommées présentées, un sous-ensemble des requêtes étudiées a été manuellement annoté. Ce sous-ensemble de requêtes est constitué de 3 000 requêtes qui ont été choisies de manière aléatoire parmi les 29 618 requêtes étudiées. Le processus d'annotation qui sera appliqué sur ces dernières se divise en deux parties : l'une servant à évaluer les techniques de classification non-supervisée, l'autre servant à évaluer les techniques de détection d'entités nommées.

3.1.1 Annotation pour l'évaluation des techniques de classification non-supervisée

Notons d'abord que l'élaboration d'une taxonomie servant à catégoriser des requêtes de recherche web n'est pas une tâche simple pour laquelle il existe une solution unique. En effet, plusieurs chercheurs utilisent diverses taxonomies, lesquelles sont dépendantes de la nature de l'étude menée et de celle des requêtes étudiées. Parmi ces différents chercheurs, Broder [9] propose une taxonomie permettant de classifier une requête de recherche web selon trois catégories distinctes qui traduisent l'intention générale de l'émetteur de cette dernière. Cette taxonomie semble être utilisée par plusieurs chercheurs et dans un bon nombre d'articles. Les trois catégories la constituant sont définies comme suit :

- ***Navigational*** : l'intention de la requête est de visiter un site web en particulier qui est mentionné dans la requête. Exemple de requête : «imdb.com».
- ***Informational*** : l'intention de la requête est d'acquérir une certaine information qui pourrait se retrouver dans une ou plusieurs pages web. Exemple de requête : «how old is Bill Murray».
- ***Transactional*** : l'intention de la requête est de rejoindre un site où certaines interactions, telles que l'achat de produits en ligne ou le téléchargement de fichiers, seront effectuées. Exemple de requête : «buy bitcoins online».

Bien que cette taxonomie permette de faire une distinction claire entre trois groupes de requêtes qui peut être très utile pour certaines analyses, elle demeure tout de même assez générale et n'est pas suffisamment précise pour le genre d'étude qui est proposé dans ce travail. Une taxonomie légèrement plus précise est présentée par Spink et al. [24] dans leur étude portant sur un ensemble d'un million de requêtes soumises au moteur de recherche Excite. Cette taxonomie contient 11 catégories, dont les suivantes : *People-places-thing*, *Education-the humanities*, *Society-culture-ethnicity-religion*, *Unknown-incomprehensible*. Elle, comparativement à celle de Broder [9] qui est axé vers l'intention générale de l'usager, est plus axée sur le sujet global des requêtes et se rapproche plus du genre de taxonomie qui pourrait être utile dans le cadre du travail présenté ici. Différentes taxonomies sont également utilisées par d'autres chercheurs, dont Gan et al. [13], Kamvar et al. [15] et Rose et al. [21].

Comme l'indique le paragraphe précédent, l'élaboration d'un ensemble de catégories servant à catégoriser des requêtes de recherche web est dépendante de la nature de l'étude voulant être effectuée et de la nature du jeu de données étudié. Ainsi, une nouvelle catégorisation spécifique au travail présenté dans ce document et à la nature des requêtes étudiées sera créée suite à une observation approfondie de ces dernières. Au total, 57 étiquettes, chacune d'entre elles représentant un domaine général, ont été créées. Ces 57 étiquettes ainsi que leur distribution sur les 3 000 requêtes étiquetées sont présentées dans le Tableau 3.1.

Étiquette	Fréquence	Étiquette	Fréquence
TBD	575	Adult	190
Music	159	Medical	140
Location	130	Food	130
Car	124	Movie	122
Techno	109	Entertainment	98
Animal	97	Sport	97
Tv	86	Junk	85
Science	84	Restaurant	75
VideoGame	56	Book	50
Religion	44	Travel	35
Education	35	House	33
Law	32	Beauty	31
Health	31	Military	29
History	24	RealEstate	24
Politics	22	Language	21
Energy	18	GeneralShopping	17
Public	16	Finance	15
News	13	Plant	13
Culture	13	Clothing	13
Insurance	11	Command	11
Tool	10	Motorcycle	9
OutdoorRecreation	9	Transport	9
Radio	8	Social	7
Weather	7	Lottery	6
Hotel	6	Video	5
Business	4	Geography	4
Time	2	Agriculture	2
Construction	2	Audio	1
Event	1		

Tableau 3.1 – Distribution des étiquettes attribuées aux 3000 requêtes.

L'étiquette *TBD* (To Be Defined | To Be Determined) a été utilisée pour identifier les requêtes pour lesquelles un domaine était difficilement identifiable. Des exemples de certaines de ces requêtes sont présentés dans le Tableau 3.2a. Comme cette étiquette ne représente pas un domaine en soi, elle ne sera jamais considérée comme étiquette majoritaire d'un *cluster* dans le processus d'évaluation, et ce même s'il s'agit de l'étiquette ayant la plus grande fréquence parmi celles des requêtes étiquetées du *cluster* considéré. Il est important de souligner que cette étiquette est la plus fréquente dans l'ensemble des 3 000 requêtes étiquetées ($\sim$ 19%). Ceci semble indiquer que l'ensemble de requêtes considéré n'est pas nécessairement facile à diviser en groupe de requêtes partageant le même domaine et donc la tâche à laquelle ce travail s'attaque est relativement difficile étant donné les requêtes étudiées.

Requêtes TBD
inferno
data
melamine
who line
yearbook
hello somebody
best ufo design
balloon

(a) Requêtes *TBD*

Requêtes Junk
hey
what can
can you
oh
hello
what up bitches
blah blah blah
wtf

(b) Requêtes *Junk*

Tableau 3.2 – Exemples de requête.

Une autre étiquette similaire à l'étiquette *TBD* est l'étiquette *Junk*. Cette étiquette a été attribuée aux requêtes incomplètes, haineuses ou pour lesquelles il était évident que l'utilisateur n'était pas sérieux dans sa recherche. Le Tableau 3.2b présente certaines de ces requêtes. Pour ce qui est des 55 autres étiquettes, elles sont toutes reliées à un domaine en particulier.

3.1.2 Annotation pour l'évaluation des techniques de détection d'entités nommées

Le schéma d'annotation décrit précédemment sera utilisé pour évaluer les résultats obtenus par les techniques de classification non-supervisée considérées. Afin de pouvoir également calculer des mesures d'évaluation supervisées sur les résultats obtenus par le processus de détection d'entités nommées présenté à la section 2.6, les frontières des entités nommées contenues dans chacune des 3 000 requêtes sélectionnées seront également identifiées lors du processus d'annotation manuelle de ces requêtes.

3.2 Classification des requêtes par K-Means

On présentera dans cette section les résultats obtenus en appliquant aux 29 618 requêtes étudiées les différents algorithmes de type K-Means présentés à la section 2.2. Dans ces expériences, chaque requête sera uniquement représentée par les mots qui la constituent à l'aide d'un VSM du type TF-IDF. Le but derrière ces expérimentations est de donner un aperçu des limitations des algorithmes de classification non-supervisée qui ne se servent que des mots contenus à l'intérieur des requêtes pour effectuer leur classification. La faible pureté des *clusters* obtenus motivera l'exploration de techniques faisant l'utilisation d'informations supplémentaires, telles que le texte de la page web visitée par l'émetteur d'une requête, pour effectuer la classification d'un ensemble de requêtes de recherche web. De plus, il sera possible de remarquer aussi que l'algorithme *Spherical K-Means* présenté à la section 2.2.3 mène à de meilleurs résultats que l'algorithme K-Means standard. Il sera également possible de constater les effets de la méthode d'initialisation présentée à la section 2.2.2 sur les différents algorithmes considérés.

Les algorithmes testés sont :

- **K-Means** : algorithme K-Means de base avec initialisation aléatoire.
- **K-Means++** : algorithme K-Means de base avec la technique d'initialisation présentée à la section 2.2.2.
- **Spherical K-Means** : algorithme présenté à la section 2.2.3 avec initialisation

aléatoire.

– **Spherical K-Means++** : algorithme présenté à la section 2.2.3 avec la technique d'initialisation présentée à la section 2.2.2.

Plusieurs expériences ont été menées avec les différents algorithmes et différentes valeurs du paramètre K, qui représente le nombre de *cluster* à créer. Pour chaque algorithme et chaque valeur de K considérés, une séquence de 100 expérimentations a été effectuée. De plus, deux séquences d'expérimentations ont été effectuées pour chacun des algorithmes K-Means et K-Means++, une séquence utilisant les vecteurs donnés par la représentation TF-IDF des requêtes telle que décrite dans la section 2.1.1 et l'autre utilisant la version normalisée de ces vecteurs. Quelques détails supplémentaires concernant ces expériences :

– Nombre de requêtes à classifier : 29 618
– Taille du vocabulaire engendré par les requêtes : 19 583
– Représentation TF-IDF :
  – poids local : équation (2.1)
  – poids global : équation (2.4)

Les Tableaux 3.3 à 3.8 présentent des statistiques sur la pureté globale des ensembles de *clusters* obtenus lors des expérimentations et ce pour des valeurs de K égales à 50, 60, 70, 80, 90 et 100 respectivement.[1] Ces tableaux présentent également des statistiques sur le nombre d'itérations qui ont été nécessaires à chaque algorithme pour construire les ensembles de *clusters* obtenus. Il est possible de remarquer que peu importe la valeur de K, la pureté globale des *clusters* obtenus avec l'algorithme Sperical K-Means est supérieure, en moyenne, à celle obtenue avec l'algorithme K-Means et ce peu importe la méthode d'initialisation choisie et peu importe si la normalisation des vecteurs représentant les requêtes a été effectuée. De plus, peu importe la valeur de K, peu importe si les vecteurs ont été normalisés et peu importe que ce soit pour l'algorithme K-Means ou Spherical K-Means, la méthode d'initialisation présentée à la section 2.2.2 produit des ensembles de *clusters* dont la pureté globale

1. Dans ces tableaux, les algorithmes K-Means* et K-Means++* représentent les algorithmes K-Means et K-Means++ utilisant la version normalisée des vecteurs TF-IDF.

Algo	Nombre d'itérations				Pureté globale			
	min.	moy.	max.	é.t.	min.	moy.	max.	é.t.
K-Means	4	18,2	56	10,6	0,065	0,083	0,110	0,009
K-Means++	6	21,8	48	8,9	0,076	0,096	0,125	0,010
K-Means*	12	25,3	56	8,5	0,111	0,125	0,140	0,006
K-Means++*	15	29,7	62	9,7	0,127	0,140	0,153	0,004
S. K-Means	18	29,3	53	6,7	0,155	0,171	0,186	0,006
S. K-Means++	16	28,0	60	7,8	0,163	0,175	0,186	0,005

Tableau 3.3 – Évaluation des résultats obtenus avec les différents algorithmes considérés pour une valeur de $K = 50$.

Algo	Nombre d'itérations				Pureté globale			
	min.	moy.	max.	é.t.	min.	moy.	max.	é.t.
K-Means	5	18,8	60	10,5	0,069	0,086	0,110	0,009
K-Means++	5	21,8	44	8,1	0,079	0,102	0,130	0,011
K-Means*	11	27,3	59	10,4	0,113	0,131	0,145	0,006
K-Means++*	15	31,7	76	10,1	0,137	0,150	0,163	0,005
S. K-Means	19	30,3	55	8,0	0,167	0,179	0,190	0,005
S. K-Means++	17	27,9	56	7,5	0,172	0,185	0,199	0,005

Tableau 3.4 – Évaluation des résultats obtenus avec les différents algorithmes considérés pour une valeur de $K = 60$.

est supérieure, en moyenne, à celle des ensembles de *clusters* produits en utilisant une technique d'initialisation aléatoire. Ces résultats semblent donc indiquer que l'algorithme Spherical K-Means jumelé à la technique d'initialisation présentée à la section 2.2.2 est l'algorithme de type K-Means, parmi ceux qui ont été ici considérés, qui devrait être utilisé pour obtenir des ensembles de *clusters* avec la plus grande pureté globale possible.

Spherical K-Means : Création d'un *cluster* pour les données inclassables
La distance cosinus est bornée supérieurement par le nombre réel 1. Dans le cas ici étudié, deux requêtes seront à une distance cosinus de 1 si et seulement si elles ne partagent aucun mot l'une et l'autre. Or, il existe des requêtes dans l'ensemble de requêtes considéré qui se retrouvent à une distance cosinus de 1 avec chacune des requêtes de l'ensemble. Ainsi, à moins que cette requête soit choisie comme un des

Algo	Nombre d'itérations				Pureté globale			
	min.	moy.	max.	é.t.	min.	moy.	max.	é.t.
K-Means	5	19,2	55	9,8	0,068	0,087	0,115	0,010
K-Means++	6	23,2	50	9,1	0,076	0,105	0,134	0,011
K-Means*	13	25,1	74	9,4	0,122	0,137	0,150	0,006
K-Means++*	16	32,5	63	9,7	0,149	0,158	0,171	0,005
S. K-Means	18	29,5	59	8,0	0,171	0,187	0,199	0,006
S. K-Means++	15	26,0	52	6,3	0,181	0,193	0,208	0,005

Tableau 3.5 – Évaluation des résultats obtenus avec les différents algorithmes considérés pour une valeur de $K = 70$.

Algo	Nombre d'itérations				Pureté globale			
	min.	moy.	max.	é.t.	min.	moy.	max.	é.t.
K-Means	4	21,0	51	9,3	0,070	0,091	0,114	0,009
K-Means++	9	23,3	55	8,9	0,084	0,110	0,146	0,012
K-Means*	11	25,9	63	8,6	0,132	0,144	0,160	0,006
K-Means++*	17	30,0	61	9,4	0,151	0,165	0,177	0,005
S. K-Means	17	28,1	51	6,7	0,180	0,194	0,210	0,006
S. K-Means++	15	27,1	60	7,5	0,186	0,200	0,213	0,005

Tableau 3.6 – Évaluation des résultats obtenus avec les différents algorithmes considérés pour une valeur de $K = 80$.

Algo	Nombre d'itérations				Pureté globale			
	min.	moy.	max.	é.t.	min.	moy.	max.	é.t.
K-Means	6	22,1	55	11,6	0,077	0,092	0,122	0,011
K-Means++	8	20,3	39	6,0	0,086	0,105	0,132	0,013
K-Means*	14	22,5	30	4,2	0,139	0,149	0,161	0,006
K-Means++*	18	31,2	94	16,5	0,163	0,172	0,180	0,005
S. K-Means	16	29,8	50	8,8	0,189	0,202	0,220	0,007
S. K-Means++	20	24,9	38	4,6	0,200	0,208	0,214	0,004

Tableau 3.7 – Évaluation des résultats obtenus avec les différents algorithmes considérés pour une valeur de $K = 90$.

Algo	Nombre d'itérations				Pureté globale			
	min.	moy.	max.	é.t.	min.	moy.	max.	é.t.
K-Means	6	21,9	58	9,7	0,072	0,094	0,115	0,009
K-Means++	8	22,7	45	8,0	0,085	0,116	0,145	0,012
K-Means*	13	24,4	50	8,0	0,141	0,156	0,167	0,005
K-Means++*	16	29,6	65	8,4	0,170	0,180	0,193	0,005
S. K-Means	19	27,7	67	7,9	0,192	0,208	0,223	0,005
S. K-Means++	16	26,6	43	5,9	0,200	0,214	0,226	0,005

Tableau 3.8 – Évaluation des résultats obtenus avec les différents algorithmes considérés pour une valeur de $K = 100$.

prototypes initiaux lors de l'étape d'assignation de l'algorithme K-Means, tous les prototypes seront à une distance de 1 de cette requête. Dans ce cas, l'algorithme utilisé place la requête en question dans un $(K+1)^{\text{ième}}$ *cluster*. Les données de ce *cluster* seront considérées comme étant du bruit et ne seront pas considérées dans l'évaluation du résultat obtenu en fin d'algorithme. Dans les expérimentations qui ont été effectuées, ce $(K+1)^{\text{ième}}$ *cluster* contient, en fin d'algorithme, entre 1 157 et 1 163 requêtes. Cette variation peut être expliquée par la nature aléatoire du choix des prototypes de *cluster* initiaux pour amorcer l'algorithme.

De plus, étant donné le nombre moyen de mots constituant une requête $(3,8)$ et la taille du vocabulaire engendré par les 29 618 requêtes étudiées (19 583), en utilisant une représentation donnée par un VSM avec une pondération TF-IDF telle que décrite à la section 2.1.1, un grand nombre de paires de requêtes se verront attribuer une similarité cosinus de 0 et ce même si ces requêtes sont issues du même domaine. Par exemple, les requêtes «music by lady gaga» et «latest madonna album» sont toutes deux reliées au domaine de la musique, mais elles ne possèdent aucun mot en commun et donc la similarité cosinus entre ces dernières est de 0. Comme on s'intéresse ici à classifier les requêtes selon leurs domaines, il serait souhaitable d'obtenir une mesure de similarité non-nulle et préférablement élevée entre ces deux requêtes. Ce phénomène explique en parti la faible pureté des *clusters* obtenus par l'algorithme Spherical K-Means présentés dans cette section.

