Système de recommandations vidéo fondé sur l'analyse des sous-titres dans le but de soutenir le développement linguistique d'enfants autistes

par

Simon-Olivier HAREL

MÉMOIRE PAR ARTICLE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE AVEC MÉMOIRE EN GÉNIE LOGICIEL
M. Sc. A.

MONTRÉAL, LE 17 MAI 2023

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

Mme Sylvie Ratté, directrice de mémoire
Département de génie logiciel et des technologies de l'information à l'École de technologie supérieure

M. Laurent Mottron, codirecteur de mémoire
Faculté de médecine à l'Université de Montréal

M. Tony Wong, président du jury
Département de génie des systèmes à l'École de technologie supérieure

M. Luc Duong, membre du jury
Département de génie logiciel et des technologies de l'information à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 20 AVIRL 2023

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Tout d’abord, je tiens à remercier la directrice de ce mémoire, professeure Sylvie Ratté pour votre soutien et votre encadrement tout au long de mon projet de recherche. Votre expertise et votre passion pour votre domaine ont été une source d'inspiration pour moi, et votre mentorat a été inestimable pour mon développement professionnel.

Je souhaiterais exprimer mes sincères remerciements à mon co-directeur, le Professeur Laurent Mottron, pour le temps qu'il m'a consacré, en particulier pour m'avoir initié au domaine de la pédopsychiatrie. Grâce à notre relation, j'ai également pu développer ma capacité à adapter mon discours à un interlocuteur provenant d'un milieu différent du mien.

Je tiens également à exprimer mon plus grand respect à tous ceux qui ont participé à la phase d’évaluation de mon système.

En outre, je suis reconnaissant envers la Fondation Les Petits Trésors, la Chaire Marcel & Roland Gosselin, IBM France et Mitacs pour leur soutien financier.

Enfin, je tiens à remercier ma famille, mes amis, mon chien et ma copine pour leur écoute attentive, pour leur soutien et encouragement tout au long de cette aventure intellectuelle. Leur présence et leur soutien ont été précieux pour moi.

Système de recommandations vidéo fondé sur l'analyse des sous-titres dans le but de soutenir le développement linguistique d'enfants autistes

Simon-Olivier HAREL

RÉSUMÉ

Il a été remarqué que par l'exposition à des vidéos comportant des lettres, des chiffres, des formes, l'enfant autiste d'âge préscolaire et le plus souvent non ou minimalement verbal, atteint rapidement grâce à cette exposition une connaissance autodidacte du code écrit ainsi qu'un vocabulaire qui facilitent sa scolarisation plusieurs années plus tard. Le programme de recherche dans lequel s'inscrit ce projet est d'en arriver à développer un système de recommandations vidéo multimodal et personnalisé qui soutiendra leur progrès linguistique ainsi que de concevoir une base de données sur l'évolution du développement du langage chez ces enfants.

Concrètement, ce projet consiste à développer un système de recommandation fondé sur les sous-titres des vidéos de manière à pouvoir, par la suite, intégrer des caractéristiques extraites d'autres signaux comme les couleurs, les objets, les sons, afin de mieux cerner les centres d'intérêts de l'enfant pour ainsi personnaliser davantage les recommandations. Pour y arriver, nous avons développé un cadriciel d'optimisation comportant une étape d'évaluation approfondie de la stabilité de la paramétrisation du modèle LDA. Nous l'avons ensuite utilisé pour modéliser un corpus constitué de transcriptions automatiques provenant de vidéo pour enfant offert sur YouTube. À l'aide de cette modélisation des sous-titres, nous avons développé un système de recommandations et une interface illustrant l'évolution de l'estimation des centres d'intérêt à travers les thématiques décrivant les sous-titres des vidéos (carte sémantique). Afin de valider la qualité du modèle optimisé, des calculs de recommandations et du potentiel descriptif de la carte sémantique, nous avons conçu une application web afin d'effectuer une première évaluation de ce système à l'aide de participants issus de la communauté universitaire.

À la suite des différentes expérimentations, l'analyse des résultats concernant le cadriciel d'optimisation démontre l'importance de l'analyse en stabilité pour sélectionner une modélisation quasi optimale au lieu de s'appuyer aveuglément sur des procédures dirigées par des métaheuristiques. Concernant l'évaluation du système de recommandations, les résultats démontrent le potentiel d'utiliser LDA comme base de calcul pour concevoir un système de recommandations. En ce qui concerne la carte sémantique, elle a suscité l'intérêt des participants à vouloir l'adopter dans un contexte de recherche et de sélection de vidéos. Finalement, les participants admettent la pertinence d'y visualiser l'évolution de leur centre d'intérêt ainsi que sa particularité à décrire le contenu des vidéos qu'ils ont aimées.

Mots-clés: Système de recommandations vidéo, visualisation des centres d'intérêts, traitements automatique des langues naturelles, optimisation, Latent Dirichlet Allocation

Video recommendation system based on subtitle analysis to support the language development of children with autism

Simon-Olivier HAREL

ABSTRACT

It has been observed that through exposure to videos containing letters, numbers, and shapes, preschool children with autism, who are usually non-verbal or minimally verbal, quickly acquire a self-taught knowledge of the written code and a vocabulary that will facilitate their schooling for several years later. The research program of which this project is part aims to develop a personalized multimodal video recommendation system that supports their linguistic progress and to design a database on language development in these children.