Les résultats qui ont été présentés dans cette section motivent l'utilisation d'informations additionnelles pour pouvoir effectuer une meilleure classification de l'ensemble de requêtes de recherche web considéré. Dans cette optique, la section 3.3 présentera les résultats obtenus par la technique de classification non-supervisée proposée dans le présent travail. Comme il a été mentionné précédemment, cette technique utilise la connaissance des pages web visitées par les émetteurs des requêtes comme information supplémentaire pour effectuer sa classification.

3.3 Classification des requêtes en utilisant le texte des pages web visitées

Cette section présentera les résultats obtenus en appliquant la technique de classification non-supervisée proposée dans ce travail, à la section 2.4, pour classifier les requêtes de recherche web étudiées. De façon plus détaillée, la section 3.3.1 présentera quelques détails sur l'application du processus d'extraction de texte, décrit à la section 2.4.1, sur l'ensemble des pages web considérées, la section 3.3.2 présentera les résultats intermédiaires obtenus par l'application des algorithmes LDA et TISK-LDA et finalement la section 3.3.3 présentera les résultats de la classification finale des requêtes obtenues en utilisant les résultats donnés par l'un où l'autre des algorithmes LDA et TISK-LDA.

3.3.1 Extraction du texte

L'ensemble de données étudié contient un total de 29 935 URLs de page web distincts. Un programme a parcouru ces 29 935 pages web pour en extraire le texte qu'elles contenaient et ainsi former un corpus de documents. Un problème a cependant été rencontré lors de ce processus d'extraction. En effet, certaines pages n'ont pas pu être consultées et uniquement 25 217 des pages web qui ont pu être consultées contenaient du texte. Le travail se fera donc à partir de ces 25 217 pages web.

Il est à noter que l'une des principales raisons pour lesquelles les pages web n'ont pas pu être consultées est représentée par le message d'erreur «We failed to reach the

server. Reason : Not Found». En effet, il est possible que certaines de ces pages web aient été retirées du web entre le moment où les émetteurs des requêtes qui leur sont associées les ont consultées et celui où le processus d'extraction de texte a été effectué pour ce travail (délais de 2 ans). Une plus courte période entre ces deux événements aurait sans doute pu mener à une plus grande quantité de textes extraits.

Domaines visités : Le Tableau 3.9 présente la distribution du domaine des 25 217 pages web visitées pour lesquelles il a été possible d'extraire du texte. Ce tableau permet de remarquer que certains domaines, tels que wikipedia.org, youtube.com, answers.yahoo.com et amazon.com, sont très fréquents. Pour cette raison, des méthodes d'extraction de texte particulières à chacune des pages web provenant de ces quatre domaines ont été utilisées afin d'éviter l'extraction de mots strictement relatifs aux sites et à leurs structures.

En plus de ceci, pour les mêmes raisons que celles énoncées dans le paragraphe précédent, pour chaque groupe de documents provenant des sites wikipedia.org, youtube.com answers.yahoo.com, amazon.com et ask.com, les mots se retrouvant dans plus de 40% des documents du groupe considéré ont été retranchés des documents du groupe en question. Ce processus de prétraitement a eu pour effet de retirer tous les mots de 45 des 25 217 documents. Ces 45 documents étant maintenant vides, le travail se fera sur un corpus de 25 172 documents.

Comme le corpus considéré passe de 29 935 documents à 25 172 documents, certaines requêtes se retrouveront liées à aucun des documents du corpus étudié et donc un processus de classification non-supervisée se servant uniquement de ces documents pour faire la classification des requêtes ne pourra pas classifier la totalité des requêtes de l'ensemble étudié. Dans ce dernier, le nombre de requêtes pouvant être classifiées par un tel processus est de 25 295 (sur un total de 29 618 requêtes).

Domaine du URL	Fréquence	Pourcentage	Cumulatif
wikipedia.org	5184	20,56%	20,56%
youtube.com	1792	7,11%	27,66%
answers.yahoo.com	1377	5,46%	33,12%
amazon.com	501	1,99%	35,11%
ask.com	252	1,00%	36,11%
pornhub.com	190	0,75%	36,86%
dictionary.com	176	0,70%	37,56%
imdb.com	160	0,63%	38,20%
yp.com	107	0,42%	38,62%
twitter.com	99	0,39%	39,01%
whitepages.com	98	0,39%	39,40%
tripadvisor.com	76	0,30%	39,70%
wikihow.com	75	0,30%	40,00%
manta.com	73	0,29%	40,29%
superpages.com	72	0,29%	40,58%
urbanspoon.com	71	0,28%	40,86%
askville.amazon.com	60	0,24%	41,10%
usablenet.com	60	0,24%	41,33%
apple.com	54	0,21%	41,55%
last.fm	48	0,19%	41,74%
edmunds.com	48	0,19%	41,93%
...	...	...	...
Total	25217		

Tableau 3.9 – Distribution des domaines des URLs

3.3.2 Application des algorithmes *Latent Dirichlet Allocation* et *Topic-In-Set Knowledge Latent Dirichlet Allocation*

Le processus de classification non-supervisée des pages web tel que décrit à la section 2.4.2 est principalement basé sur l'application de l'algorithme LDA ou de sa variante TISK-LDA sur l'ensemble des textes de pages web. Cette section sera donc consacrée aux détails de l'application de ces algorithmes sur les pages web étudiées et sur les *topics* obtenus en sortie de ces derniers.

Logiciel utilisé

L'implémentation de l'algorithme LDA qui a été utilisée ici est celle se retrouvant dans Mallet [19], qui est un logiciel écrit en Java qui contient l'implémentation de plusieurs algorithmes d'apprentissage artificiel orientés sur l'analyse de données textuelles. Il faut noter que le code source de l'algorithme a été modifié afin, entre autres, d'ajouter la modification présentée dans la section 2.3.9 à l'implémentation déjà présente de LDA.

Caractéristiques des lois de Dirichlet utilisées

Le vecteur $\boldsymbol{\alpha}$ et le scalaire η, jouant respectivement les rôles de paramètre de la loi de Dirichlet sur les distributions de *topics* et de paramètre de la loi de Dirichlet symétrique sur les distributions des mots du vocabulaire des *topics*, sont des quantités à fournir en entrée à l'algorithme LDA. L'algorithme LDA qui est présenté dans ce travail assume une distribution de Dirichlet asymétrique pour les distributions de *topics* des documents et une distribution symétrique pour les distributions des mots du vocabulaire des *topics*. Ce choix est appuyé par les résultats présentés par Wallach et al. [29]. Ce travail compare les différentes combinaisons possibles (asymétrique - asymétrique, asymétrique - symétrique, symétrique - symétrique et symétrique - asymétrique) pour les deux lois de Dirichlet utilisées dans l'algorithme LDA sur plusieurs jeux de données variés et présente de meilleurs résultats pour la combinaison asymétrique - symétrique. Il faut noter qu'il s'agit également d'un choix qui semble être

populaire dans la littérature en général.

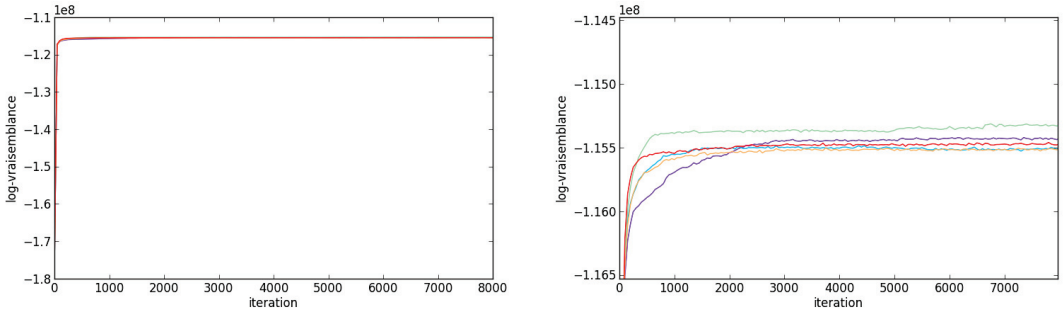
Convergence de la méthode de *Collapsed Gibbs Sampling*

L'algorithme de LDA utilisé dans le présent travail fait appel à un processus itératif du type MCCM, plus précisément à une technique de CGS, pour effectuer un tirage aléatoire du vecteur $\boldsymbol{z}$ selon la loi de probabilité $\mathbb{P}(\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\alpha}, \eta)$ donnée par le modèle. Bien que pour ce genre de technique la convergence vers la distribution souhaitée est théoriquement garantie [11], aucune méthode standard n'existe pour détecter cette convergence. Dans le cas du CGS de LDA, certains auteurs proposent d'utiliser la variation de la log-vraisemblance $\log\left[\mathbb{P}(\boldsymbol{w}_{1:M}, \boldsymbol{z}_{1:M}|\boldsymbol{\alpha}, \eta)\right]$ d'une itération à l'autre pour attester de cette convergence [12]. C'est le critère qui sera considéré dans ce travail.

Comme discuté dans la section 2.3.8, la vraisemblance est donnée par le produit $\mathbb{P}(\boldsymbol{w}_{1:M}|\boldsymbol{z}_{1:M}, \eta)\mathbb{P}(\boldsymbol{z}_{1:M}|\boldsymbol{\alpha})$. Les calculs de ces deux quantités sont donnés par les équations (2.15) et (2.16). La Figure 3.1 contient le graphique présentant la log-vraissemblance $\log\left[\mathbb{P}(\boldsymbol{w}_{1:M}, \boldsymbol{z}_{1:M}|\boldsymbol{\alpha}, \eta)\right]$ à chaque itération du processus de CGS lors de cinq applications de l'algorithme LDA sur le corpus étudié avec les valeurs de paramètres $\boldsymbol{\alpha} = (0,1;0,1;\ldots;0,1)$, $\eta = 0,01$ et $K = 50$ et où le processus d'optimisation des paramètres $\boldsymbol{\alpha}$ et η n'est pas activé. On peut remarquer que, pour les cinq expériences, après l'itération 3 000, la log-vraisemblance semble demeurer assez stable (croître très peu). Cette information sera utile pour déterminer le nombre d'itérations qui sera nécessaire de laisser passer avant de déclencher le processus d'optimisation des paramètres $\boldsymbol{\alpha}$ et η dans les expériences où ce processus sera activé.

Optimisation des paramètres $\boldsymbol{\alpha}$ et η

Étant donné que, dans ce cas ci, 3 000 semble être un bon estimé du nombre «maximum» d'itérations nécessaires pour que le processus de CGS converge (Figure 3.1), l'algorithme LDA sera appliqué de nouveau au corpus étudié, et ce avec les mêmes paramètres initiaux qu'auparavant, mais en activant cette fois le processus d'optimisation des paramètres $\boldsymbol{\alpha}$ et η. L'optimisation des paramètres se fera après



(a) Itérations 0 à 8000 (b) Itérations 0 à 8000 - Zoom

Figure 3.1 – Log-vraisemblance $\mathbb{P}(\boldsymbol{w}_{1:M}, \boldsymbol{z}_{1:M}|\boldsymbol{\alpha}, \eta)$ à chaque itération du processus de *Collapsed Gibbs Sampling* de l'algorithme *Latent Dirichlet Allocation*.

l'itération 3 250 et sera ré-appliquée après chaque tranche de 250 itérations. La Figure 3.2 présente un graphique permettant d'observer comment la log-vraisemblance $\log\left[\mathbb{P}(\boldsymbol{w}_{1:M}, \boldsymbol{z}_{1:M}|\boldsymbol{\alpha}, \eta)\right]$ varie au cours des itérations du processus de CGS et ce en incluant le processus d'optimisation des paramètres $\boldsymbol{\alpha}$ et η (lignes pleines) et de comparer l'évolution de ces variations avec l'évolution de celles obtenues lorsque ce processus n'est pas activé (lignes pointillées).

Comme il est possible d'observer dans la Figure 3.2, le processus d'optimisation des paramètres du modèle a bel et bien l'effet voulu, c'est-à-dire d'augmenter la valeur de la vraisemblance $\mathbb{P}(\boldsymbol{w}_{1:M}, \boldsymbol{z}_{1:M}|\boldsymbol{\alpha}, \eta)$. En effet, il est possible de remarquer une augmentation de la log-vraisemblance de manière plus prononcée aux itérations 3 250, 3 500, 3 750, ... qui sont les itérations où l'optimisation des paramètres a eu lieu. Ce graphique permet également de remarquer que l'optimisation des paramètres aux itérations 4 500, 4 750, 5 000, ... ne semble pas avoir une très grande incidence sur la variation de la vraisemblance. Ceci est probablement dût au fait que le processus de CGS a mené à une loi très «proche» de la loi désirée, ainsi les échantillons tirés d'itération en itération sont «équivalents» et donc les paramètres maximisant la vraisemblance sont sensiblement les mêmes.

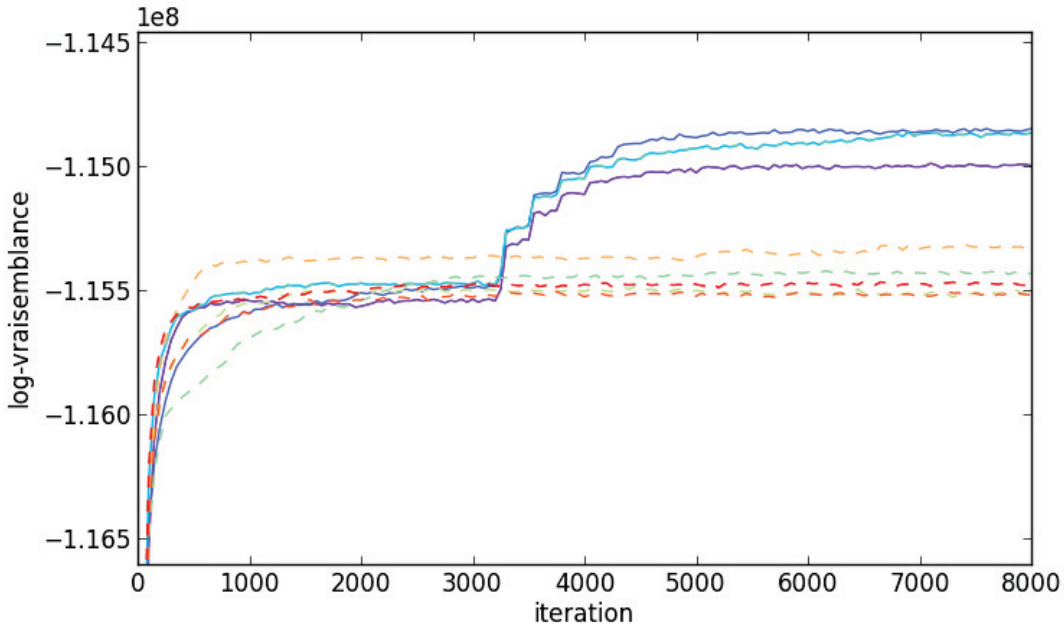


Figure 3.2 – Log-vraisemblance $\mathbb{P}(\boldsymbol{w}_{1:M}, \boldsymbol{z}_{1:M}|\boldsymbol{\alpha}, \eta)$ à chaque itération du processus de *Collapsed Gibbs Sampling* de l'algorithme *Latent Dirichlet Allocation* incluant les expérimentations où l'optimisation des paramètres $\boldsymbol{\alpha}$ et η était activée (lignes pleines) et non-activée (lignes pointillées)

Paramètre $\boldsymbol{\alpha}$ En effet, en utilisant la fonction

$$f(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\alpha}^{(j)}) = \frac{\sum_{k=1}^{K} |\alpha_k^{(i)} - \alpha_k^{(j)}|}{K} \tag{3.1}$$

pour comparer la variation des valeurs des composantes du vecteur de paramètres $\boldsymbol{\alpha}$ pour deux itérations i et j données, il est possible de comparer entres eux les différents vecteurs $\boldsymbol{\alpha}$ obtenu à la suite des processus d'optimisation aux itérations 3 250, 3 500, ..., 8 000. Le Tableau 3.10a présente la valeur de cette fonction pour les différentes valeurs de $\boldsymbol{\alpha}$ obtenues après les processus d'optimisation qui ont eu lieu aux itérations 3 250, 3 500, ..., 8 000. On peut remarquer que les vecteurs $\boldsymbol{\alpha}$ obtenus par les premiers processus d'optimisation sont beaucoup plus différents entre eux que ceux obtenus dans les derniers processus d'optimisation. En effet, la valeur de la fonction f est beaucoup plus faible à partir des itérations 5 250 - 5 500 (plus de

Itération	Var. Moyenne
0-3250	0,028494
3250-3500	0,014029
3500-3750	0,007288
3750-4000	0,004230
4000-4250	0,002546
4250-4500	0,001616
4500-4750	0,001093
4750-5000	0,000648
5000-5250	0,000573
5250-5500	0,000439
5500-5750	0,000406
5750-6000	0,000364
6000-6250	0,000302
6250-6500	0,000332
6500-6750	0,000451
6750-7000	0,000371
7000-7250	0,000435
7250-7500	0,000321
7500-7750	0,000420
7750-8000	0,000277

(a) Variation du paramètre $\boldsymbol{\alpha}$

Itération	Valeur	Variation
0	0,01	-
3250	0,009431	0,000569
3500	0,009485	0,000053
3750	0,009609	0,000125
4000	0,009706	0,000096
4250	0,009771	0,000065
4500	0,009812	0,000041
4750	0,009843	0,000031
5000	0,009860	0,000017
5250	0,009864	0,000004
5500	0,009882	0,000018
5750	0,009883	0,000001
6000	0,009895	0,000011
6250	0,009890	0,000004
6500	0,009892	0,000001
6750	0,009895	0,000003
7000	0,009890	0,000005
7250	0,009891	0,000001
7500	0,009905	0,000014
7750	0,009897	0,000008
8000	0,009906	0,000009

(b) Variation du paramètre η

Tableau 3.10 – Variation des paramètres α et η

50 fois plus faible) que la valeur observée pour les itérations du départ.

Paramètre η Pour ce qui est du paramètre η, comme il s'agit d'un scalaire la fonction de valeur absolue peut-être utilisée pour étudier les variations entre deux valeurs de η obtenues suite aux processus d'optimisation des itérations 3 250, 3 500, ..., 8 000. Le Tableau 3.10b présente la valeur du paramètre η après chaque processus d'optimisation ainsi que la variation de cette valeur entre deux processus d'optimisation successifs. On peut remarquer que cette valeur ne varie pas énormément d'une itération à l'autre et qu'elle semble tourner autour de $0,0099$.

Choix des valeurs de paramètres

Étant donné les observations précédentes, l'algorithme LDA qui sera utilisé dans le processus de classification des pages web dont il est ici question sera lancé en utilisant les paramètres suivants :

- nombre de *topics* : K=50
- loi de Dirichlet sur la distribution des *topics* pour les documents : asymétrique
- loi de Dirichlet sur la distribution des mots pour les *topics* : symétrique
- valeur initiale de $\boldsymbol{\alpha}$: $(0,1;0,1;\ldots;0,1)$
- valeur initiale de η : $0,01$
- optimisation de $\boldsymbol{\alpha}$ et η activée après 3250 itérations
- nombre d'itérations entre chaque processus d'optimisation : 250
- nombre total d'itérations : 8 000

Une analyse des *topics* obtenus suite à l'application, sur les pages web étudiées, de l'algorithme LDA utilisant ces paramètres est présentée dans les paragraphes suivants.

Topics obtenus par LDA

Le Tableau 3.11 présente les *topics* obtenus en appliquant l'algorithme LDA au corpus de pages web étudié avec les valeurs de paramètres mentionnées au paragraphe précédent. Ce tableau contient les quatre mots[2] les plus importants (les plus probables) de chaque *topic*. Les paragraphes qui suivent présentent quelques observations sur la nature des *topics* présentés dans ce tableau.

***Topic* relié à un domaine** Certains *topics* semblent être reliés à un domaine en particulier comme par exemple *song-music-album-record*, *school-univers-student-educ*, *food-recip-cook-fruit* ou encore *dog-speci-anim-cat*.

***Topic* non-relié à un domaine** D'autres ne semblent pas être reliés à un domaine en particulier comme par exemple *time-year-good-peopl*, *page-code-email-*

2. Ces mots sont modifiés par un processus de *stemming*.

Topic	Topic
time-year-good-peopl	news-video-photo-read
develop-work-human-person	page-code-email-number
offic-busi-program-center	citi-state-counti-north
episod-assist-john-michael	famili-death-die-man
park-open-event-ticket	price-shop-store-product
girl-man-boy-anim	countri-popul-citi-centuri
color-light-blue-black	school-univers-student-educ
pm-aug-august-hour	film-show-seri-episod
tax-price-compani-market	food-recip-cook-fruit
build-construct-oper-design	song-music-album-record
ca-san-angel-texa	health-medic-care-patient
gun-part-rifl-heart	fish-water-lake-river
word-languag-english-de	dog-speci-anim-cat
water-plant-produc-acid	presid-state-govern-nation
book-stori-publish-chapter	iphon-appl-app-phone
law-court-case-state	car-engin-model-vehicl
system-sound-power-electr	sex-video-fuck-view
review-hotel-restaur-travel	war-forc-armi-militari
system-earth-theori-measur	game-team-season-leagu
god-church-christian-king	post-repli-view-favorit
hair-nude-pic-celebr	diseas-drug-blood-effect
game-player-level-play	movi-tv-review-list
team-win-match-cup	war-star-power-luke
women-sexual-sex-men	heat-jpg-al-air
sale-estat-sqft-bed	share-permalink-cathol-option

Tableau 3.11 – Pour chaque *topic* obtenu par l'algorithme *Latent Dirichlet Allocation*, les quatre mots les plus importants (ou encore les plus probables) du *topic*.

number, *share-permalink-cathol-option*, *pm-aug-august-hour* ou encore *post-repli-view-favorit*.

***Topic* relié à plusieurs domaines** De plus, quelques *topics* semblent être reliés à plusieurs domaines, comme *review-hotel-restaur-travel* qui semble traiter des domaines de la restauration et de l'hôtellerie. Bien sûr, ceci dépend de notre définition et division des domaines. Quelqu'un aurait très bien pu dire que ce *topic* est relié au domaine du tourisme.

***Topics* traitant du même domaine** Finalement, il semble aussi y avoir des *topics* traitant du même domaine, comme par exemple les *topics* : *health-medic-care-patient* et *diseas-drug-blood-effect* semblent tous deux traiter du domaine de la médecine.

Dans le cadre de l'application dont il est ici question (c.-à-d. la classification de requêtes de recherche web selon leurs domaines), les résultats de LDA sont utilisés pour faire la classification des pages web visitée par les émetteurs des requêtes considérées pour ensuite effectuer la classification de ses requêtes. Comme chaque *topic* sera éventuellement associé à un *cluster*, il est souhaitable d'obtenir des *topics* reliés à un seul domaine (p. ex. *song-music-album-record* ou encore *food-recip-cook-fruit*) et non à plusieurs domaines (p. ex. *review-hotel-restaur-travel*). Pour remédier à ce problème de *topic multi-domaine*, l'algorithme de TISK-LDA pourrait être utilisé en lui fournissant en entrée des groupes de mots spécifiques à chaque domaine traité dans le *topic multi-domaine* de façon à séparer ces domaines dans plusieurs *topics*.

Topics obtenus par TISK-LDA

Pour éviter d'obtenir des résultats tels que décrits dans les deux paragraphes précédents (*topic* concernant plus d'un domaine et domaine représenté par plus d'un *topic*) et pour aider à orienter les *topics* vers certains domaines d'intérêts, la version modifiée de l'algorithme LDA présentée dans la section 2.3.9 de ce document sera appliquée sur le corpus de pages web étudié. Les groupes de mots utilisés dans ce travail sont présentés à l'annexe C dans le Tableau C.1. Chaque ligne de ce tableau

est associée à un groupe de mots qui servira à diriger un *topic*. La première colonne de ce tableau contient un identificateur numérique pour chaque groupe de mots, la deuxième colonne présente le domaine associé à chaque groupe de mots et la troisième colonne, les groupes de mots eux-mêmes. Une brève explication de la raison d'être de ces groupes de mots est donnée dans les paragraphes suivants.

Restauration et hôtellerie Un des groupes de mots créés, portant l'identifiant 3, est constitué de mots portant sur le domaine de l'hôtellerie et un autre, portant l'identifiant 1, est constitué de mots portant sur le domaine de la restauration. Ces deux groupes de mots ont été créés dans le but d'obtenir, en sortie de l'algorithme TISK-LDA, deux *topics* distincts traitant respectivement de ces deux domaines plutôt que d'obtenir un seul *topic* traitant des deux domaines comme il semble que ce soit le cas dans les résultats obtenus avec LDA, présentés dans le Tableau 3.11 (*topic review-hotel-restaur-travel*). De plus, afin de prévenir l'obtention d'un *topic* relié à la fois à la restauration et à la nourriture en général (information et recette), le groupe portant l'identificateur 2 a été créé. Ainsi, ces trois groupes de mots donnés en entrée à TISK-LDA ont pour but de créer trois *topics* distincts traitant respectivement d'hôtellerie, de restauration et de nourriture en général.

Cinéma et télévision Dans les résultats de LDA présentés précédemment, les deux *topics film-show-seri-episod* et *movi-tv-review-list* semblent traiter tous les deux à la fois de cinéma et de télévision. Afin de tenter d'obtenir un *topic* traitant de cinéma et un autre traitant de télévision, les deux groupes de mots portant les identifiants 4 et 5 ont été créés.

Le choix des sujets traités par les autres groupes de mots a été fait de façon à tester à quel point il était possible de diriger certains *topics* vers certains thèmes. Les thèmes (domaines) choisis pour ces groupes de mots ont été déterminés en connaissance du type des requêtes présentes dans le jeu de données qui est à l'étude dans ce travail. De plus, bien que les mots formant ces groupes de mots directeurs ont été choisis après une sommaire inspection des *topic* générer par LDA, un souci particulier a été apporté afin de fournir des mots qui sont somme toute assez communs relativement

à leurs domaines.

Le Tableau 3.12 présente, pour chaque *topic*, les quatre mots apparaissant le plus souvent dans le *topic* et ce basé sur les résultats donnés par l'application, sur l'ensemble de pages web ici étudié, de l'algorithme TISK-LDA utilisant les mêmes paramètres qui ont été utilisés avec l'algorithme LDA dont il a été plutôt question, en ajoutant cependant en entrée les groupes de mots contenus dans le Tableau C.1. Dans ce tableau, les *topics* qui n'ont pas été dirigés par un des groupes de mots présentés dans le Tableau C.1 sont mis en italique. De plus, pour chacun des *topics* qui ont été dirigés, l'identifiant du groupe de mots qui leur est associé est indiqué entre parenthèses.

Dans le Tableau 3.12, le nom des *topics* qui ont été dirigés par un des groupes de mots donnés en entrée semble bien exprimer le thème qui était suggéré par ces groupes de mots. En particulier, comme souhaité, trois *topics* traitant respectivement d'hôtellerie, de restauration et de nourriture en général ont été créés (les topics *review-hotel-travel-room*, *restaur-food-drink-bar* et *food-recip-cook-fruit*). De plus les *topics* *show-episod-seri-tv* et *film-movi-director-share* semblent bien séparer les thèmes du cinéma et de la télévision. Aussi, comme voulu, il ne semble plus y avoir qu'un seul *topic* relié à la pornographie. Les autres groupes de mots semblent également avoir bien dirigé leurs *topics* respectifs.

Bien que les groupes de mots fournis en entrée semblent avoir donné les résultats escomptés quant à la création des *topics*, il reste à vérifier si cette modification apporte un gain réel dans le processus entier de classification de requêtes dont il est ici question. Les résultats présentés dans la prochaine section permettront d'évaluer si cette modification apporte ou non un certain gain dans l'entièreté de ce processus.

3.3.3 Classification des requêtes

Étant donné les résultats obtenus par l'application de l'un ou l'autre des algorithmes LDA et TISK-LDA, cette section présentera les résultats obtenus en utilisant la technique énoncée dans la section 2.4.2 pour créer une partition de l'ensemble des

Topic	Topic
news-video-read-photo	*time-good-love-peopl*
show-episod-seri-tv (5)	book-publish-stori-author (11)
tax-state-number-requir	*develop-human-person-studi*
film-movi-director-share (4)	*school-univers-student-program*
music-song-album-record (10)	*st-state-north-counti*
team-game-player-sport (7)	health-medic-effect-diseas (8)
restaur-food-drink-bar (1)	car-engin-vehicl-model (13)
park-event-ticket-club	*price-shop-store-ship*
pm-aug-august-day	*water-product-heat-air*
iphon-appl-app-download	govern-presid-state-nation (12)
compani-market-product-busi	dog-anim-speci-cat (9)
countri-popul-island-nation	food-recip-cook-fruit (2)
sex-video-girl-fuck (6)	*citi-build-area-park*
left-head-side-turn	*ca-san-texa-code*
color-black-light-white	*art-design-paint-wed*
game-play-player-video (14)	god-church-christian-cathol (15)
review-hotel-travel-room (3)	*plant-tree-water-wind*
word-languag-english-letter	*law-court-case-crime*
bodi-weight-exercis-skin	*forc-war-armi-oper*
system-earth-theori-measur	*page-data-al-test*
network-view-famili-user	*post-repli-favorit-tweet*
war-centuri-german-french	*fish-lake-water-boat*
de-la-el-fight	*war-star-kill-luke*
acid-cell-mg-form	*jpg-sound-imag-radio*
open-door-estat-sale	*gun-rifl-paintbal-barrel*

Tableau 3.12 – Pour chaque *topic* obtenu par l'algorithme *Topic-In-Set Knowledge Latent Dirichlet Allocation*, les quatre mots apparaissant le plus souvent dans ce *topic*.

pages web (URLs) pour ensuite utiliser la technique énoncée dans la section 2.4.3 pour partitionner les requêtes. Les résultats pouvant être obtenus avec les résultats donnés par LDA seront d'abord présentés, pour ensuite présenter ceux pouvant être obtenus avec les résultats donnés par TISK-LDA. Lorsque des pourcentages de pureté seront donnés dans cette section, ils seront basés sur les 3 000 requêtes manuellement étiquetées dont il est question à la section 3.1.

Classification basée sur les résultats de l'algorithme LDA

Il faut d'abord noter que le processus de classification des requêtes débute par l'application de la technique de classification de pages web, présentée à la section 2.4.2, basée sur les résultats de l'application de l'algorithme LDA. Ce processus permet de générer un ensemble de K *clusters* de pages web. Ces K *clusters* de pages web sont ensuite utilisés pour effectuer la classification des requêtes en suivant la technique présentée à la section 2.4.3. Ce sont les résultats de cette technique de classification des requêtes qui sont présentés dans cette section.

En appliquant les étapes mentionnées précédemment pour effectuer la classification des requêtes, un ensemble de *clusters* de requêtes d'une pureté globale de 40% est obtenu, où la pureté des différents *clusters* formés varie de 7% à 93%. Les prochains paragraphes discuteront de certaines caractéristiques des *clusters* obtenus dont entre autres la faible valeur de pureté de certains d'entre eux. Pour accompagner cette discussion, les Tableaux 3.13 et 3.14 présentent, pour chaque *cluster* de requêtes, sa taille, sa pureté, le nom du *topic* qui lui est associé, l'étiquette majoritaire des requêtes étiquetées qu'il contient et la valeur de rappel rattachée à cette étiquette pour ce *cluster*.

Une des premières choses qu'il est possible de remarquer est la faible valeur de pureté attribuée à certains *clusters*. Hors, il est important de mentionner que cette pureté est relative à la manière dont les 3 000 requêtes de l'ensemble d'évaluation ont été étiquetées. En effet, les étiquettes apposées ont été orientées vers le domaine de la requête. Ceci aura pour effet qu'une requête traitant par exemple d'un concessionnaire

Nom du *cluster*	Taille	Étiquette	Pureté	Rappel
sex-video-fuck-view	699	Adult	0.93	0.48
game-team-season-leagu	449	Sport	0.88	0.37
car-engin-model-vehicl	698	Car	0.87	0.62
dog-speci-anim-cat	596	Animal	0.82	0.63
movi-tv-review-list	363	Movie	0.79	0.36
system-earth-theori-measur	353	Science	0.76	0.42
song-music-album-record	1031	Music	0.76	0.55
game-player-level-play	400	VideoGame	0.74	0.66
food-recip-cook-fruit	973	Food	0.74	0.69
school-univers-student-educ	354	Education	0.70	0.70
diseas-drug-blood-effect	499	Medical	0.69	0.29
book-stori-publish-chapter	354	Book	0.67	0.37
god-church-christian-king	375	Religion	0.61	0.79
sale-estat-sqft-bed	204	RealEstate	0.59	0.71
iphon-appl-app-phone	669	Techno	0.59	0.46
team-win-match-cup	292	Sport	0.56	0.23
countri-popul-citi-centuri	344	Location	0.56	0.17
review-hotel-restaur-travel	399	Restaurant	0.54	0.31
health-medic-care-patient	683	Medical	0.51	0.29
women-sexual-sex-men	191	Adult	0.50	0.08
film-show-seri-episod	645	Tv	0.47	0.38
law-court-case-state	237	Law	0.40	0.26
hair-nude-pic-celebr	239	Adult	0.38	0.06
presid-state-govern-nation	284	Politics	0.37	0.64
fish-water-lake-river	398	Location	0.37	0.13

Tableau 3.13 – Le nom, la taille, la pureté, l'étiquette majoritaire et le rappel qui lui est associé pour chaque *cluster* de requêtes obtenu en se basant sur les résultats de l'algorithme LDA

Nom du *cluster*	Taille	Étiquette	Pureté	Rappel
share-permalink-cathol-option	47	Military	0.33	0.04
episod-assist-john-michael	245	Movie	0.32	0.05
water-plant-produc-acid	422	Science	0.31	0.15
word-languag-english-de	505	Language	0.30	0.76
park-open-event-ticket	1092	Entertainment	0.28	0.40
build-construct-oper-design	308	Travel	0.27	0.24
war-star-power-luke	169	Movie	0.26	0.06
war-forc-armi-militari	194	Military	0.26	0.19
gun-part-rifl-heart	514	Military	0.24	0.48
girl-man-boy-anim	557	Junk	0.23	0.13
system-sound-power-electr	288	Music	0.21	0.05
heat-jpg-al-air	116	Location	0.20	0.03
tax-price-compani-market	363	Finance	0.17	0.38
citi-state-counti-north	587	Location	0.17	0.09
ca-san-angel-texa	334	Location	0.17	0.06
famili-death-die-man	412	Movie	0.15	0.05
post-repli-view-favorit	363	Adult	0.14	0.03
develop-work-human-person	379	Book	0.12	0.10
page-code-email-number	1670	Techno	0.11	0.21
time-year-good-peopl	1028	Music	0.10	0.08
price-shop-store-product	838	Clothing	0.10	0.08
color-light-blue-black	288	Plant	0.10	0.23
pm-aug-august-hour	305	Beauty	0.08	0.10
news-video-photo-read	1076	Adult	0.08	0.06
offic-busi-program-center	1463	Public	0.07	0.09

Tableau 3.14 – Suite - Le nom, la taille, la pureté, l'étiquette majoritaire et le rappel qui lui est associé pour chaque *cluster* de requêtes obtenu en se basant sur les résultats de l'algorithme LDA

automobile se verra apposer l'étiquette *Car* et non une étiquette du genre *Service* ou *Company*, qui pourraient également être des choix d'étiquettes valables.