Specifically, this project involves developing a recommendation system based on video subtitles to integrate later features extracted from other signals, such as colors, objects, and sounds, to understand the child's interests better and personalize recommendations further. To achieve this, we developed an optimization framework with a thorough evaluation of the stability of the LDA model parameterization. The framework was then used to model a corpus of automatic transcriptions from children's videos available on YouTube. Using this subtitle modeling, we developed a recommendation system and an interface illustrating the evolution of the estimation of interests through the themes describing the video subtitles (semantic map). To validate the quality of the optimized model, recommendation calculations, and the descriptive potential of the semantic map, we designed a web application to conduct an initial evaluation of this system with participants from the academic community.

Following the different experiments, the analysis of the results concerning the optimization framework demonstrates the importance of stability analysis to select a near-optimal model instead of blindly relying on metaheuristic-driven procedures. Regarding evaluating the recommender system, the results demonstrate the potential of using LDA as a computational basis for designing a recommender system. The participants were interested in adopting the semantic map in a video search and selection context. Finally, the participants appreciated visualizing the evolution of their interests and their relevance in describing the content of the videos they liked.

Keywords: Video recommendation system, interest visualization, automatic natural language processing, linguistic model optimization

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AG	Algorithme Génétique
API	Application Programming Interface
BERTopic	Neural topic modeling with a class-based TF-IDF procedure
BERT	Bidirectional Encoder Representations from Transformers
c-TF-IDF	Class based-Term Frequency-Inverse Document Frequency
DRM	(Digital Right Management
GloVe	Global Vector
TF-IDF	Term Frequency-Inverse Document Frequency
HDBSCAN	Hierarchical Density-Based Spatial
HTML	The HyperText Markup Language
LDA	Latent Dirichlet Allocation
LSA	Lantent Semantic Analysis
LD	Longueur d'un document
NMF	Non-negative Matrix Factorization
pLSA	Probabilictic Latent Semantic Analysis
RIAA	Recording Industry Association of America
TALN	Traitement automatique des langues naturelles
TSA	Trouble du Spectre de l'Autisme
TDA/H	Trouble du Déficit de l'Attention avec/ou sans Hyperactivité
UMAP	Uniform Manifold Approximation and Projection

INTRODUCTION

C'est en septembre 2019 qu'une première ébauche de ce projet de recherche a été développée sous la forme d'une preuve de concept. Il a été proposé comme projet de fin d'études à un groupe d'étudiants du baccalauréat pour en valider sa faisabilité. Son évaluation finale s'est conclue par une proposition de poursuivre ce projet au cycle supérieur à partir de l'été 2020. Ce projet est issu d'une collaboration entre le Dr Laurent Mottron, clinicien chercheur au Département de psychiatrie et d'addictologie de la faculté de médecine de l'Université de Montréal, titulaire de la chaire de recherche en Autisme de l'Université de Montréal, et les professeurs Sylvie Ratté et Roberto Érick Lopez Herrejon du Département de génie logiciel et des technologies de l'information de l'École de technologie supérieure.

Ce projet est également soutenu par un partenaire industriel (IBM France) ainsi que des partenaires non universitaires (la fondation Les Petits Trésors et la Chaire Marcel & Rolande Gosselin). Cette fondation est impliquée, via les décisions de son conseil d'administration, dans le choix de cibles pour des dons philanthropiques majeurs orientés vers la santé mentale des enfants. Elle a ainsi été co-organisatrice, avec les fondations de Sainte Justine et de McGill, de l'utilisation des dons de la fondation Marie-Jeanne Coutu pour le RTSA/TACC (réseau pour la transformation des soins en autisme) qui regroupe l'ensemble des forces vives en recherche sur l'autisme au Québec. Le soutien de la part du cette fondation ajoute au projet une perspective venant du milieu de la santé mentale et confirme que le projet proposé sert les intérêts de la population cible. De plus, une fois le projet complété, un des premiers rôles de la fondation sera de promouvoir la technologie mise sur pied auprès des parents d'enfants autistes et tous les milieux susceptibles de tirer parti de celle-ci.

0.1 Contexte général du programme de recherche

Le programme de recherche dans lequel s'inscrit ce projet vise à apporter une solution innovante et responsable à une problématique sociale définie avec les milieux preneurs.

Concrètement, l'utilisation des écrans, tablettes et cellulaires par les enfants, est une préoccupation croissante en pédopsychiatrie. En effet, les enfants affectés par certaines conditions, tel que le trouble du déficit de l'attention avec/ou sans hyperactivité (TDA/H) tendent à en faire un usage excessif. En revanche, des enfants affectés par d'autres conditions, comme l'autisme, peuvent probablement en tirer des bénéfices (Kientz et al., 2013). Des acteurs du milieu, des chercheurs et intervenants spécialisés en autisme, et les parents des enfants autistes ont fréquemment noté qu'une proportion importante de ceux-ci présente un intérêt marqué pour les tablettes et les téléphones intelligents, même s'il est exclu que cet usage puisse être impliqué dans la survenue de l'autisme (Heffler et al., 2020). Les travaux du groupe de L. Mottron ont montré que ces enfants, le plus souvent non ou minimalement verbaux, s'orientent spontanément vers des applications ou des vidéos comportant des lettres, des chiffres ou des formes (Jacques et al., 2018). De plus, une proportion notable de ces enfants atteint rapidement, grâce à cette exposition (Kissine et al., 2019), une connaissance autodidacte du code écrit et d'un vocabulaire qui facilite leur scolarisation plusieurs années plus tard et leur entrée ultérieure dans un langage oral à fonction communicative.