À titre d'exemple, le choix d'étiquette qui a été fait peut expliquer en partie la faible pureté du *cluster* *offic-busi-program-center*, qui est estimée à 7 %. En effet, bien que les requêtes contenues dans ce *cluster* ne semblent pas être toutes reliées à un même domaine associé à une des étiquettes utilisées ici pour annoter les requêtes, un certain nombre d'entre elles semblent être reliées à un service ou à une compagnie. Afin d'illustrer cette remarque, le Tableau 3.15 présente 15 requêtes, accompagnées de leurs étiquettes, choisies aléatoirement parmi les 149 requêtes étiquetées contenues dans le *cluster* *offic-busi-program-center*. On peut remarquer qu'en effet un certain nombre (11/15) de ces requêtes sont en fait reliées à un service ou une compagnie quelconque (identifiées par un astérisque dans la troisième colonne du tableau). Il est également intéressant de noter que le nom du *cluster*, *offic-busi-program-center*, est tout à fait cohérent avec le thème *services et compagnies* qui semble bien représenter une bonne partie des requêtes du *cluster* en question.

L'observation liée au *cluster* *offic-busi-program-center* mentionnée dans le paragraphe précédent s'avère également vrai pour d'autres *clusters* possédant une valeur de pureté plutôt faible. Afin de fournir un deuxième exemple, le Tableau 3.16 présente 15 requêtes, accompagnées de leurs étiquettes, choisies aléatoirement parmi les 101 requêtes étiquetées contenues dans le *cluster* *price-shop-store-product*, *cluster* dont la valeur de pureté est de 10 %. Il est intéressant de noter que le nom du *cluster*, *price-shop-store-product*, semble être relié au thème du magasinage et qu'un certain nombre de requêtes présentées dans le Tableau 3.16 (identifiées par un astérisque en troisième colonne) peuvent être vues comme étant en lien avec un produit ou un magasin et donc en lien avec le thème général du magasinage. Ainsi, tout comme pour le *cluster* *offic-busi-program-center*, une grande quantité de requêtes du *cluster* semblent avoir un certain lien cohérent entres elles, même si ce lien n'est pas reflété par les étiquettes qui leur ont été apposées.

Requêtes	Étiquettes	
quincy compressors	Tool	*
bluffton indiana street fair	TBD	-
kv works	TBD	*
greenwood mississippi wastewater plant	Public	*
pinellas county water authority	Public	*
merit electric incorporated	Energy	*
christian brothers tire little rock arkansas	Car	*
emancipation proclamation	History	-
health careers	TBD	-
where can i go hang gliding	Entertainment	*
montana drivers license division	Public	*
orange county diagnostics	Medical	*
new york state firemen 's home	Medical	*
flight standards district office in grand rapids michigan	Public	*
find contact r***** s*****	TBD	-

Tableau 3.15 – Quinze requêtes, accompagnées de leurs étiquettes, choisies aléatoirement parmi les 149 requêtes étiquetées contenues dans le *cluster* *offic-busi-program-center*

L'observation faite pour les deux *clusters* *offic-busi-program-center* et *price-shop-store-product* dans les paragraphes précédents permet de nuancer la faible valeur de pureté calculée pour certains *clusters*. En effet, bien qu'en se basant sur la taxonomie empruntée pour étiqueter les 3 000 requêtes d'évaluation, certains *clusters* soient considérés comme étant très impurs, certains d'entre eux contiennent tout de même un certain nombre de requêtes qui peuvent être regroupées sous un même thème.

Il est cependant important à noter que l'observation liée aux *clusters* *offic-busi-program-center* et *price-shop-store-product* mentionnée dans les paragraphes précédents n'est pas nécessairement vrai pour tous les *clusters* possédant une valeur de pureté plutôt faible. En effet, il peut être difficile de constater une certaine cohérence entre les requêtes de certains d'entre eux. À titre d'exemple, les Tableaux 3.17 et 3.18 présentent chacun 15 requêtes, accompagnées de leurs étiquettes, choisies aléatoirement parmi les requêtes étiquetées contenues dans les *clusters* *time-year-good-peopl* et *color-light-blue-black* respectivement, *clusters* dont les valeurs de pureté sont toutes

Requêtes	Étiquettes	
coupons	TBD	*
eleanor rose	Clothing	*
natick collection	GeneralShopping	*
personalized horse lead ropes	Animal	*
crate and barrel	House	*
merona flip flops	Clothing	*
academy sport	Sport	*
w00t	GeneralShopping	*
date fruit chinese	Food	-
frozen yogurt location	Food	*
cruiser customizing	Motorcycle	*
abc distributing	GeneralShopping	*
looney tune for gameboy	VideoGame	*
kayaks	Entertainment	*
lapd	Public	-

Tableau 3.16 – Quinze requêtes, accompagnées de leurs étiquettes, choisies aléatoirement parmi les 101 requêtes étiquetées contenues dans le *cluster* *price-shop-store-product*

deux de 10 %. Il peut être intéressant de remarquer que les noms de ces deux *clusters* semblent beaucoup moins orientés vers un thème concret de recherche web comme le sont les noms des *clusters* *offic-busi-program-center* et *price-shop-store-product*.

Les paragraphes précédents se sont penchés sur les *clusters* ayant une faible valeur de pureté. Les prochains paragraphes présenteront quelques *clusters* ayant une valeur de pureté plus élevée, afin montrer également à quel point certains *clusters* peuvent présenter des regroupements de requêtes intéressants.

Le nom du *cluster* *game-team-season-leagu* semble indiquer que ce dernier contiendra des requêtes liées au sport et en effet, 88 % des 40 requêtes étiquetées contenues dans ce *cluster* se sont vues attribuer l'étiquette *Sport*. Afin de donner un aperçu du genre de requêtes se retrouvant dans ce *cluster*, le Tableau 3.19 présente 15 requêtes, accompagnées de leurs étiquettes, choisies aléatoirement parmi les 40 requêtes étiquetées. Une seule de ces 15 requêtes n'est pas associée avec l'étiquette *sport* : la requête

Requêtes	Étiquettes
back scratcher	TBD
how to feel you got more hours of sleep	Health
bad headaches	Medical
bulging balls	Medical
butt hole	Adult
find rodney saulsberry music	Music
different ways for women to masturbate	TBD
kaze v three	Tv
i hate you	Junk
how to get over an obsession	TBD
my face is weird	Junk
horror effect plug in fcp	Techno
find nearest prostitute	Adult
eb einstein sleeper	TBD
my car won't start why	Car

Tableau 3.17 – Quinze requêtes, accompagnées de leurs étiquettes, choisies aléatoirement parmi les 128 requêtes étiquetées contenues dans le *cluster* *time-year-good-peopl*

Requêtes	Étiquettes
show me the us state flags	TBD
what is the difference between a bush and a shrub	Plant
new york state motto	Location
skies of blue flowers and leedy	TBD
sending flowers	TBD
led lightbulbs	Energy
show me candice olson lighting	House
australopithecus	TBD
prayer beads	Religion
who painted the monalisa	TBD
arabesque	TBD
boston henna net	TBD
effects of black light tattoos	TBD
drawing app	Techno
free rosetta stone trial	Techno

Tableau 3.18 – Quinze requêtes, accompagnées de leurs étiquettes, choisies aléatoirement parmi les 30 requêtes étiquetées contenues dans le *cluster* *color-light-blue-black*

Requêtes	Étiquettes
start time for the rangers game	Sport
buffalo wild	Restaurant
what was the score of the alabama florida game	Sport
katie meyer	Sport
aaron garcia	Sport
nineteen seventy nine michigan quarterback	Sport
bill oliver he's okay	Sport
starting pitchers for yankees tonight against twins	Sport
florida hurricanes schedule	Sport
what conference are the texas longhorns in	Sport
two thousand three detroit tigers	Sport
david klingler	Sport
what time does the cleveland barons play today	Sport
amare stoudemire stats	Sport
what time did the mariners game start today	Sport

Tableau 3.19 – Quinze requêtes, accompagnées de leurs étiquettes, choisies aléatoirement parmi les 40 requêtes étiquetées contenues dans le *cluster* *game-team-season-leagu*

«buffalo wild». En effet «buffalo wild» semble référer à un restaurant-bar sportif et la requête a donc été étiquetée comme faisant partie du domaine *Restaurant*, ce qui est tout à fait correct. La cause de cette erreur se situe dans le fait que cette requête traite d'un restaurant-bar *sportif* et donc la page web qui lui est associée contient des termes fortement associés au monde du sport.

Un autre *cluster* intéressant qui possède une valeur relativement élevée de pureté (67 % des 27 requêtes étiquetées sont associées à l'étiquette *Book*) est le *cluster* nommé *book-stori-publish-chapter*. Le nom de ce *cluster* semble indiquer qu'il contiendra des requêtes reliées au monde de la publication et de la littérature, ce qui est en effet le cas. Tout comme pour les *clusters* dont il a été question précédemment, le Tableau 3.20, présente 15 requêtes, accompagnées de leurs étiquettes, choisies aléatoirement parmi les requêtes étiquetées contenues dans le *cluster* étudié. Seront discutés ici quelques faits intéressants reliés à ces 15 requêtes. Tout d'abord, 10 de ces 15 requêtes se sont vues attribuer l'étiquette *Book*, ce qui semble concorder avec la valeur de pureté

Requêtes	Étiquettes
exciting novels	Book
the book the way you wear your hat	Book
or what	Junk
shit books	Junk
david sedaris	Book
lucy larson	Book
the twelve kingdoms book	Book
incognito the secret lives of the brain	Book
the thirty nine clues book twelve	Book
interview with jk rolling about her religion	Book
leigh miller's lips	TBD
city walks with kids in rome	Entertainment
preston and child	Book
books about network plus	Book
open context book	TBD

Tableau 3.20 – Quinze requêtes, accompagnées de leurs étiquettes, choisies aléatoirement parmi les 27 requêtes étiquetées contenues dans le *cluster* *book-stori-publish-chapter*

de 67 % calculée à partir des 27 requêtes étiquetées. Du plus, il est intéressant de constater que parmi les cinq requêtes qui ne sont pas associées à l'étiquette *Book*, au moins deux requêtes sont clairement associées au domaine du livre et auraient dû être étiquetés comme telle. En effet, la requête «city walks with kids in rome» réfère à une collection de livres de tourisme «city walks with kids» et «leigh miller's lips» semble être reliée à l'écrivaine Robin Leigh Miller. Ainsi, le processus de classification ici utilisé a été capable d'identifier le domaine de requêtes pour lesquelles même un humain a éprouvé de la difficulté à effectuer la tâche. En ce qui concerne les requêtes «shit books» et «open context book», associées respectivement aux étiquettes *Junk* et *TBD*, leur appartenance à un *cluster* traitant de livre et de littérature est discutable.

Il est intéressant de remarquer que chaque groupe de requêtes présentés dans les tableaux précédents contient des requêtes qui ne possèdent pas forcément de mots en commun. Ceci vient appuyer le fait que de considérer la source d'information supplémentaire que sont les pages web visitées permet d'obtenir des groupes de requêtes qui

serait difficiles voir impossible d'obtenir en utilisant un algorithme qui se base uniquement sur les mots qui composent ces dernières pour effectuer le partitionnement de l'ensemble de requêtes.

Les résultats de classification de requêtes qui viennent d'être présentés ont été obtenus en utilisant, à l'intérieur du processus, les résultats de l'algorithme LDA présentés dans la section 3.3.2. La section qui suit présentera les résultats obtenus en utilisant les résultats de l'algorithme TISK-LDA présentés eux aussi dans la section 3.3.2 au lieu de ceux de LDA.

Classification basée sur les résultats de l'algorithme TISK-LDA

Les paragraphes qui suivent présenteront les résultats obtenus par le même processus de classification de requêtes qui a été utilisé pour obtenir les résultats présentés dans les paragraphes précédents à l'exception que les résultats de l'algorithme TISK-LDA présentés dans la section 3.3.2 sont utilisés au lieu des résultats de l'algorithme LDA standard.

Il faut tout d'abord noter que la valeur de pureté globale obtenue, calculée à l'aide des 3 000 requêtes manuellement annotées présentées à la section 3.1, est de 40 %, ce qui est également la valeur de pureté globale obtenue en utilisant simplement l'algorithme LDA. De plus, tout comme pour les résultats obtenus avec l'algorithme LDA, la valeur de pureté des cinquante *clusters* varie beaucoup d'un *cluster* à l'autre (valeurs allant de 7 % à 91 %). Cependant, en ne considérant que les quinze *clusters* qui sont associés aux quinze *topics* qui ont été dirigés par un groupe de mots, la valeur de pureté globale passe de 40 % à 67 %, avec des valeurs de pureté allant de 29 % à 91 %. Il faut ici noter que ces quinze *clusters* contiennent 9 316 des 25 292 requêtes considérées. Afin de donner plus de détails, le Tableau 3.21 présente, pour chaque *cluster* de requêtes associés à un *topic* qui a été dirigé par un des groupes de mots présentés dans le Tableau C.1, sa taille, sa pureté, le nom du *topic* qui lui est associé [3], l'étiquette majoritaire des requêtes étiquetées qu'il contient et la valeur de

3. l'identifiant du groupe de mots qui lui est associé, présenté dans le Tableau C.1, est indiqué entre parenthèses

Nom du *cluster*	Taille	Étiquette	Pureté	Rappel
sex-video-girl-fuck (6)	860	Adult	0.91	0.54
food-recip-cook-fruit (2)	740	Food	0.85	0.58
dog-anim-speci-cat (9)	625	Animal	0.84	0.66
car-engin-vehicl-model (13)	718	Car	0.80	0.62
team-game-player-sport (7)	624	Sport	0.75	0.54
god-church-christian-cathol (15)	284	Religion	0.73	0.70
film-movi-director-share (4)	603	Movie	0.70	0.37
music-song-album-record (10)	1133	Music	0.69	0.60
health-medic-effect-diseas (8)	724	Medical	0.65	0.40
game-play-player-video (14)	499	VideoGame	0.63	0.68
book-publish-stori-author (11)	597	Book	0.52	0.55
show-episod-seri-tv (5)	692	Tv	0.46	0.44
restaur-food-drink-bar (1)	676	Restaurant	0.42	0.56
govern-presid-state-nation (12)	318	Politics	0.30	0.64
review-hotel-travel-room (3)	223	Travel	0.29	0.18

Tableau 3.21 – Le nom, la taille, la pureté, l'étiquette majoritaire et le rappel qui lui est associé pour chaque *cluster* de requêtes associé à un topic qui a été dirigé par un des groupes de mots donnés en entrée à l'algorithme *Topic-In-Set Knowledge Latent Dirichlet Allocation*

rappel rattachée à cette étiquette pour ce *cluster*.

Tout d'abord, il est intéressant de remarquer que, à l'exception du *cluster review-hotel-travel-room*, l'étiquette majoritaire de chacun des *clusters* associés à un *topic* dirigé par un groupe de mots concorde avec l'étiquette associée à ce groupe de mots dans le Tableau C.1. Pour ce qui est du *cluster review-hotel-travel-room*, l'étiquette majoritaire qui aurait été souhaitable de lui voir associé, au lieu de l'étiquette *Travel*, est l'étiquette *Hotel*. Or, ce *cluster* ne contient qu'une requête étiquetée *Hotel* parmi les 21 requêtes étiquetées qu'il contient. Et donc pour ce *cluster*, la tentative de diriger le *cluster* vers le domaine de l'hôtellerie n'a pas été un succès. Cependant, le domaine du voyage représenté par l'étiquette *Travel* n'est pas si loin du domaine de l'hôtellerie représenté par l'étiquette *Hotel* et donc dans ce sens, le *cluster* n'est pas complètement incohérent.

Il est également important de souligner que bien que l'utilisation de TISK-LDA ne semble pas apporter d'amélioration, par rapport à l'utilisation de LDA, au niveau de la pureté globale de l'ensemble des *clusters* créés, elle permet toutefois de cibler directement, parmi les *clusters* créés, les *clusters* qui ont de l'intérêt pour l'utilisateur de cette technique. En effet, comme l'utilisateur fourni lui-même en entrée des groupes de mots associés aux domaines pour lesquels il possède un certain intérêt, il pourra se concentrer directement sur l'étude des *clusters* associés aux *topics* qu'il a «dirigés» sans avoir à explorer la totalité des *clusters* (ce qui n'empêche en aucun cas l'exploration de ces derniers). De plus, dans les expérimentations effectuées dans ce travail, les *clusters* associés à des *topics* dirigés possèdent une plus grande valeur de pureté globale que ceux de l'ensemble total des *clusters* créés (67 % VS 40 %), ce qui est une bonne chose si l'utilisateur ne se concentre que sur ces derniers.