Parmi les milieux preneurs de ce programme, les milieux où les enfants autistes sont scolarisés (classes TSA) sont encore majoritairement fixés sur les stratégies comportementales de type behavioristes qui n'utilisent pas les forces des enfants. Les milieux de l'éducation sont sollicités pour renouveler leur programme et leur mandat auprès de ces enfants. Ce programme représente donc un changement d'orientation dans la manière de familiariser les enfants avec le code écrit. L'observation des compétences développées par des enfants autistes, qui s'exposent eux-mêmes à des applications d'alphabétisation, suggère en effet d'exploiter leurs forces et leurs intérêts spontanés afin de faciliter l'accès au langage écrit et oral dans la période scolaire. En effet, dans le cadre des techniques d'éducation spécialisée offertes dans les classes pour les enfants atteints du TSA, l'exposition supervisée à des écrans est un moyen qui peut faciliter les acquis préscolaires (p. ex. littératie) des enfants (Lopez-Herrejon et al., 2020).

Cependant, la condition autistique impose toutefois de surmonter certains défis pour mener ce programme à bien :

1. Le refus actif de l'interaction de tutelle propre à l'autisme à ces âges empêche de guider activement les enfants dans leur utilisation des technologies et leur choix des contenus.
2. L'intérêt limité de ces enfants et leur préférence marquée pour des activités connues et prévisibles font qu'ils explorent les applications de manière exhaustive et répétitive.
3. L'exposition spontanée et non supervisée à des applications disponibles sur le Web favorise des applications commerciales ou dépourvues de valeur didactique, au détriment d'environnements d'apprentissage plus variés, enrichissants, et personnalisés pour les enfants autistes.

La solution que nous proposons répondra à deux problématiques inhérentes à l'autisme : les difficultés d'interaction due aux limitations de la communication verbale et la limitation de leurs intérêts.

Mentionnons également lorsque ces enfants sont exposés à des écrans dans un contexte supervisé, ils ne suivent pas les recommandations explicites et n'indiquent pas leurs préférences autrement que par le choix qu'ils font sur les écrans. De plus, leur intérêt pour ce type de matériel a été démontré par une méta-analyse systématique (Ostrolenk et al., 2017) ainsi que leur fascination (surdétection) pour les lettres et les chiffres (Kaldy et al., 2011).

Par ces faits, nous croyons qu'un soutien numérique intelligent et personnalisé permettrait d'exploiter les forces et les intérêts des enfants pour la littératie. De plus, cette technologie pourrait facilement être utilisée dans les foyers et les milieux cliniques pour aider les enfants à acquérir des compétences préscolaires. En outre, les données collectées sur l'évolution de l'apprentissage de leur acquis installeraient chez les milieux preneurs une vision du

développement éducatif des enfants autistes qui intègrent leurs modes d'apprentissage spécifiques.

0.2 Proposition d'une solution

Pour ce faire, nous proposons le développement d'un système de recommandations vidéo pour les enfants autistes. Ce système s'appuiera sur les caractéristiques comme les sous-titres, les couleurs, les objets et le son des vidéos afin de déterminer les centres d'intérêts de l'enfant pour émettre des recommandations. De plus, le contenu des recommandations sera personnalisé, puisqu'il sera fondé sur les préférences et compétences que l'enfant aura démontrées dans ses choix antérieurs lors de l'utilisation de l'application.

0.3 Objectif général du projet de recherche

En guise de première étape pour ce projet d'envergure (le programme de recherche), l'objectif général sera de développer un système de recommandations vidéo fondé sur l'analyse des sous-titres via un modèle de langue. Nous partons de l'hypothèse soutenue (Basu et al., 2016; Mei et al., 2011; Zhu, Shyu et Wang, 2013) que l'utilisation des sous-titres dans un contexte de vidéo offre une source d'information ayant un potentiel descriptif prometteur quant à l'interprétation de la vidéo. Finalement, ce système sera développé de manière à pouvoir, par la suite, intégrer d'autres caractéristiques telles que la détection des objets, l'analyse des couleurs et du son.

Cet objectif général se décline en 5 sous-objectifs :

SO-1 : Extraction des sous-titres des vidéos en ligne

Ce sous-objectif permettra d'analyser et de concevoir le corpus nécessaire à la génération des recommandations vidéo. L'extraction des sous-titres associés à des vidéos mis à la disposition du public sur une plateforme de visionnement en ligne telle que YouTube soulève des questions légales et techniques quant à leur obtention.

SO-2 : Optimisation du modèle de langue Latent Dirichlet Allocation (LDA)

L'interprétation du langage humain par une machine se fait à l'aide des techniques liées aux traitements automatiques des langues naturelles. Afin de voir automatiquement émerger des sujets latents contenus dans un ensemble de textes, nous optons pour l'utilisons de l'algorithme Latent Dirichlet Allocation (LDA) (Blei, Ng et Jordan, 2003). Cet objectif consiste à développer une approche de recherche automatique des paramètres quasi optimaux de LDA et d'en vérifier sa stabilité.

SO-3 : Optimisation d'un modèle de langue sur un corpus constitué de transcriptions automatiques de vidéo pour enfant

Ce sous-objectif consiste principalement à appliquer la méthodologie développée en SO-2 sur le corpus généré en SO-1 afin de produire un modèle de langue qui servira de base pour les différents calculs de génération de recommandations spécialisées à l'utilisateur.

SO-4 : Génération et visualisation des recommandations vidéo et des centres d'intérêt de l'utilisateur

L'émission des recommandations vers les utilisateurs sera possible grâce à un système informatique déployé sur un serveur Web. Le modèle de langue préalablement développé en SO-3 sera intégré au système. De plus, afin d'exposer le processus de recommandation à l'utilisateur, une interface, que nous définirons comme la carte sémantique, sera développée afin présenter l'évolution de son centre d'intérêt au sein des thématiques identifiées par le modèle.