Afin de comparer les quinze *clusters* dirigés obtenus à l'aide de TISK-LDA avec les résultats obtenus par LDA, un humain a été chargé de faire le travail suivant : pour chacun des quinze groupes de mots présentés dans le Tableau C.1, il lui a été demandé d'associer le *topic* qui lui semblait le plus cohérent par rapport au groupe de mots en question parmi les 50 *topics* produits par LDA (*topics* présentés dans le Tableau 3.11). Ce faisant chacun des quinze *clusters* issus des *topics* dirigés de TISK-LDA pourra être associé et comparé à un *cluster* produit à l'aide des *topics* de LDA.

Cette comparaison TISK-LDA VS LDA est présentée dans le Tableau 3.22. Ce dernier présente quinze groupes de deux lignes associés aux quinze groupes de mots du Tableau C.1. La première ligne de chaque groupe correspond à un *cluster* obtenu par TISK-LDA et la deuxième correspond au *cluster* de LDA qui lui a été associé par le travail décrit dans le paragraphe précédent (nom de *cluster* écrit en italique). L'étiquette majoritaire [4] de chaque *cluster* du tableau est également présentée avec les valeurs de précision et de rappel qui lui sont associées. Les colonnes nommées «Var.» qui sont situés à droite de la colonne «Précision» et de la colonne «Rappel»

4. Une seule exception : pour les *clusters* *review-hotel-travel-room* et *review-hotel-restaur-travel* l'étiquette *Hotel* n'est pas l'étiquette majoritaire, mais elle est cependant l'étiquette qui est associée au groupe de mots du Tableau C.1 auquel ces deux *clusters* sont associés.

présentent la variation entre les valeurs de précision et de rappel pour chaque pair de *clusters* présentés.

Le Tableau 3.22 permet de faire plusieurs observations sur les avantages et inconvénients qu'apporte l'utilisation de TISK-LDA par rapport à LDA pour les expérimentations présentées dans ce document. Tout d'abord, en se basant sur les résultats de ce tableau, l'utilisation du TISK-LDA ne semble pas apporter de manière systématique un gain en précision. En effet, huit des quinze *clusters* dirigés possèdent une valeur de pureté d'au moins 7 % inférieure à celle du *cluster* obtenu à l'aide de LDA qui leur est associé. Cependant, pour certains de ces *clusters* cette plus faible valeur de précision se voit accompagnée d'une plus grande valeur de rappel. En particulier, les *clusters* team-game-player-sport, book-publish-stori-author et restaur-food-drink-bar possèdent des valeurs de précision de respectivement 13 %, 15 % et 12 % inférieures à celles des *clusters* obtenus à l'aide de LDA qui leur sont associés, mais possèdent par contre des valeurs de rappel de respectivement 17 %, 18 % et 25 % supérieures à celles de ces mêmes *clusters* associés. Cette baisse de précision n'est cependant pas non plus systématique. En effet, pour trois des quinze *clusters* obtenus à l'aide de TISK-LDA, il est possible d'observer une valeur de précision supérieure d'au moins 11 % comparativement aux *clusters* de LDA qui leur sont associés.

Pour ce qui est des valeurs de rappel des quinze *clusters* reliés à TISK-LDA, elles semblent dans la majorité des cas (13/15) être égales ou supérieures aux valeurs de rappel des *clusters* reliés à LDA. Dans les deux cas où les valeurs de rappel des *clusters* reliés à TISK-LDA sont inférieures à celles des *clusters* reliés à LDA, les valeurs de précision sont quant à elles supérieures.

Ces résultats ne permettent donc pas de conclure que l'utilisation de TISK-LDA mènera systématiquement à des *clusters* avec des valeurs de précision ou de rappel supérieures ou inférieures à celles de *clusters* qui seraient obtenus avec LDA. L'utilisation de TISK-LDA permet cependant, comme mentionnée plutôt dans ce document, de cibler automatiquement les *clusters* d'intérêts obtenus en sortie du processus de classification de requêtes et donc, dans certains contextes, cet avantage pourrait être

Nom du *cluster*	Étiquette	Précision	Var.	Rappel	Var.
sex-video-girl-fuck (6)	*Adult*	0,91	-0,02	0,54	+0,06
sex-video-fuck-view	*Adult*	0,93		0,48	
food-recip-cook-fruit (2)	*Food*	0,85	+0,11	0,58	-0,11
food-recip-cook-fruit	*Food*	0,74		0,69	
dog-anim-speci-cat (9)	*Animal*	0,84	+0,02	0,66	+0,03
dog-speci-anim-cat	*Animal*	0,82		0,63	
car-engin-vehicl-model (13)	*Car*	0,80	-0,07	0,62	-
car-engin-model-vehicl	*Car*	0,87		0,62	
team-game-player-sport (7)	*Sport*	0,75	-0,13	0,54	+0,17
game-team-season-leagu	*Sport*	0,88		0,37	
god-church-christian-cathol (15)	*Religion*	0,73	+0,12	0,70	-0,09
god-church-christian-king	*Religion*	0,61		0,79	
film-movi-director-share (4)	*Movie*	0,70	-0,09	0,37	+0,01
movi-tv-review-list	*Movie*	0,79		0,36	
music-song-album-record (10)	*Music*	0,69	-0,07	0,60	+0,05
song-music-album-record	*Music*	0,76		0,55	
health-medic-effect-diseas (8)	*Medical*	0,65	+0,14	0,40	+0,11
health-medic-care-patient	*Medical*	0,51		0,29	
game-play-player-video (14)	*VideoGame*	0,63	-0,11	0,68	+0,02
game-player-level-play	*VideoGame*	0,74		0,66	
book-publish-stori-author (11)	*Book*	0,52	-0,15	0,55	+0,18
book-stori-publish-chapter	*Book*	0,67		0,37	
show-episod-seri-tv (5)	*Tv*	0,46	-0,01	0,44	+0,06
film-show-seri-episod	*Tv*	0,47		0,38	
restaur-food-drink-bar (1)	*Restaurant*	0,42	-0,12	0,56	+0,25
review-hotel-restaur-travel	*Restaurant*	0,54		0,31	
govern-presid-state-nation (12)	*Politics*	0,30	-0,07	0,64	-
presid-state-govern-nation	*Politics*	0,37		0,64	
review-hotel-travel-room (3)	*Hotel**	0,05	+0,02	0,17	-
review-hotel-restaur-travel	*Hotel**	0,03		0,17	

Tableau 3.22 – Comparaison entre les quinze *clusters* dirigés obtenus à l'aide de l'algorithme TISK-LDA et quinze *clusters* similaires obtenus à l'aide de l'algorithme LDA.

une bonne raison d'opter pour une telle méthode. Ceci étant dit, il sera intéressant dans le future de tester de nouvelles approches pour tenter de diriger sémantiquement les *clusters* obtenus par un processus de classification de requêtes de recherche web.

3.4 Détection d'entités nommées

Les résultats obtenus en appliquant les deux processus de découverte d'entités nommées décrits à la section 2.6 sur l'ensemble de requêtes étudié seront présentés dans cette section. La section 3.4.1 présentera les résultats obtenus en utilisant la technique se basant sur les balises de type *microdata* de Shema.org contenues dans le code source de la page web et la section 3.4.2 présentera ceux obtenus avec la méthode faisant usage du texte entier de la page web.

Il est à noter que tout au long de cette section, le terme Entité Nommée Solitaire (ENS) fera référence à une entité nommée qui constitue à elle seule la requête qui la contient. Par exemple, la requête «Joy Division» n'est constituée que de l'entité nommée «Joy Division» et donc cette entité nommée sera qualifiée de ENS.

3.4.1 Méthode basée sur Schema.org

Il est tout d'abord important de mentionner que seulement 2 738 pages web sur l'ensemble des 25 217 considérées contiennent des balises de type *microdata* utilisant la taxonomie proposée par Schema.org. Ces 2 738 page web contiennent un total de 3 584 mots ou groupes de mot identifiés par de telles balises. Comme mentionné dans la section 2.6.2, le fait que les balises considérées ici contiennent un «type» donne la possibilité à l'utilisateur de choisir quels «types» sont d'intérêt pour lui. Dans cette optique, la distribution des «types» des 3 584 balises est présentée dans le Tableau 3.23.

Suite à l'observation de cette table, il est possible de remarquer que les balises du type «VideoObject» sont très nombreuses dans l'ensemble des balises considérées (57,59 %). Ces balises proviennent majoritairement des pages issues du site youtube.com (99,3 %) et sont utilisées pour identifier le titre du vidéo présenté dans la

Type	Fréquence	Pourcentage	Cumulatif
VideoObject	2064	57,59%	57,59%
Person	500	13,95%	71,54%
Movie	220	6,14%	77,68%
Product	155	4,32%	82,00%
LocalBusiness	98	2,73%	84,74%
Organization	77	2,15%	86,89%
Restaurant	70	1,95%	88,84%
Article	49	1,37%	90,21%
YoutubeChannelV2	47	1,31%	91,52%
BlogPosting	46	1,28%	92,80%
WebPage	36	1,00%	93,81%
Hotel	35	0,98%	94,78%
TVEpisode	31	0,86%	95,65%
Event	25	0,70%	96,34%
Review	13	0,36%	96,71%
TVSeries	13	0,36%	97,07%
website	12	0,33%	97,41%
Recipe	12	0,33%	97,74%
Blog	8	0,22%	97,96%
Place	8	0,22%	98,19%
SoftwareApplication	7	0,20%	98,38%
Store	5	0,14%	98,52%
TheaterEvent	5	0,14%	98,66%
MobileSoftwareApplication	4	0,11%	98,77%
SportingGoodsStore	4	0,11%	98,88%
EducationalOrganization	3	0,08%	98,97%
NewsArticle	3	0,08%	99,05%
MusicAlbum	3	0,08%	99,14%
MusicRecording	2	0,06%	99,19%
ItemPage	2	0,06%	99,25%
HousePainter	2	0,06%	99,30%
WebApplication	2	0,06%	99,36%
VideoGallery	1	0,03%	99,39%
...	...	...	...

Tableau 3.23 – Distribution du «type» des balises de type *microdata* utilisant la taxonomie de Schema.org pour les 3 584 balises contenues dans les pages web étudiées

page en question. Or, le titre d'un vidéo de Youtube peut contenir plusieurs entités nommées (p. ex. sexy lady by mc magic), aucune entité nommée ou encore une entité nommée entourée d'un certain contexte (e.g. how to assemble a wheel chair). Pour cette raison, les paires (ENP, NIM) où le NIM est du type «VideoObject» ne correspondront pas souvent à une réelle entité nommée.

En utilisant un seuil minimum de similarité de $\delta = 0,2$, la méthode ici considérée a associé 509 des 3 584 NIM récoltés avec une ENP contenu dans une requête. La distribution du type de ces 509 balises est présentée dans le Tableau 3.24. Parmi les types les plus fréquents présentés dans ce tableau, un système semblable à un MRI pourrait, par exemple, être intéressé à identifier les types *Restaurant*, *Product*, *LocalBusiness*, *TVSeries* et *Movie*. 143 des 509 entités nommées découvertes par cette méthode possèdent l'un de ces types. Après vérification, 142 de ces 143 entité nommées ont été identifiées comme étant de réelles entités nommées, menant à une précision estimée de 99,3 % pour l'ensemble de ces cinq types (la seule erreur qui a été découverte est l'identification de «ryan ransdell» comme étant un film alors qu'il s'agit d'un acteur). Il peut également être intéressant de noter que 62 des 143 (43 %) entités nommées qui ont été découvertes sont des ENSs et donc n'auraient pas pu être découvertes par les systèmes semi-supervisées proposés par Pasca [20] et Xu et al. [31].

Afin de donner quelques exemples du genre d'entités nommées découvertes par cette technique, les Tableaux 3.25, 3.26, 3.27, 3.28 et 3.29 présentent quelques paires (ENP, NIM) sélectionnées aléatoirement parmi les 143 entités nommées dont il a été question précédemment. Chaque tableau est associé à un type parmi les types *Restaurant*, *Product*, *LocalBusiness*, *TVSeries* et *Movie* et chaque ENP présentée dans les tables est entourée du contexte (maximum de deux mots avant et deux mots après) dans lequel elle est située dans la requête. La présence du symbole «∼» dans la colonne de gauche indique que l'entité nommée découverte dans la requête n'est pas écrite de la même façon que le texte, provenant de la page web, qui lui est associé et indique donc les entités nommées pour lesquelles le système permet de détecter une erreur qui a été commise pas le système de reconnaissance vocale.

Type	Fréquence	Pourcentage	Cumulatif
VideoObject	286	56,19%	56,19%
Movie	87	17,09%	73,28%
YoutubeChannelV2	26	5,11%	78,39%
Restaurant	25	4,91%	83,30%
Person	18	3,54%	86,84%
LocalBusiness	13	2,55%	89,39%
Product	13	2,55%	91,94%
Recipe	7	1,38%	93,32%
TVSeries	5	0,98%	94,30%
Article	4	0,79%	95,09%
WebPage	4	0,79%	95,87%
Place	4	0,79%	96,66%
BlogPosting	3	0,59%	97,25%
SoftwareApplication	2	0,39%	97,64%
MobileSoftwareApplication	2	0,39%	98,04%
VideoGallery	1	0,20%	98,23%
Brand	1	0,20%	98,43%
Dentist	1	0,20%	98,62%
MusicRecording	1	0,20%	98,82%
HousePainter	1	0,20%	99,02%
Organization	1	0,20%	99,21%
WebApplication	1	0,20%	99,41%
ApartmentComplex	1	0,20%	99,61%
MusicGroup	1	0,20%	99,80%
AutoDealer	1	0,20%	100,00%

Tableau 3.24 – Distribution du «type» des balises de type *microdata* utilisant la taxonomie de Schema.org pour les 509 balises contenues dans les pages web étudiées correspondant à une Entité Nommée Potentielle

	Entité nommée de la requête (avec contexte)	Entité nommée de la page web
	blue ginger	**blue ginger**
	philly hoagie house	**philly hoagie house**
	mexico viejo *fayetteville arkansas*	**mexico viejo**
∼	**bra three brightwell** *easton maryland*	**brasserie brightwell**
	houston s *boca raton*	**houston s**
∼	**nero s gyros**	**niro s gyros**
	number for **the vine** *in lubbock*	**the vine**
	phoenix asian diner	**phoenix asian diner**
∼	*where is* **pancakes and things** *located*	**pancakes n things**
	the black sparrow *menu*	**the black sparrow**
∼	**miss julie s kitchen** *akron ohio*	**ms julie s kitchen**
	is the **glass onion** *in yarmouth*	**glass onion**
	sauce *at norterra*	**sauce**
∼	**texas to brazil** *baton rouge*	**texas de brazil**
	hong kong taste	**hong kong taste**

Tableau 3.25 – Nom d'Item MicroData provenant de balises Schema.org de type Restaurant correspondant à une Entité Nommée Potentielle

Balises VideoObject Le «type» de balises de type *microdata* le plus fréquent dans les 509 entités nommées découvertes par cette méthode est *VideoObject*. Le Tableau 3.30 présente un échantillon de quinze paires (ENP, NIM) trouvées par la technique dont il est présentement question et pour lesquelles la balise entourant le NIM est de type *VideoObject*. Cette table illustre bien le fait mentionné précédemment, spécifiant que les NIMs identifiés par des balises de type *VideoObject* correspondent rarement à des entités nommées.

	Entité nommée de la requête (avec contexte)	Entité nommée de la page web
	towing capacity **two thousand seven honda pilot** *year*	**two thousand seven honda pilot**
	two thousand eleven toyota matrix	**two thousand eleven toyota matrix**
	two thousand eight saturn aura *exr*	**two thousand eight saturn aura**
	two thousand seven dodge caravan	**two thousand seven dodge caravan**
	two thousand eleven honda odyssey	**two thousand eleven honda odyssey**
	two thousand five nissan altima	**two thousand five nissan altima**
	picture **two thousand five jeep liberty** *frame*	**two thousand five jeep liberty**
∼	**captain america boxer briefs**	**captain america symbol boxer briefs**
	two thousand dodge neon	**two thousand dodge neon**
	two thousand four toyota tacoma	**two thousand four toyota tacoma**
∼	*can a* **two thousand and eight honda accord** *tell*	**two thousand eight honda accord**
	two thousand eight volkswagen jetta *review*	**two thousand eight volkswagen jetta**
	two thousand nine nissan titan *reviews*	**two thousand nine nissan titan**

Tableau 3.26 – Noms d'Item MicroData provenant de balises Schema.org de type Product correspondant à une Entité Nommée Potentielle

	Entité nommée de la requête (avec contexte)	Entité nommée de la page web
∼	**tony s fabrics**	**tony s fabric**
	lakeside feed	**lakeside feed**
	signs now columbia *phone number*	**signs now columbia**
	del rey **chase**	**chase**
	drive to **loxley s**	**loxley s**
	cajun quick mart	**cajun quick mart**
	julie s hair design	**julie s hair design**
	nameless valley ranch	**nameless valley ranch**
∼	**tangs wok**	**tang s wok**
	world of wireless	**world of wireless**
∼	**whidden baum chiropractic** *dublin california*	**widenbaum chiropractic**
	michael timothy s *national*	**michael timothy s**
	plymouth nursery	**plymouth nursery**

Tableau 3.27 – Noms d'Item MicroData provenant de balises Schema.org de type LocalBusiness correspondant à une Entité Nommée Potentielle

	Entité nommée de la requête (avec contexte)	Entité nommée de la page web
	hawthorne *episode summary*	**hawthorne**
	captain kangaroo	**captain kangaroo**
∼	*finale for* **americas got talent**	**america s got talent**
	cd usa	**cd usa**
∼	**best thing i ever ate** *them*	**the best thing i ever ate**

Tableau 3.28 – Noms d'Item MicroData provenant de balises Schema.org de type TvSeries correspondant à une Entité Nommée Potentielle

	Entité nommée de la requête (avec contexte)	Entité nommée de la page web
	ramona and beezus	**ramona and beezus**
	burlesque *reviews*	**burlesque**
∼	**rise of the planet of the eight** *online*	**rise of the planet of the apes**
	actors in **role models**	**role models**
	x men first class	**x men first class**
	step brothers *quotes*	**step brothers**
	the movie **taxi**	**taxi**
	stan helsing *review*	**stan helsing**
	shark night three d	**shark night three d**
	all dogs go to heaven *review*	**all dogs go to heaven**
	the movie **congo**	**congo**
∼	**terranova**	**terra nova**
	titanic *the movie*	**titanic**
∼	**momento**	**memento**
	the holiday *movie reviews*	**the holiday**

Tableau 3.29 – Noms d'Item MicroData provenant de balises Schema.org de type Movie correspondant à une Entité Nommée Potentielle

	Entité nommée de la requête (avec contexte)	Entité nommée de la page web
∼	**two kids in a sandbox**	**two kids one sandbox**
	funny falls	**funny falls**
∼	**can dance**	**i can dance**
∼	**little john bass test**	**lil jon bass test**
	penis penis penis penis *penis*	**penis penis penis penis**
	the **poop song**	**poop song**
	i m your father	**i m your father**
	like a gdi *parody*	**like a gdi**
	throw some d s	**throw some d s**
∼	**funny cat s**	**funny cats**
	terry tate office linebacker	**terry tate office linebacker**
	gummy bear song	**gummy bear song**
	be a **billionaire** *halo remix*	**billionaire**
	the mean kitty song	**the mean kitty song**
	a **short trip home**	**short trip home**

Tableau 3.30 – Noms d'Item MicroData provenant de balises Schema.org de type VideoObject correspondant à une Entité Nommée Potentielle

3.4.2 Méthode basée sur le texte entier de la page web

En lançant le processus sur l'ensemble de 25 217 pages web avec leurs requêtes associées, avec un seuil de $\delta = 0,1$, ce dernier présente un ensemble de 29 491 entités nommées découvertes. De ces 29 491 entités nommées, 3 115 sont issues de requêtes pour lesquelles les entités nommées ont été manuellement identifiées, permettant ainsi une certaine évaluation supervisée des résultats. Cette évaluation montre que 969 des 3 115 entités nommées ont été identifiées comme étant de réelles entités nommées par l'annotateur humain, menant ainsi à une estimation de 31 % de précision pour le processus de détection d'entités nommées en question. Cette faible valeur de précision est entre autres due à une difficulté du système à traiter les entités nommées formées d'un seul mot. Le paragraphe suivant présentera les résultats obtenus si le système rejetait systématiquement ce genre d'entités nommées.

Rejeter les entités nommées formées d'un seul mot Il faut tout d'abord remarquer que 17 641 des 29 491 entités nommées découvertes par la méthode dont il est présentement question sont des entités nommées qui sont constituées d'un seul mot. En retranchant ces entités nommées de l'ensemble de 29 491 entités nommées initiales, on obtient un ensemble de 6 850 entités nommées. De ces 6 850, 716 proviennent de requêtes pour lesquelles les entités nommées ont été manuellement identifiées par un humain et 423 de ces 716 entités nommées ont été identifiées comme étant de réelles entités nommées par l'annotateur humain, menant à une précision estimée de 59 %. La technique utilisée semble donc produire de meilleurs résultats en rejetant systématiquement les entités nommées formées d'un seul mot. Cette constatation semble indiquer que la technique proposée n'est pas assez sélective lorsqu'elle considère des ENPs formées d'un mot. Il faudrait donc, dans des travaux futurs, tenter d'ajuster d'une certaine façon la manière de sélectionner les FTIs dans la technique proposée à la section 2.6.3.

Entité Nommée Solitaire De plus, sur l'ensemble de 6 850 entités nommées (étant formées de deux mots ou plus) découvertes par le système, 3 457 peuvent être considérées comme étant des ENS. De ces 3 457 entités nommées, 347 proviennent de requêtes pour lesquelles les entités nommées ont été manuellement identifiées par un

humain et 285 de ces 347 entités nommées ont été identifiées comme étant de réelles entités nommées par ce dernier. Ceci mène donc à une précision estimée de 82 % pour l'identification de ENS de plus de deux mots. Il est important de mentionner que la technique proposée permet d'identifier automatiquement les ENS parmi les entités nommées découvertes.

Un groupe de 25 ENSs a été sélectionné de manière aléatoire à partir des 285/347 entités nommées correctement identifiées. Ces 25 ENS sont présentées dans le Tableau 3.31. Ce tableau inclut également le type qui a été attribué à ces entités nommées par l'annotateur humain. Tout comme les tableaux précédents, la présence du symbole «∼» dans la colonne de gauche indique que l'entité nommée découverte dans la requête n'est pas écrite de la même façon que le texte provenant de la page web qui lui est associée.

Afin de donner également un aperçu du genre d'erreurs commises par cette technique, le Tableau 3.32 présente 15 «fausses» ENSs choisies aléatoirement parmi les 62/347 «fausses» ENSs identifiées par la technique. Dans ce tableau, le symbole «-» apparaît dans la colonne de gauche d'une ligne si le fait que l'entité nommée contenue dans cette ligne ne soit pas une réelle entité nommée est discutable (i.e. le travail de l'annotateur humain qui a manuellement identifié les entités nommées dans les requêtes est discutable).

Note Il est intéressant de noter que si l'on considère également les ENSs ne contenant qu'un seul mot dans le processus d'identification de ENS, le nombre de ENSs découvertes passe de 3 457 à 4 807 et la précision estimée passe de 82 % à 78 %. Ceci semble indiquer que le problème observé avec les entités nommées formées d'un seul mot n'affecte pas autant la précision lorsqu'on ne considère que les ENS.

La méthode de détection non-supervisée d'entités nommées dont il est ici question semble donc être beaucoup plus efficace pour détecter les ENSs. Afin d'améliorer les résultats obtenus pour les entités nommées qui ne sont pas des ENSs, il pourrait être intéressant, dans de futurs travaux, de tester de nouvelles façons, plus restrictives,

	Entité nommée de la requête	Entité nommée de la page web	Vérité
	cheese platter	cheese platter	Food
	power gravel cleaner	power gravel cleaner	AnimalProduct
∼	blow gun	blowgun	Weapon
	kaze v three	kaze v three	TvShow
∼	porn hub	pornhub	AdultURL
	stink bug	stink bug	Animal
∼	selena gamez	selena gomez	Actor
	comma splice	comma splice	GrammarConcept
	tony s cable car	tony s cable car	Restaurant
	ibbotson heating	ibbotson heating	HouseCompany
	kite runner	kite runner	Book
	matrix revisited	matrix revisited	Documentary
∼	groove shark	grooveshark	MusicURL
	jet set men	jet set men	AdultURL
	the compass of pleasure	the compass of pleasure	Book
	lou holtz	lou holtz	SportPerson
	guinea pigs	guinea pigs	Animal
	seventy seven sunset strip	seventy seven sunset strip	TvShow
	white america	white america	Song
	back scratcher	back scratcher	TBDProduct

Tableau 3.31 – Vingt-cinq Entités Nommées Solitaires correctement identifiées sélectionnées de manière aléatoire parmi les 285/347 entités nommées correctement identifiées

	Entité nommée de la requête	Entité nommée de la page web
-	montgomery county spca	montgomery county spca
	mix tapes	mixtapes
	italian flag	italian flag
	b sixteen	b sixteen
	lf o music	lfo music
	formal region	formal region
	countertransference	counter transference
	robotics merit badge	robotics merit badge
-	union south	union south
-	ohio lottery	ohio lottery
-	conner hubbard and company	conner hubbard and company
	how to make a wand	how to make a wand
	funny dog	funny dog
-	the venus project	the venus project
	trans testicles	trans testicles

Tableau 3.32 – Quinze Entités Nommées Solitaires incorrectement identifiées

pour générer des ENPs à partir d'une requête.

Conclusion

Le travail présenté dans ce document porte sur le développement de techniques pour effectuer deux tâches distinctes qui ont toutes deux comme but premier d'extraire, à partir des *logs* d'un MRI, de l'information pouvant être utilisée pour l'amélioration du système de TALN utilisé par le dit MRI. La première tâche dont il a été question est la classification non-supervisée de requêtes de recherche web présentes dans les *logs* d'un MRI, et ce selon leurs domaines. Cette classification permettrait, entre autres, de détecter des domaines de recherche pour lesquels le système actuel du MRI n'est pas entraîné à fournir directement les informations demandées et pour lesquels ils existent un nombre significatifs de requêtes qui ont été soumises au système. La deuxième tâche qui a été considérée est la détection non-supervisée d'entités nommées contenues dans des requêtes de recherche web présentes dans les *logs* d'un MRI. En supposant que cette détection d'entités nommées soit effectuée sur des requêtes pour lesquelles le système de NERC du MRI n'a pas été capable d'identifier les entités nommées qu'elles contiennent, les entités nommées détectées pourraient être utilisées de différentes façons pour améliorer ce système. Par exemple, elles pourraient être utilisées pour améliorer le dictionnaire d'entités nommées utilisé par le système de NERC ou encore pour effectuer l'annotation partielle de nouvelles données d'entraînement pour ce même système.

En ce qui concerne la classification non-supervisée de requêtes de recherche web, la technique qui a été présentée dans ce document fait usage de la connaissance des pages web qui ont été consultées par les émetteurs des requêtes. Ce qui distingue la technique ici présentée est l'utilisation du texte contenu dans les pages web visitées pour effectuer la classification des requêtes. De plus, cette technique attribue

automatiquement à chaque *cluster* un nom représentant la nature des requêtes qu'il contient et pouvant être facilement interprété par un humain. La technique proposée permet également, moyennant une légère intervention humaine, de diriger certains des *clusters* qui seront créés vers des domaines qui ont été déterminés à l'avance par l'utilisateur de l'algorithme. Bien que les *clusters* obtenus par cette technique ne soient pas à 100 % purs, les résultats obtenus sont encourageants et mettent en évidence la capacité de cette technique à regrouper des requêtes qui sont similaires du point de vue sémantique, mais qui ne partagent pas nécessairement de mots en commun ou qui ne sont pas nécessairement syntaxiquement similaires. Cette dernière caractéristique constitue un grand avantage comparativement à des techniques se basant uniquement sur les mots formant les requêtes pour effectuer leurs classifications.

Pour ce qui est de la détection non-supervisée d'entités nommées, deux techniques ont été présentées dans ce travail et tout comme pour la technique de classification non-supervisée présentée, ces deux techniques utilisent le texte des pages web visitées pour accomplir la tâche visée. La première de ces deux techniques fait usage des balises HTML augmentées d'attributs provenant de la spécification HTML nommée *microdata* pouvant se retrouver dans le code source des pages web visitées par les émetteurs des requêtes. Comme les résultats présentés dans ce document l'indiquent, cette technique particulière permet de détecter des entités nommées avec une très grande précision et d'identifier le type de ces dernières. Elle ne détecte cependant que peu d'entre elles. La deuxième technique proposée permet quant à elle de détecter un plus grand nombre d'entités nommées, mais le fait au dépend d'une certaine perte en précision et ne permet pas l'identification de leurs types. Deux des principaux avantages de ces deux techniques sont qu'elles sont complètement non-supervisées et qu'elles permettent la détection de ENSs, chose que les techniques semi-supervisées présentées par Pasca [20] et par Xu et al .[31] ne sont pas capables d'effectuer. Les techniques présentées possèdent également l'avantage de pouvoir détecter des entités nommées n'étant pas correctement orthographiées dans la requête. Dans le cas où les requêtes ont été soumises par le biais d'un système de reconnaissance vocale, cette caractéristique permet de détecter certaines des erreurs commises par ce dernier, ce qui est un avantage qui est particulièrement intéressant.

Conclusion

Dans de futurs travaux axés sur la classification non-supervisée de requêtes de recherche web, il pourrait être intéressant de concevoir une technique permettant d'incorporer des informations supplémentaires également fournies par les *logs* (p. ex. l'identifiant de l'usager ou ses coordonnées géographiques) dans le processus de classification non-supervisée de requêtes présenté ici. Pour ce qui est de la détection d'entités nommées, il serait intéressant de développer une technique pour identifier le type des entités nommées détectées par la deuxième technique de détection présentée dans ce document qui ne permet pas d'effectuer cette identification. Une méthode de classification non-supervisée basée sur le contexte (c.-à-d. les mots) entourant les mention des entités nommées détectées dans les pages web visitées pourrait être une avenue à considérer.

Annexe A

Graphe biparti

Étant donné un ensemble de requêtes de recherche web pour lesquelles le URL de la page web consulté par l'utilisateur est connu, certaines techniques basées sur une représentation des données par un graphe biparti [4][32] ont été développées pour parvenir à regrouper ensemble les requêtes similaires du point de vue de l'information consultée. Un aperçu de la technique élaborée par Beeferman et al. [4] sera présenté dans cette section.

Graphe biparti Étant donnés les couples de requêtes - URLs contenus dans l'ensemble de données étudié, la construction d'un graphe biparti peut être effectuée. Dans ce graphe, les requêtes et les URLs sont représentés par des sommets. Pour chaque couple de requêtes - URL de l'ensemble, une arête reliant la requête à l'URL qui lui est associé est construite. Il est à noter que si un URL ou une requête revient plus d'une fois dans l'ensemble de données, il ou elle sera représenté(e) par un seul et unique sommet. À titre d'exemple la Figure A.1 représente le graphe qui serait construit pour l'ensemble de données contenu dans le Tableau A.1.

Notation Dans la suite de cette section, pour un graphe biparti donné $\mathcal{G}$, l'ensemble de ses sommets requêtes et l'ensemble de ses sommets URLs seront notés respectivement $\mathcal{R}$ et $\mathcal{U}$. De plus, pour n'importe quel sommet s d'un graphe biparti donné, $\mathcal{N}(s)$ représentera l'ensemble contenant les sommets du graphe qui sont voisins (reliés par une arrête) du sommet s. Il est à noter que si le sommet $s \in \mathcal{R}$ (resp. $s \in \mathcal{U}$)

Requêtes	URLs
news of the day	www.cbc.ca/news
cbc headlines today	www.cbc.ca/news
news of the day	ca.news.yahoo.com/
breakfast places in plymouth ma	www.yelp.ca/biz/all-american-diner-plymouth
all american diner	www.yelp.ca/biz/all-american-diner-plymouth
traffic in portland	here.com/traffic/usa/portland-or

Tableau A.1 – Ensemble de données jouets utilisé dans l'exemple de construction d'un graphe biparti pour des données de type (requête - URL).

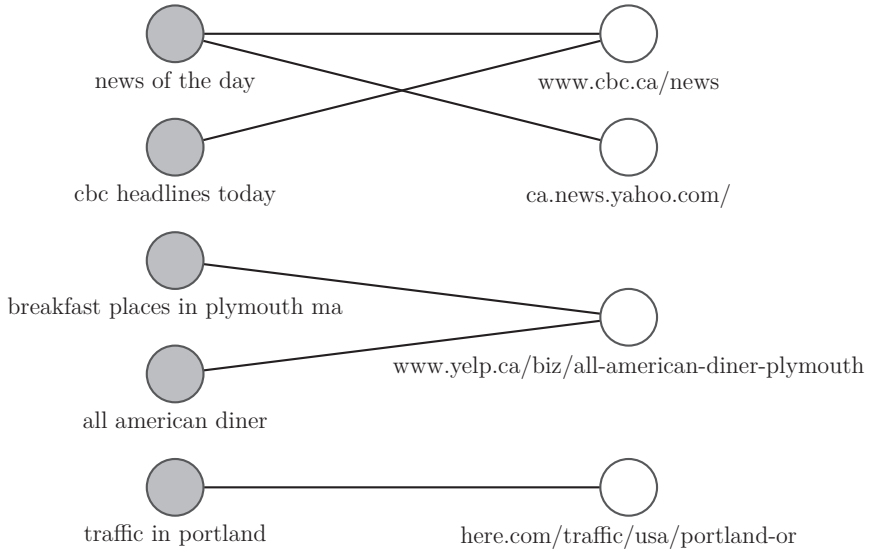


Figure A.1 – Graphe biparti construit avec les données du tableau A.1

alors $\mathcal{N}(s) \subseteq \mathcal{U}$ (resp. $\mathcal{N}(s) \subseteq \mathcal{R}$).