SO-5 : Évaluation du système par des utilisateurs

Une phase de test avec un premier groupe d'utilisateurs neurotypiques sera réalisée afin de recueillir l'appréciation du système global, d'évaluer la pertinence de la carte sémantique et la qualité du modèle de langue.

Les écrits de ce mémoire sont organisés comme suit. Le premier chapitre abordera une revue de littérature scientifique qui permettra de guider le développement de ce projet. Suivra une description de la méthodologie générale qui permettra le développement des différents sous-objectifs. Afin d'expliciter les raisonnements qui permettront l'atteinte de ces différents sous-objectifs (cinq), nous leur accordons leur propre chapitre. Suivra ensuite une analyse des résultats obtenus lors de la phase d'évaluation. Et finalement, nous conclurons ce manuscrit à l'aide d'une discussion générale sur l'ensemble de ce projet, des propositions de travaux futurs et d'une ouverture vers l'applicabilité de ce type de système pour différents domaines, dont l'autisme.

CHAPITRE 1

REVUE DE LITTÉRATURE

Cette revue de littérature a pour objectif d'introduire les concepts directeurs qui permettront de développer ce projet. Pour ce faire, nous aborderons les différentes approches couramment utilisées dans l'élaboration d'un système de recommandations, décrirons les principaux modèles de langues liés à la modélisation par sujets, discuterons de l'évaluation de la performance de ces types de modèles de langue et conclurons sur l'approche que nous allons privilégier pour solutionner notre projet de recherche.

1.1 Système de recommandations

L'idée d'un système de recommandations a émergé assez tôt dans l'histoire de l'informatique et a été initiée à cause du domaine de la recherche d'information. Pensons par exemple au système *Grundy* (Rich, 1979) qui jouait le rôle d'un bibliothécaire en fournissant des suggestions de lecture aux utilisateurs en basant ses recommandations sur des descriptions stéréotypées. Comme l'aurait fait un bibliothécaire en connaissant les intérêts de ses clients et son répertoire de livres, *Grundy* créait les stéréotypes des utilisateurs et des livres via des formulaires et des informations encodées en dur afin de suggérer de nouvelles recommandations. Ensuite, c'est durant les années 1990, à la suite des améliorations des technologies entourant l'internet et de l'apparition du commerce en ligne, que la technique du filtrage collaboratif a commencé à apparaître comme une solution pour faire face à la surcharge d'information. Des systèmes tels que *Tapestry* (Goldberg et al., 1992) et *GroupLens* (Resnick et al., 1994) ont été créés, entraînant l'intérêt de diverses disciplines telles que les interactions homme-machine, l'apprentissage automatique et la recherche d'information pour parfaire les systèmes de recommandations. Ces développements ont permis l'émergence de systèmes de recommandations dans divers domaines, tels que *Ringo* (Shardanand et Maes, 1995) pour la musique, *BellCore Video Recommender* (Hill et al., 1995) pour les vidéos et *Jester* (Goldberg

et al., 2001) pour les blagues et pour tranquillement s'introduire dans le domaine du marketing en ayant pour objectif de faire mousser les ventes de produits. C'est à la fin des années 1990 qu'*Amazon* a lancé son propre système de recommandations qui était basé sur le filtrage collaboratif. Depuis son adoption et par l'avantage concurrentiel que ce type de technologie a su leur procurer, l'utilisation de ce type de technologie est devenue une quasi-nécessité. Étant devenu un élément essentiel et un domaine de recherche indépendant au milieu des années 1990 (Goldberg et al., 1992), cela a donc motivé l'innovation afin d'introduire de nouvelles méthodes de recommandations comme l'approche basée sur le contenu et les approches hybrides qui combinent différentes approches afin de minimiser les désavantages liés à l'utilisation d'une seule approche. L'engouement pour la recherche sur les algorithmes de recommandations a continué de prendre de l'ampleur en 2006 avec le lancement du *Netflix Prize* (Bennett et Lanning, 2007), qui a suscité une grande activité dans les milieux académiques et amateurs. Cette compétition de 1 million de dollars a montré l'importance que les entreprises accordent à la précision des recommandations.

Comme il est possible de le déduire à la suite de ce court survol historique de l'évolution des systèmes de recommandations, leur principal objectif est de fournir des ressources pertinentes à un utilisateur en fonction de ses préférences. Cela a pour avantage de réduire son temps de recherche et de lui offrir des suggestions auxquelles il n'aurait pas spontanément prêté attention. Pour atteindre un tel objectif, ces systèmes font appel à la modélisation de 2 entités de base qui sont l'item et le profil de l'utilisateur. L'item fait référence aux éléments spécifiques contenus dans la base de données du système de recommandations. Quant au profil de l'utilisateur, celui-ci fait référence aux différentes caractéristiques qui permettront de préciser ses préférences pour certains items lors des recommandations. Par exemple, ces caractéristiques peuvent être l'historique des items qu'il a consommés, ses rétroactions faites sur différents items, des caractéristiques qui permettent de décrire le contexte de l'utilisation du système et toutes autres caractéristiques qui permettent de caractériser les comportements de l'utilisateur.

En fonction du niveau d'interaction entre ces 2 entités, il est possible de classifier les systèmes de recommandations sous plusieurs approches où les plus connues peuvent être basées sur : le filtrage collaboratif, le contexte, le contenu, les données démographiques des participants, et finalement, le calcul de l'utilité de chaque élément pour l'utilisateur.

Dans le contexte de cette revue de littérature, nous aborderons les approches suivantes : l'approche par filtrage sur le contenu, l'approche de recommandations basée sur filtrage collaboratif et les approches hybrides. Les prochaines sous-sections détailleront chacune de ces approches. Avant de passer aux approches hybrides, nous discuterons des avantages et inconvénients des deux premiers types d'approches de recommandations.

1.1.1 Approche par filtrage sur le contenu

Un système de recommandations utilisant l'approche par filtrage sur le contenu est dérivé du domaine de la recherche d'information. Cette approche repose sur l'idée que des items ayant des caractéristiques similaires seront également appréciés. Dans le contexte de la recommandation vidéo, les caractéristiques usuellement utilisées sont communément appelées les métadonnées. Les métadonnées sont représentées comme une liste d'éléments descriptifs de la vidéo sous la forme clé-valeur. Par exemple, les métadonnées décrivent les informations telles que le genre (horreur, science-fiction, etc.), son classement (PG-13, R, NC-17, etc.), les noms des acteurs principaux, les noms des réalisateurs et toutes autres informations qui bonifieraient la description d'une vidéo.

L'idée générale du calcul des recommandations de cette approche consiste à évaluer la similarité des caractéristiques entre les items déjà consultés par l'utilisateur (historique) à ceux qui n'ont pas encore été consultés afin de fournir des recommandations personnalisées. Cette approche par filtrage sur le contenu est parfois décrite comme étant un système de recherche d'informations dit personnalisé.

1.1.2 Approche de recommandations basée sur le filtrage collaboratif

Contrairement au filtrage par contenu, qui base son calcul des recommandations sur la similarité entre les caractéristiques des items contenue dans la base de données et l'historique de l'utilisateur, le filtrage collaboratif vise à produire des recommandations en calculant la similarité entre l'utilisateur actuel et l'ensemble des utilisateurs existants. L'objectif n'est plus de recommander de nouveaux items sur la base de son historique, mais plutôt de déterminer quels items ont été appréciés par les utilisateurs ayant des profils similaires à celui de l'utilisateur actuel. En d'autres termes et selon Goldberg *et al.* (2001), cette approche peut être décrite comme étant l'automatisation des processus sociaux. Par exemple, dans notre quotidien, si une personne a besoin d'une information, elle consultera son entourage, car son entourage a probablement déjà trouvé et évalué cette information. Finalement, nous pouvons également mentionner que cette approche est la plus répandue dans l'élaboration d'un système de recommandations.

1.1.3 Avantages et inconvénients des approches basées sur le contenu et le filtrage collaboratif

Les deux approches de recommandations préalablement discutées ont chacune leurs forces et leurs faiblesses que nous résumons ci-dessous (cf. Tableau 1.1).Notez que les avantages et inconvénients ainsi que leur explications présentées ci-après sont tirés de Burke, 2002.

Tableau 1.1 Compromis entre les techniques de recommandations

	Avantage	Inconvénient
Basé sur le contenu	2-3-4	5-8-9
Filtrage collaborative	1-2-3-4	5-6-7-8-9

1. Niches multi-genres (*Cross-genre niches or outside the box*)

   Cette approche réfère à des genres ou à des catégories qui ne sont pas nécessairement liés les uns aux autres, mais qui peuvent être appréciés par les mêmes utilisateurs. Par exemple, un utilisateur qui apprécie la musique rock, mais qui a des voisins similaires du point de vue de ce genre musical et qui aiment également la musique électronique, pourrait se faire recommander de la musique électronique, même si ces genres sont différents.

2. La connaissance du domaine n'est pas nécessaire (*Domain knowledge not needed*)

   Dans ce type d'approche, les systèmes de recommandations peuvent fournir des recommandations précises sans nécessiter une expertise approfondie dans le domaine concerné. Dans ce cas, le processus de recommandation se base uniquement sur les évaluations des items.

3. Adaptation (*Adaptative: quality improves over time*)

   Ici la qualité des recommandations fournies par un système de recommandations s'améliore à mesure que le système recueille davantage de données sur les préférences des utilisateurs. Plus le système collecte des données, plus il est utilisé par les utilisateurs et plus il est capable de détecter les tendances et les relations inhérentes aux données. Les algorithmes de recommandations peuvent ensuite utiliser ces informations pour fournir des recommandations plus précises et personnalisées.

4. Retour d'information implicite suffisant (*Implicit feedback sufficient*)

   Cela signifie que les systèmes de recommandations peuvent fonctionner efficacement en utilisant uniquement les données implicites fournies par les utilisateurs, sans nécessiter de données explicites telles que des évaluations ou des commentaires. Les données implicites sont des informations sur les préférences des utilisateurs qui peuvent être recueillis à partir de leur comportement (actions, achats, recherches, utilisations répétées, etc.). L'avantage des données implicites est qu'elles peuvent être recueillies facilement et sans effort, ce qui peut améliorer l'adoption et l'utilisation du système. Cependant, il est important de noter

que les données implicites peuvent être moins précises que les données explicites, car elles ne reflètent pas nécessairement les préférences réelles des utilisateurs et peuvent être influencées par des facteurs tels que des erreurs de mesure ou des biais.

5. Problème de démarrage à froid : cas du nouvel utilisateur (*New user ramp-up problem*)
   Ce problème concerne la pertinence des recommandations au moment où un nouvel utilisateur utilise le système. En effet, étant donné que cet utilisateur n'a pas encore fourni suffisamment d'information sur ses préférences et ses goûts, il est difficile pour le système de recommandations de personnaliser les recommandations en fonction de ses intérêts spécifiques.