Une fois le graphe construit, ce dernier est utilisé pour définir une certaine mesure de similarité entre deux sommets de $\mathcal{R}$ ou entre deux sommets de $\mathcal{U}$. Cette mesure de similarité est donnée par la fonction (A.1).

$$\sigma(s_1, s_2) = \begin{cases} \frac{\mathcal{N}(s_1) \cap \mathcal{N}(s_2)}{\mathcal{N}(s_1) \cup \mathcal{N}(s_2)} & \text{si } |\mathcal{N}(s_1) \cup \mathcal{N}(s_2)| > 0 \\ 0 & \text{sinon} \end{cases} \tag{A.1}$$

Une fois le graphe biparti construit, l'algorithme 2 est utilisé pour générer un partitionnement des requêtes et un partitionnement des URLs de l'ensemble de données étudié en modifiant de manière itérative le graphe biparti initial.

Algorithme 2 : Algorithme de *clustering* de requêtes - URLs basé sur un graphe biparti

Entrées : Graphe biparti $\mathcal{G}$

Sorties : Graphe biparti $\mathcal{G}$ modifié

tant que *un critère d'arrêt n'est pas rencontré* **faire**

Calculer $\sigma(\cdot, \cdot)$ pour tout les pairs possibles de sommets distincts de $\mathcal{R}$;

Jumeler deux sommets r_i^*, r_j^* tel que $\sigma(r_i^*, r_j^*) = \max_{\substack{r_i, r_j \in \mathcal{R} \\ i \neq j}} \sigma(r_i, r_j)$ et

$\sigma(r_i^*, r_j^*) > 0$;

Calculer $\sigma(\cdot, \cdot)$ pour tout les pairs possibles de sommets distincts de $\mathcal{U}$;

Jumeler deux sommets u_i^*, u_j^* tel que $\sigma(u_i^*, u_j^*) = \max_{\substack{u_i, u_j \in \mathcal{U} \\ i \neq j}} \sigma(u_i, u_j)$ et

$\sigma(u_i^*, u_j^*) > 0$;

fin

Chaque sommet requête (resp. URL) du graphe biparti $\mathcal{G}$ donné en sorti par l'algorithme contient alors plusieurs requêtes (resp. URL). Les sommets requêtes (resp. URL) de ce graphe $\mathcal{G}$ forment ainsi un partitionnement de l'ensemble de requêtes (resp. URLs) étudié.

Critère d'arrêt Beeferman et al. [4] proposent, en spécifiant qu'il s'agit d'un critère qui pourrait être trop permissif, de continuer le processus itératif de l'algorithme jusqu'à ce que le graphe courant ne contienne que des composantes connexes, chacune formée d'un sommet requête relié à un sommet URL. Autrement dit de continuer l'algorithme jusqu'à ce que $\max_{\substack{r_i, r_j \in \mathcal{R} \\ i \neq j}} \sigma(r_i, r_j) = 0$ et $\max_{\substack{u_i, u_j \in \mathcal{U} \\ i \neq j}} \sigma(u_i, u_j) = 0$.

Notons qu'avec le critère d'arrêt défini dans le paragraphe précédent, l'algorithme 2 est équivalent à trouver les composantes connexes du graphe $\mathcal{G}$ et à regrouper ensemble les requêtes ou les URLs faisant parti de la même composante connexe.

Annexe B

Détails des calculs de LDA

B.1 Calcul de $\mathbb{P}(\boldsymbol{z}|\boldsymbol{\alpha}, \eta)$

Note Dans les développements qui suivent, le groupe de variables $z_1, z_2, \ldots z_{l-1}, z_{l+1}, \ldots, z_L$ sera noté $\boldsymbol{z}_{-l}$.

Tout d'abord,

$$\begin{aligned}\mathbb{P}(z_l|\boldsymbol{z}_{-l}, \boldsymbol{w}, \boldsymbol{\alpha}, \eta) &= \frac{\mathbb{P}(\boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\alpha}, \eta)}{\mathbb{P}(\boldsymbol{z}_{-l}, \boldsymbol{w}|\boldsymbol{\alpha}, \eta)} \\ &\propto \mathbb{P}(\boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\alpha}, \eta)\end{aligned}$$

Il faut donc s'attarder au calcul de $\mathbb{P}(\boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\alpha}, \eta)$.

La règle de Bayes donne

$$\mathbb{P}(\boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\alpha}, \eta) = \mathbb{P}(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\alpha}, \eta)\mathbb{P}(\boldsymbol{z}|\boldsymbol{\alpha}, \eta)$$

Il faudra donc calculer $\mathbb{P}(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\alpha}, \eta)$ et $\mathbb{P}(\boldsymbol{z}|\boldsymbol{\alpha}, \eta)$.

Calcul de $\mathbb{P}(\boldsymbol{z}|\boldsymbol{\alpha}, \eta)$ Tout d'abord, notons que $\mathbb{P}(\boldsymbol{z}|\boldsymbol{\alpha}, \eta) = \mathbb{P}(\boldsymbol{z}|\boldsymbol{\alpha})$, ainsi

B.1. Calcul de $\mathbb{P}(\boldsymbol{z}|\boldsymbol{\alpha}, \eta)$

$$
\begin{aligned}
\mathbb{P}(\boldsymbol{z}|\boldsymbol{\alpha}) &= \prod_{d=1}^{M} \mathbb{P}(\boldsymbol{z}_d|\boldsymbol{\alpha}) \\
&= \prod_{d=1}^{M} \int_{\mathcal{S}_K} \mathbb{P}(\boldsymbol{z}_d, \boldsymbol{\theta}_d|\boldsymbol{\alpha}) d\boldsymbol{\theta}_d \\
&= \prod_{d=1}^{M} \int_{\mathcal{S}_K} \mathbb{P}(\boldsymbol{z}_d|\boldsymbol{\theta}_d) \mathbb{P}(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) d\boldsymbol{\theta}_d \\
&= \prod_{d=1}^{M} \int_{\mathcal{S}_K} \left[\prod_{n=1}^{N_d} \mathbb{P}(z_{d,n}|\boldsymbol{\theta}_d) \right] \left[\frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{d,k}^{\alpha_k - 1} \right] d\boldsymbol{\theta}_d \\
&= \prod_{d=1}^{M} \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \int_{\mathcal{S}_K} \left[\prod_{n=1}^{N_d} \theta_{d,z_{d,n}} \right] \left[\prod_{k=1}^{K} \theta_{d,k}^{\alpha_k - 1} \right] d\boldsymbol{\theta}_d \\
&= \prod_{d=1}^{M} \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \int_{\mathcal{S}_K} \left[\prod_{k=1}^{K} \theta_{d,k}^{F_{d,k}} \right] \left[\prod_{k=1}^{K} \theta_{d,k}^{\alpha_k - 1} \right] d\boldsymbol{\theta}_d \\
&= \prod_{d=1}^{M} \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \int_{\mathcal{S}_K} \prod_{k=1}^{K} \theta_{d,k}^{F_{d,k} + \alpha_k - 1} d\boldsymbol{\theta}_d \\
&= \prod_{d=1}^{M} \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \frac{\prod_{k=1}^{K} \Gamma\left(F_{d,k} + \alpha_k\right)}{\Gamma(N_d + \sum_{k=1}^{K} \alpha_k))}
\end{aligned}
$$

F étant une matrice $M \times K$ où l'entrée à la position (d, k) est notée $F_{d,k}$ et contient le nombre de fois le *topic* k à été associé à un mot du document d selon la valeur du vecteur $\boldsymbol{z}$.

Calcul de $\mathbb{P}(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\alpha}, \eta)$ Tout d'abord, il faut noter qu'étant donné $\boldsymbol{z}$, $\boldsymbol{w}$ ne dépend pas de $\boldsymbol{\alpha}$, ainsi $\mathbb{P}(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\alpha}, \eta) = \mathbb{P}(\boldsymbol{w}|\boldsymbol{z}, \eta)$. Pour le restant [1]

$$
\mathbb{P}(\boldsymbol{w}|\boldsymbol{z}, \eta) = \int \mathbb{P}(\boldsymbol{w}, \boldsymbol{\beta}_{1:K}|\boldsymbol{z}, \eta) d\boldsymbol{\beta}_{1:K}
$$

1. Par souci de clarté, $\int_{\mathcal{S}_N} \int_{\mathcal{S}_N} \dots \int_{\mathcal{S}_N} (\dots)\, d\boldsymbol{\beta}_K d\boldsymbol{\beta}_{K-1} \dots d\boldsymbol{\beta}_1$ sera noté $\int (\dots)\, d\boldsymbol{\beta}_{1:K}$.

$$
\begin{aligned}
&= \int \mathbb{P}(\boldsymbol{w}|\boldsymbol{\beta}_{1:K},\boldsymbol{z},\eta)\mathbb{P}(\boldsymbol{\beta}_{1:K}|\boldsymbol{z},\eta)d\boldsymbol{\beta}_{1:K} \\
&= \int \left[\prod_{l=1}^{L}\mathbb{P}(w_l|\boldsymbol{\beta}_{1:K},\boldsymbol{z})\right]\left[\prod_{k=1}^{K}\mathbb{P}(\boldsymbol{\beta}_k|\eta)\right]d\boldsymbol{\beta}_{1:K} \\
&= \int \left[\prod_{l=1}^{L}\beta_{z_l,w_l}\right]\left[\prod_{k=1}^{K}\frac{\Gamma\left(N\eta\right)}{\Gamma(\eta)^N}\prod_{n=1}^{N}\beta_{k,n}^{\eta-1}\right]d\boldsymbol{\beta}_{1:K} \\
&= \int \left[\prod_{k=1}^{K}\prod_{n=1}^{N}\beta_{k,n}^{C_{k,n}}\right]\left[\prod_{k=1}^{K}\frac{\Gamma\left(N\eta\right)}{\Gamma(\eta)^N}\prod_{n=1}^{N}\beta_{k,n}^{\eta-1}\right]d\boldsymbol{\beta}_{1:K} \\
&= \left[\frac{\Gamma\left(N\eta\right)}{\Gamma(\eta)^N}\right]^K\int\prod_{k=1}^{K}\prod_{n=1}^{N}\beta_{k,n}^{C_{k,n}+\eta-1}d\boldsymbol{\beta}_{1:K} \\
&= \left[\frac{\Gamma\left(N\eta\right)}{\Gamma(\eta)^N}\right]^K\prod_{k=1}^{K}\int_{\mathcal{S}_N}\prod_{n=1}^{N}\beta_{k,n}^{C_{k,n}+\eta-1}d\boldsymbol{\beta}_k \\
&= \left[\frac{\Gamma\left(N\eta\right)}{\Gamma(\eta)^N}\right]^K\prod_{k=1}^{K}\frac{\prod_{n=1}^{N}\Gamma\left(C_{k,n}+\eta\right)}{\Gamma(C_k+N\eta))} \\
&= \prod_{k=1}^{K}\frac{\Gamma\left(N\eta\right)}{\Gamma(\eta)^N}\frac{\prod_{n=1}^{N}\Gamma\left(C_{k,n}+\eta\right)}{\Gamma(C_k+N\eta))}
\end{aligned}
$$

C étant une matrice $K \times N$ où l'entrée à la position (k,n) est notée $C_{k,n}$ et contient le nombre de fois pour lequel le mot v_n a été associé au *topic* k selon la valeur des vecteurs $\boldsymbol{w}$ et $\boldsymbol{z}$. De plus, le nombre total de mots de $\boldsymbol{w}$ qui ont été associés au *topic* k sera noté C_k, i.e. $C_k = \sum_{n=1}^{N} C_{k,n}$

Ainsi,

$$
\begin{aligned}
&\mathbb{P}(z_l|\boldsymbol{z}_{-l},\boldsymbol{w},\boldsymbol{\alpha},\eta) \propto \mathbb{P}(\boldsymbol{z},\boldsymbol{w}|\boldsymbol{\alpha},\eta) \\
&\qquad = \prod_{d=1}^{M}\frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\frac{\prod_{k=1}^{K}\Gamma\left(F_{d,k}+\alpha_k\right)}{\Gamma(N_d+\sum_{k=1}^{K}\alpha_k))}\prod_{k=1}^{K}\frac{\Gamma\left(N\eta\right)}{\Gamma(\eta)^N}\frac{\prod_{n=1}^{N}\Gamma\left(C_{k,n}+\eta\right)}{\Gamma(C_k+N\eta))}
\end{aligned}
\tag{B.1}
$$

B.1. Calcul de $\mathbb{P}(\boldsymbol{z}|\boldsymbol{\alpha}, \eta)$

Les développements qui suivent sont tirés d'un rapport technique rédiger par B. Carpenter [10] et serviront à simplifier l'expression (B.1) et ainsi permettre d'effectuer la technique de GS de manière efficace. En enlevant les deux fractions de loi Gamma dont les arguments ne dépendent pas de z_l, l'équation devient

$$\mathbb{P}(z_l|\boldsymbol{z}_{-l}, \boldsymbol{w}, \boldsymbol{\alpha}, \eta) \propto \prod_{d=1}^{M} \frac{\prod_{k=1}^{K} \Gamma\left(F_{d,k} + \alpha_k\right)}{\Gamma(N_d + \sum_{k=1}^{K} \alpha_k))} \prod_{k=1}^{K} \frac{\prod_{n=1}^{N} \Gamma\left(C_{k,n} + \eta\right)}{\Gamma(C_k + N\eta))}$$

Soit d' le document contenant z_l et n' la position de z_l relative à ce document. Utilisant cette notation z_l peut se réécrire comme étant $z_{d',n'}$ et ainsi les termes des matrices F et C qui dépendent de la valeur de z_l sont respectivement $F_{d',k}$ et $C_{k,w_{d',n'}}$ pour $k = 1, 2, \ldots K$. Le prochain développement met en évidence ces termes dépendant de z_l.

$$\begin{aligned}\mathbb{P}(z_l|\boldsymbol{z}_{-l}, \boldsymbol{w}, \boldsymbol{\alpha}, \eta) \propto & \left[\prod_{\substack{d=1 \\ d\neq d'}} \frac{\prod_{k=1}^{K} \Gamma\left(F_{d,k} + \alpha_k\right)}{\Gamma(N_d + \sum_{k=1}^{K} \alpha_k))}\right] \left[\frac{\prod_{k=1}^{K} \Gamma\left(F_{d',k} + \alpha_k\right)}{\Gamma(N_{d'} + \sum_{k=1}^{K} \alpha_k))}\right] \\ & \left[\prod_{k=1}^{K} \prod_{\substack{n=1 \\ n\neq w_{d',n'}}}^{N} \Gamma\left(C_{k,n} + \eta\right)\right] \left[\prod_{k=1}^{K} \frac{\Gamma\left(C_{k,w_{d',n'}} + \eta\right)}{\Gamma(C_k + N\eta))}\right]\end{aligned}$$

En ne considérant que les termes dépendant de z_l, la relation suivante est obtenue

$$\mathbb{P}(z_l|\boldsymbol{z}_{-l}, \boldsymbol{w}, \boldsymbol{\alpha}, \eta) \propto \prod_{k=1}^{K} \Gamma\left(F_{d',k} + \alpha_k\right) \frac{\Gamma\left(C_{k,w_{d',n'}} + \eta\right)}{\Gamma(C_k + N\eta))}$$

Soit $F^{(-d',n')}$ et $C^{(-d',n')}$ des matrices identiques aux matrices F et C à l'exception qu'elles ignorent la valeur prise par z_l. Ainsi,

$$F_{d',z_l} = 1 + F^{(-d',n')}_{d',z_l} \text{ et } C_{z_l,w_{d',n'}} = 1 + C^{(-d',n')}_{z_l,w_{d',n'}}$$

et pour $k \neq z_l$,

$$F_{d',k} = F_{d',k}^{(-d',n')} \text{ et } C_{k,w_{d',n'}} = C_{k,w_{d',n'}}^{(-d',n')}$$

La relation précédente peut donc se réécrire comme

$$\begin{aligned}
\mathbb{P}(z_l|\boldsymbol{z}_{-l},\boldsymbol{w},\boldsymbol{\alpha},\eta) \propto & \left[\prod_{\substack{k=1\\k\neq z_l}}^{K} \Gamma\left(F_{d',k}+\alpha_k\right)\right]\Gamma\left(F_{d',z_l}+\alpha_{z_l}\right) \\
& \left[\prod_{\substack{k=1\\k\neq z_l}}^{K} \frac{\Gamma\left(C_{k,w_{d',n'}}+\eta\right)}{\Gamma(C_k+N\eta))}\right]\frac{\Gamma\left(C_{z_l,w_{d',n'}}+\eta\right)}{\Gamma(C_{z_l}+N\eta))} \\
= & \left[\prod_{\substack{k=1\\k\neq z_l}}^{K} \Gamma\left(F_{d',k}^{(-d',n')}+\alpha_k\right)\right]\Gamma\left(1+F_{d',z_l}^{(-d',n')}+\alpha_{z_l}\right) \\
& \left[\prod_{\substack{k=1\\k\neq z_l}}^{K} \frac{\Gamma\left(C_{k,w_{d',n'}}^{(-d',n')}+\eta\right)}{\Gamma(C_k^{(-d',n')}+N\eta))}\right]\frac{\Gamma\left(1+C_{z_l,w_{d',n'}}^{(-d',n')}+\eta\right)}{\Gamma(1+C_{z_l}^{(-d',n')}+N\eta))}
\end{aligned}$$

Et maintenant, en utilisant le fait que $\Gamma(1+x) = x\Gamma(x)$,

$$\begin{aligned}
\mathbb{P}(z_l|\boldsymbol{z}_{-l},\boldsymbol{w},\boldsymbol{\alpha},\eta) \propto & \left[\prod_{\substack{k=1\\k\neq z_l}}^{K} \Gamma\left(F_{d',k}^{(-d',n')}+\alpha_k\right)\right]\Gamma\left(F_{d',z_l}^{(-d',n')}+\alpha_{z_l}\right)\left(F_{d',z_l}^{(-d',n')}+\alpha_k\right) \\
& \left[\prod_{\substack{k=1\\k\neq z_l}}^{K} \frac{\Gamma\left(C_{k,w_{d',n'}}^{(-d',n')}+\eta\right)}{\Gamma(C_k^{(-d',n')}+N\eta))}\right]\frac{\Gamma\left(C_{z_l,w_{d',n'}}^{(-d',n')}+\eta\right)\left(C_{z_l,w_{d',n'}}^{(-d',n')}+\eta\right)}{\Gamma(C_{z_l}^{(-d',n')}+N\eta)(C_{z_l}^{(-d',n')}+N\eta)} \\
= & \left[\prod_{k=1}^{K} \Gamma\left(F_{d',k}^{(-d',n')}+\alpha_k\right)\right]\left(F_{d',z_l}^{(-d',n')}+\alpha_{z_l}\right) \\
& \left[\prod_{k=1}^{K} \frac{\Gamma\left(C_{k,w_{d',n'}}^{(-d',n')}+\eta\right)}{\Gamma(C_k^{(-d',n')}+N\eta))}\right]\frac{\left(C_{z_l,w_{d',n'}}^{(-d',n')}+\eta\right)}{(C_{z_l}^{(-d',n')}+N\eta)}
\end{aligned}$$

Comme les produits de fonction Γ ne dépendent pas de z_l, la relation suivante est vérifiée

$$\mathbb{P}(z_l|\boldsymbol{z}_{-l},\boldsymbol{w},\boldsymbol{\alpha},\eta) \propto \left(F_{d',z_l}^{(-d',n')}+\alpha_{z_l}\right)\frac{\left(C_{z_l,w_{d',n'}}^{(-d',n')}+\eta\right)}{(C_{z_l}^{(-d',n')}+N\eta)}$$

En se basant sur cette relation, les itérations de GS pourront être effectuées.

B.2 Loi a posteriori des variables $\boldsymbol{\beta}_{1:K}$ et $\boldsymbol{\theta}_{1:M}$

Il sera d'abord question de la loi de probabilité de $\boldsymbol{\theta}_d|\boldsymbol{z}_{1:M},\boldsymbol{\alpha}$. Notons que selon le modèle, étant donné $\boldsymbol{\alpha}$, les variables $\boldsymbol{\theta}_d$ et $\boldsymbol{z}_{d',n}$ sont indépendants à l'exception des $\boldsymbol{z}_{d',n}$ pour lesquels $d'=d$. Ainsi on cherche la loi de probabilité de $\boldsymbol{\theta}_d|\boldsymbol{z}_d,\boldsymbol{\alpha}$.

Par la règle de Bayes,

$$\begin{aligned}
\mathbb{P}(\boldsymbol{\theta}_d|\boldsymbol{z}_d,\boldsymbol{\alpha}) &= \frac{\mathbb{P}(\boldsymbol{z}_d|\boldsymbol{\theta_d})\mathbb{P}(\boldsymbol{\theta}_d|\boldsymbol{\alpha})}{\mathbb{P}(\boldsymbol{z_d}|\boldsymbol{\alpha})}\\
&\propto \mathbb{P}(\boldsymbol{z}_d|\theta_d)\mathbb{P}(\boldsymbol{\theta}_d|\boldsymbol{\alpha})\\
&= \prod_{n=1}^{N_d}\theta_{d,z_{d,n}}\frac{\Gamma\left(\sum_{k=1}^K\alpha_k\right)}{\prod_{k=1}^K\Gamma(\alpha_k)}\prod_{k=1}^K\theta_{d,k}^{\alpha_k-1}\\
&\propto \prod_{n=1}^{N_d}\theta_{d,z_{d,n}}\prod_{k=1}^K\theta_{d,k}^{\alpha_k-1}\\
&= \prod_{k=1}^K\theta_{d,k}^{F_{d,k}}\prod_{k=1}^K\theta_d^{\alpha_k-1}\\
&= \prod_{k=1}^K\theta_{d,k}^{F_{d,k}+\alpha_k-1}
\end{aligned}$$

Ainsi, $\mathbb{P}(\boldsymbol{\theta}_d|\boldsymbol{z}_d,\boldsymbol{\alpha}) \propto \prod_{k=1}^K\theta_{d,k}^{F_{d,k}+\alpha_k-1}$. On reconnaît ici, à une constante de normalisation prêt , une loi de Dirichlet de paramètre $\boldsymbol{\alpha'}=(F_{d,1}+\alpha_1,F_{d,2}+\alpha_2,\dots F_{d,K}+\alpha_K)$. Et donc

$$\boldsymbol{\theta}_d|\boldsymbol{z}_d,\boldsymbol{\alpha} \sim Dir(F_{d,1}+\alpha_1,F_{d,2}+\alpha_2,\dots F_{d,K}+\alpha_K)$$

B.2. Loi a posteriori des variables $\boldsymbol{\beta}_{1:K}$ et $\boldsymbol{\theta}_{1:M}$

Il sera maintenant question de la loi de probabilité de $\boldsymbol{\beta}_k|\boldsymbol{z}_{1:M}, \boldsymbol{w}_{1:M}, \eta$. En notant $\boldsymbol{w}^{(\boldsymbol{k})}$ l'ensemble des variables $w_{d,n}$ pour lesquels $z_{d,n} = k$, selon le modèle et étant donné $\boldsymbol{z}_{1:M}$ et η, $\boldsymbol{\beta}_k$ est indépendant des variables de l'ensemble $\boldsymbol{w}^{(\boldsymbol{k'})}$ pour $k' \neq k$ et dépendant des variables de l'ensemble $\boldsymbol{w}^{(\boldsymbol{k})}$. Ainsi la loi de probabilité cherchée est celle de $\boldsymbol{\beta}_k|\boldsymbol{z}_{1:M}, \boldsymbol{w}^{(\boldsymbol{k})}, \eta$

Par la règle de Bayes,

$$\begin{aligned}
\mathbb{P}(\boldsymbol{\beta}_k|\boldsymbol{z}_{1:M}, \boldsymbol{w}^{(\boldsymbol{k})}, \eta) &\propto \mathbb{P}(\boldsymbol{w}^{(\boldsymbol{k})}|\boldsymbol{\beta}_k)\mathbb{P}(\boldsymbol{\beta}_k|\eta) \\
&\propto \prod_{w \in \boldsymbol{w}^{(\boldsymbol{k})}} \beta_{k,w} \prod_{v=1}^{N} \beta_{k,v}^{\eta-1} \\
&= \prod_{v=1}^{N} \beta_{k,v}^{C_{k,v}} \prod_{v=1}^{N} \beta_{k,v}^{\eta-1} \\
&= \prod_{v=1}^{N} \beta_{k,v}^{C_{k,v}+\eta-1}
\end{aligned}$$

Ainsi, $\mathbb{P}(\boldsymbol{\beta}_k|\boldsymbol{z}_{1:M}, \boldsymbol{w}_{1:M}, \eta) \propto \prod_{v=1}^{N} \beta_{k,v}^{C_{k,v}+\eta-1}$. On reconnaît ici, à une constante de normalisation prêt, une loi de Dirichlet de paramètre $\boldsymbol{\alpha'} = (C_{k,1}+\eta, C_{k,2}+\eta, \dots C_{k,N}+\eta)$. Et donc

$$\boldsymbol{\beta}_k|\boldsymbol{z}_{1:M}, \boldsymbol{w}_{1:M}, \eta \sim Dir(C_{k,1} + \eta, C_{k,2} + \eta, \dots C_{k,N} + \eta)$$

Connaissant maintenant ces deux lois, un candidat possible pour représenter le document d en terme de proportion de *topics* serait

$$\mathrm{E}\left[\boldsymbol{\theta}_{\mathrm{d}}|\boldsymbol{z}_{\mathrm{d}}, \boldsymbol{\alpha}\right] = \left(\frac{F_{d,1} + \alpha_1}{N_d + \sum_{k=1}^{K} \alpha_k}, \frac{F_{d,2} + \alpha_2}{N_d + \sum_{k=1}^{K} \alpha_k}, \dots \frac{F_{d,K} + \alpha_K}{N_d + \sum_{k=1}^{K} \alpha_k} \right)$$

et un candidat possible pour représenter un *topic* en terme de distribution sur les mots du vocabulaire serait

$$\mathrm{E}\left[\boldsymbol{\beta}_{\mathrm{k}}|\boldsymbol{z}_{1:\mathrm{M}}, \boldsymbol{w}_{1:\mathrm{M}}, \eta\right] = \left(\frac{C_{k,1} + \eta}{\sum_{n=1}^{N} C_{k,n} + N\eta}, \frac{C_{k,2} + \eta}{\sum_{n=1}^{N} C_{k,n} + N\eta}, \dots \frac{C_{k,N} + \eta}{\sum_{n=1}^{N} C_{k,n} + N\eta} \right)$$

Annexe C

Mots clés pour TISK-LDA

ID	Domaine	Groupe de mots
1	Restaurant	restaur bar drink dine food chef cuisin meal
2	Food	recip food ingredi cookbook dish meal casserol macaroni_chees slow_cooker recip_ingredient
3	Hotel	hotel room holiday_inn
4	Movie	film movi box_offic screenplay actor director trailer
5	Tv	episod show tv televis full_episod tv_seri cast_member tv_show charact seri season
6	Adult	porn sex fuck nude nake xxx butt ass fetish hardcor tit
7	Sport	team game sport player leagu coach basebal seri major_leagu footbal world_seri stanley_cup ice_hockey
8	Medical	diseas caus drug effect infect treatment health medic patient doctor hospit medicin physician diagnosi
9	Animal	anim speci breed prey mammal pet herbivor carnivor gestat_period
10	Music	music song album tour rock concert pop album_releas record_session music_festiv music_video
11	Book	novel book literatur author publish literari writer narrat short_stori
12	Politics	democrat elect democr republican govern presid conserv senat

Tableau C.1 – Les 15 groupes de mots donnés en entrée à l'algorithme TISK-LDA.

ID	Domaine	Groupe de mots
13	Car	vehicl sedan car engin fuel passeng auto automobil motor wheel automot auto sport_car automat_transmiss manual_transmiss station_wagon hybrid_electr concept_car diesel_engin
14	VideoGame	videogam game playstat xbox nintendo video_game consol_game game_boy
15	Religion	god church christian jesus bibl religion

Tableau C.2 – Suite - Les 15 groupes de mots donnés en entrée à l'algorithme TISK-LDA.

Note : Les mots qui sont joints par un «_», correspondent à un n-gram. Dans ces cas, la présence du n-gram en entier dans un document est nécessaire pour que TISK-LDA modifie la distribution de probabilité des *topics* associés aux mots du n-gram.

Bibliographie

[1] David Andrzejewski.
« *Incorporating Domain Knowledge in Latent Topic Models* ».
Thèse de doctorat, University of Wisconsin-Madison, 2010.

[2] David Andrzejewski et Xiaojin Zhu.
« Latent Dirichlet Allocation with Topic-in-Set Knowledge ».
NAACL 2009 Workshop on Semi-supervised Learning for NLP, pages 43–48, 2009.

[3] David Arthur et Sergei Vassilvitskii.
« k-means ++ : The Advantages of Careful Seeding ».
Dans *Proceedings of the 18th annual ACM-SIAM symposium on discrete algorithms.*, pages 1027–1035, 2007.

[4] Doug Beeferman et Adam Berger.
« Agglomerative clustering of a search engine query log ».
Dans *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 407–416. ACM Press, 2000.

[5] Daniel M Bikel, Scott Miller, Richard Schwartz et Ralph Weischedel.
« Nymble : a High-Performance Learning Name-finder ».
Dans *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 194–201, 1997.

[6] David M. Blei et John D. Lafferty.
« Topic models ».
Dans *Text Mining : Classification, Clustering, and Applications*, pages 71–93. 2009.

[7] David M. Blei, Andrew Y. Ng et Michael I. Jordan.
« Latent Dirichlet Allocation ».
Journal of Machine Learning Research, 3:993–1022, 2003.

[8] Andrew Borthwick, John Sterling, Eugene Agichtein et Ralph Grishman.
« NYU : Description of the MENE Named Entity System as Used in MUC-7 ».
Dans *Proceedings of the 7th Message Understanding Conference (MUC-7)*, 1998.

[9] Andrei Broder.
« A taxonomy of web search ».
Special Interest Group on Information Retrieval Forum, 36(2):3–10, 2002.

[10] Bob Carpenter.
« Integrating Out Multinomial Parameters in Latent Dirichlet Allocation and Naive Bayes for Collapsed Gibbs Sampling ».
Rapport Technique 4, 2010.

[11] George Casella et Edward I. George.
« Explaining the Gibbs Sampler ».
The American Statistician, 46(3):167–174, 1992.

[12] William M. Darling.
« A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling ».
Rapport Technique, 2011.

[13] Qingqing Gan, Josh Attenberg, Alexander Markowetz et Torsten Suel.
« Analysis of Geographic Queries in a Search Engine Log ».
Dans *Proceedings of the First International Workshop on Location and the Web*, pages 49–56. ACM Press, 2008.

[14] Thomas L. Griffiths et Mark Steyvers.
« Finding Scientific Topics ».
Dans *Proceedings of the National Academy of Sciences of the United States of America*, volume 101, pages 5228–5235, avril 2004.

[15] Maryam Kamvar et Shumeet Baluja.
« A Large Scale Study of Wireless Search Behavior : Google Mobile Search ».

Dans *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 701—-709, 2006.

[16] Chuanren Liu, Tianming Hu, Young Ge et Hui Xiong.
« Which Distance Metric is Right : An Evolutionary K-Means View ».
Dans *Proceedings of the SIAM International Conference on Data Mining*, pages 907–918, 2012.

[17] Jean-Michel Marin et Christian P. Robert.
« Les bases de la statistique bayésienne ».
Techniques de l'ingénieur, pages 1–25, 2009.

[18] Andrew Mccallum et Wei Li.
« Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons ».
Dans *Proceedings of the 7th Conference on Natural Language Learning*, pages 188–191, 2003.

[19] Andrew K. McCallum.
« MALLET : A Machine Learning for Language Toolkit ».
http ://mallet.cs.umass.edu, 2002.

[20] Marius Pasca.
« Weakly-Supervised Discovery of Named Entities Using Web Search Queries ».
Dans *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 683–690, 2007.

[21] Daniel E. Rose et Danny Levinson.
« Understanding user goals in web search ».
Dans *Proceedings of the 13th International World Wide Web Conference*, pages 13–19. ACM Press, 2004.

[22] Satoshi Sekine et Chikashi Nobata.
« Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy Chikashi Nobata ».
Dans *Proceedings of the Conference on Language Resources and Evaluation*, pages 1977–1980, 2004.

[23] Shokri Z. Selim et M. A. Ismail.
« K-means-type algorithms : a generalized convergence theorem and characterization of local optimality. ».
IEEE transactions on pattern analysis and machine intelligence, 6(1):81–87, janvier 1984.

[24] Amanda Spink, Dietmar Wolfram, Major B. J. Jansen et Tefko Saracevic.
« Searching the Web : The Public and Their Queries ».
Journal of the American Society for Information Science and Technology, 52(3):226–234, 2001.

[25] Rohini Srihari, Cheng Niu et Wei Li.
« A Hybrid Approach for Named Entity and Sub-Type Tagging ».
Dans *Proceedings of Applied Natural Language Processing Conference*, pages 247–254. Association for Computational Linguistics, 2000.

[26] Pang-Ning Tan, Michael Steinbach et Vipin Kumar.
Introduction to Data Mining, (First Edition).
Addison-Wesley Longman Publishing Co., Inc., 2005.

[27] Yee Whye Teh, David Newman et Max Welling.
« A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation ».
Dans *Proceedings of Neural Information Processing Systems Conference*, pages 1353–1360, 2006.

[28] Hanna M. Wallach.
« *Structured Topic Models for Language* ».
Thèse de doctorat, University of Cambridge, 2008.

[29] Hanna M. Wallach, David Mimno et Andrew Mccallum.
« Rethinking LDA : Why Priors Matter ».
Dans *Proceedings of Neural Information Processing Systems Conference*, 2009.

[30] Ji-Rong Wen et Hong-Jiang Zhang.
« Query Clustering in the Web Context ».
Dans *Clustering and Information Retrieval*, pages 1–30. 2002.

[31] Gu Xu, Shuang-Hong Yang et Hang Li.
« Named Entity Mining from Click-through Data Using Weakly Supervised Latent Dirichlet Allocation ».
Dans *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1365–1374. ACM Press, 2009.

[32] Jeonghee Yi et Farzin Maghoul.
« Query clustering using click-through graph ».
Dans *Proceedings of the 18th International World Wide Web Conference*, pages 1055–1056. ACM Press, 2009.