6. Problème de démarrage à froid : cas du nouvel item (*New item ramp-up problem*)
   Ce problème concerne la difficulté de recommander un nouvel item aux utilisateurs lorsque celui-ci n'a pas encore reçu suffisamment d'évaluation de la part des autres utilisateurs pour que cet item soit considéré dans le processus de recommandation.

7. Problème du mouton gris (*"Gray sheep" problem*)
   Ce problème concerne les utilisateurs qui ont des intérêts qui varient de la norme (atypiques) décrivant les autres utilisateurs. C'est-à-dire que ces utilisateurs ont des goûts inhabituels ou des opinions divergentes qui se traduisent par de faibles corrélations avec les autres utilisateurs. Dans ce cas, le système de recommandations aura de la difficulté à recommander des items pertinents.

8. La qualité dépend d'un vaste ensemble de données historiques (*Quality dependent on large historical data set*)
   Cela fait référence au fait que plus le système dispose d'un vaste historique de données (les interactions passées des utilisateurs avec le système) à analyser, meilleur il sera à estimer les préférences des utilisateurs et donc, à fournir des recommandations plus pertinentes et plus personnalisées.

9. Le problème de la stabilité versus la flexibilité (*Stability vs plasticity problem*)

Ce problème peut être vu comme l'inverse du problème de démarrage à froid. D'une part, le système doit être suffisamment stable pour fournir des recommandations cohérentes dans le temps afin de garantir des items pertinents aux utilisateurs. D'autre part, le système doit être assez flexible pour s'adapter aux changements de préférences et des tendances des utilisateurs afin de garantir que les recommandations restent pertinentes et à jour. Par exemple, une fois que le profil des préférences d'un utilisateur a été établi dans un système de recommandations de restaurants, il lui sera difficile de changer ses préférences. À titre d'illustration, un mangeur de steak qui devient végétarien continuera à recevoir des recommandations de Steakhouse de la part d'un système de recommandations pendant un certain temps, du moins, jusqu'à ce que de nouvelles évaluations aient suffisamment d'influence pour que le système commence à lui recommander des restaurants végétariens.

1.1.4 Approche de recommandations hybrides

Les systèmes de recommandations hybrides sont définis comme étant une combinaison de différentes approches dans le but de pallier leurs limitations. Dans la littérature actuelle, ce sont ces approches qui sont les plus présentées, car elles démontrent une plus grande efficacité. Généralement, ce type de système est constitué de deux étapes. La première étape consiste à filtrer les items, par exemple à l'aide de méthodes collaboratives ou par filtrage sur le contenu. La seconde étape, consiste à combiner les items aux profils des utilisateurs à l'aide de méthodes d'hybridation telles que celles présentées aux tableaux suivants tirés de Burke, 2002.

Tableau 1.2 Liste de méthodes d'hybridation

Méthode	**Description**
Pondérée (*Weighted*)	Un système de recommandation hybride pondéré est un système dans lequel le score d'un élément recommandé est calculé à partir des résultats de toutes les techniques de recommandations disponibles dans le système.
Changeante (*Switching*)	Le système passe d'une technique de recommandations à l'autre en fonction de la situation.
Mixte (*Mixed*)	Les recommandations de différentes techniques sont présentées en même temps.
Combinaison de caractéristiques (*Feature combination*)	Les caractéristiques de différentes sources de données sont regroupées dans un unique algorithme de recommandations.
Cascade (*Cascade*)	Une technique de recommandations est d'abord employée pour produire un premier classement grossier d'items à recommander et une seconde technique de recommandations est ensuite appliquée afin d'affiner les recommandations finales.

Tableau 1.3 Liste de méthodes d'hybridation (suite)

Méthode	Description
Augmentation des caractéristiques (*Feature augmentatio*n)	Une première technique est utilisée afin de produire une caractéristique (évaluation, classification) d'un item et cette caractéristique est ensuite incorporée dans le processus de recommandation.
Métaniveau (*Meta-level*)	Cette méthode combine deux techniques de recommandations. Elle consiste à utiliser le modèle généré par l'une des techniques comme une valeur d'entrée pour la seconde technique de recommandations.

1.2 Modèle de langue et choix des paramètres

Dans le domaine de l'analyse de texte, les algorithmes les plus courants pour extraire les sujets principaux d'un corpus de texte appartiennent à la famille de la modélisation par sujet. À l'aide de ces modèles de langue, il est possible d'identifier des sujets latents pour ensuite les filtrer (approche de filtrage sur le contenu) afin d'émettre des recommandations.

Les sous-sections suivantes aborderont d'abord la distinction entre le TALN classique et le TALN moderne. Ensuite, l'importance des choix des paramètres d'un modèle de langue et une comparaison entre LDA (Blei, Ng et Jordan, 2003) et BERTopic (Grootendorst, 2022) seront discutées.

1.2.1 TALN classique et TALN moderne

Depuis l'avènement des techniques de plongement de mots dans la littérature, une distinction a été établie pour différencier les techniques antérieures et postérieures à cette innovation. Les

anciennes techniques sont qualifiées de « classiques » ou « traditionnelles », tandis que les nouvelles sont qualifiées de « modernes ». Le TALN classique utilise des approches linguistiques basées sur des règles et des modèles statistiques pour comprendre et générer du langage naturel à partir d'un corpus. Le TALN moderne, quant à lui, applique des techniques de l'apprentissage profond sur des quantités massives de données textuelles pour entraîner un modèle.

Bien que ces nouvelles avancées soient largement médiatisées (*ChatGPT*), ces techniques présentent plusieurs inconvénients, notamment :

- Ce sont des modèles complexes qui nécessitent l'ajustement de millions de paramètres, rendant leur entraînement et leur utilisation coûteux en termes de ressources informatiques, de temps et d'argent.
- Pour que ces modèles produisent des résultats optimaux, les données d'entraînement et les données qu'ils auront à traiter doivent être similaires des points de vue du contexte et du contenu. Par exemple, un corpus d'entrainement basé sur Wikipédia est plus similaire à des données provenant de source journalistique qu'à des corpus médicaux.
- Ces modèles manquent de transparence en raison de leur complexité, ce qui rend difficile de comprendre leur fonctionnement et les raisons pour lesquelles ils prennent certaines décisions.
- Certains de ces modèles ont soulevé des préoccupations éthiques, notamment en matière de confidentialité des données. Par exemple, la provenance des données utilisées pour leur entrainement (une grande partie du web), a causé des préjudices quant au respect du droit d'auteur et de la confidentialité de l'information. De plus, parce qu'ils sont entrainés sur cette quantité innombrable de textes, ils peuvent générer des résultats comportant des biais de stéréotypes de genre, de racisme et autres.

Néanmoins, cela ne veut pas dire qu'il faut ignorer les techniques modernes, car des approches fondées sur les *Transformers* telles que BERTopic ont justement été créées pour élargir les choix de techniques liées à la modélisation par sujet.

1.2.2 Importance des paramètres d'un modèle de langue

Les algorithmes les plus couramment utilisés pour la modélisation par sujet sont l'analyse sémantique latente (LSA), l'analyse sémantique latente probabiliste (pLSA), l'allocation Dirichlet latente (LDA) et la factorisation matricielle non négative (NMF). Cependant, la principale difficulté lors de l'utilisation de ces différents algorithmes réside dans l'ajustement de leurs paramètres, car ce sont des algorithmes d'apprentissage machine non supervisé.

Dans le cadre du programme de recherche dans lequel s'inscrit ce projet, nous avons analysé les 4 modèles mentionnés précédemment. Pour illustrer les difficultés d'ajustement des paramètres, nous discuterons de LDA dans les paragraphes suivants puisque c'est la technique la plus documentée dans la littérature actuelle. De plus, par soucis de concisions, nous référons le lecteur à l'article de Blei *et al.* (2003) ou toutes autres sources sur le web pour de détails concernant LDA.

L'algorithme LDA est une technique de modélisation des sujets proposée par Blei *et al*. (2003). Cet algorithme est un modèle génératif et probabiliste qui extrait des sujets latents (listes de mots) décrivant le corpus (un ensemble de documents). Cependant, cet algorithme est livré avec un certain nombre de paramètres que l'on doit fixer avant de passer à son entraînement. Par exemple, le modèle LDA nécessite de choisir le nombre de sujets K, le nombre d'itérations N (dans le cas de l'utilisation de l'échantillonneur de Gibbs) et deux hyperparamètres α et β affectant la répartition des sujets entre les documents et les termes.

Historiquement, les chercheurs ont développé différentes méthodes pour déterminer le nombre optimal K, mais sans tenir compte des autres paramètres (Arun et al., 2010a; Brunet et al.,

2004; Greene, O'Callaghan et Cunningham, 2014), laissant ainsi aux diverses implémentations de LDA le choix des valeurs par défaut. Par exemple, dans les librairies *Gensim*[1] et *Sklearn*[2], ces valeurs par défaut peuvent systématiquement diminuer la stabilité de la modélisation et ne peuvent être universellement utilisées sur différents corpus. Ensuite, plusieurs études ont démontré l'importance de l'ajustement adéquat des hyperparamètres α et β (Agrawal, Fu et Menzies, 2018; Maier et al., 2018; Panichella, 2021; Syed et Spruit, 2018). Il a également été remarqué que LDA souffre d'effet d'ordre, c'est-à-dire que si l'ordre des données d'entraînement (l'ordre des documents) est modifié, les différents sujets ne seront pas décrits par les mêmes regroupements de mots. Cet effet d'ordre introduit donc une erreur systématique pour toute analyse ou étude d'un corpus et est souvent décrit comme l'instabilité du modèle (Agrawal, Fu et Menzies, 2018). Conséquemment, une paramétrisation sous-optimale produira des résultats inexacts pouvant semer la confusion chez le praticien et rendant difficile la réplication de ses résultats.

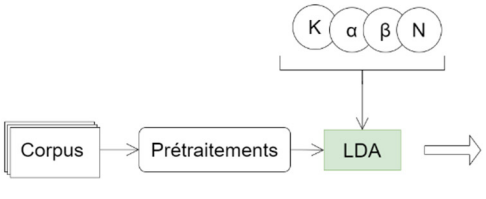
Par ces remarques, nous comprenons que ces paramètres (c'est-à-dire K, α, β et N) sont d'une importance considérable pour la modélisation des thématiques qui en résultent. Ainsi, la sélection de ces paramètres est essentielle afin de calibrer un modèle qui reflétera adéquatement les données et qui produira des résultats significatifs et interprétables. Jusqu'à présent, il n'y a pas de procédure statistique standard pour guider cette sélection. Elle reste donc l'une des tâches les plus compliquées dans l'application de la modélisation des sujets à l'aide de LDA. Il est d'autant plus important de ne pas perdre de vue que cette recherche des paramètres optimaux est basée sur le jeu de données que le praticien utilisera pour la calibration et que l'étape de prétraitement aura un impact important sur cette sélection (Denny et Spirling, 2018).

---

[1] https://radimrehurek.com/gensim/models/ldamodel.html
[2] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html

1.2.3 LDA ou BERTopic

Comme nous venons de l'expliquer, LDA repose sur l'hypothèse que chaque document est composé de plusieurs sujets (thèmes ou thématiques) et que chacun de ces sujets est caractérisé par une distribution de mots. Dans son processus de génération, LDA tente d'estimer les distributions de sujets latents qui décriront au mieux l'ensemble des différents documents du corpus à l'étude. De manière simplifiée, nous illustrons à la figure 1.1 un exemple des éléments nécessaires pour sa mise en oeuvre (corpus, prétraitement et paramètres) ainsi que ses deux éléments de sortie (matrices sujet-termes et document-sujets).



Matrice sujet-termes

Sujets	Poids	Termes
Sujet 1	40%	Fleur
	10%	Rose
	50%	Plante
Sujet 2	70%	Compagnie
	20%	Employé
	10%	Patron

Matrice document-sujets

Sujets	Poids	Documents
Sujet 1	40%	doc 1
	10%	doc 7
	50%	doc 3
Sujet 2	70%	doc 4
	20%	doc 6
	10%	doc 2

Figure 1.1 Mise en œuvre de LDA

La notion la plus importante à remarquer dans cette image est la représentation des éléments de sortie. Prenons par exemple le sujet 1 de la matrice sujet-termes : ce sujet est décrit par une liste de termes (Fleur, Rose, Plante) où leur poids respectif (40%, 10%, 50%) représente l'importance du terme pour décrire ce sujet. En ce qui concerne la matrice document-sujets, son interprétation est similaire à celle de la matrice sujet-termes à l'exception que les sujets sont décrits par des documents.

En revanche, tel qu'illustrer à la figure 1.2, l'auteur de BERTopic utilise le modèle de langue préentraîné appelé BERT (*Bidirectional Encoder Representations from Transformers*) pour encoder chaque document (Grootendorst, 2022). Il réduit ensuite la dimensionnalité de ces documents encodés avec l'algorithme UMAP, puis utilise un algorithme de clustering pour

regrouper les documents en sujets (HDBSCAN). Afin d'identifier les termes les plus descriptifs de chacun des clusters, il regroupe sous la forme d'un sac-de-mots (*Bag-of-words*) les documents relatifs à chacun des clusters pour ensuite appliquer une technique de décompte pondéré (c-TF-IDF[3]).

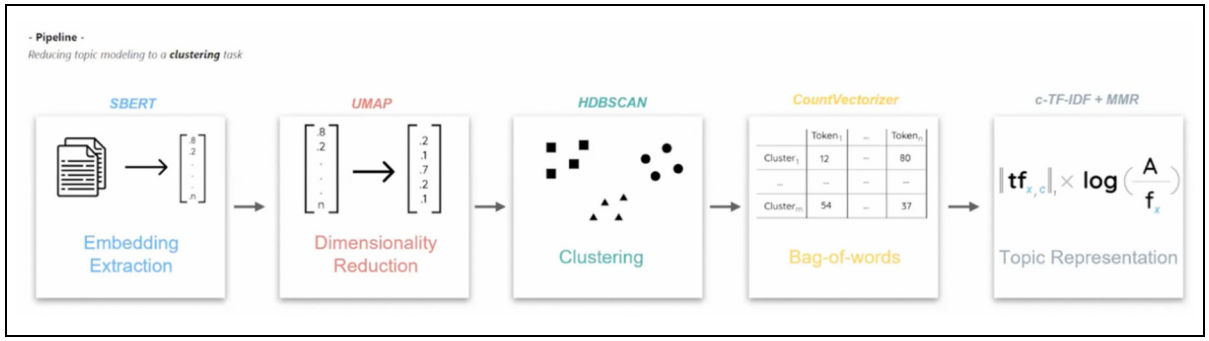


Figure 1.2 Mise en œuvre de BERTopic
tirée de co:here, 1 novembre 2022

Les éléments de sortie de BERTopic sont illustrés aux figures suivantes. La Figure 1.3 illustre la représentation sujet-termes pour le sujet 0 (*Topic 0*). Contrairement à représentation sujet-termes de LDA, les valeurs numériques de l'axe horizontal ne sont pas des probabilités, c'est plutôt une pondération de la fréquence du terme issue du calcul c-TF-IDF.

[3] c-TF-IDF est une adaptation de TF-IDF proposée par Grootendorst (2022) afin de calculer les poids des termes sur les documents qui constituent un cluster.

Figure 1.3 Représentation sujet-termes selon BERTopic tirée de co:here, 1 novembre 2022

La figure 1.4 tente de représenter la matrice document-sujets de LDA. Dans cet exemple, une réduction de la dimensionnalité a été appliquée afin de créer cette représentation en 2 dimensions. Pour ce faire, l'algorithme de réduction de la dimensionnalité (UMAP) a été utilisé pour réduire la représentation vectorielle d'un terme qui selon BERT est représenté sur 768 dimensions (Devlin et al., 2018). Ce que nous pouvons déduire de cette image est que le cluster de points orange condensés à gauche est regroupé à cause d'une similarité vectorielle. En ce qui concerne les autres points orange, nous pouvons déduire qu'ils sont vectoriellement moins similaires à l'agglomération et nous laisse pressentir qu'ils sont plus similaires à d'autres agglomérations. Finalement, tous ces points orange, peu importe leur position dans ce plan, peuvent être décrits par la liste de terme listé à la Figure 1.3.

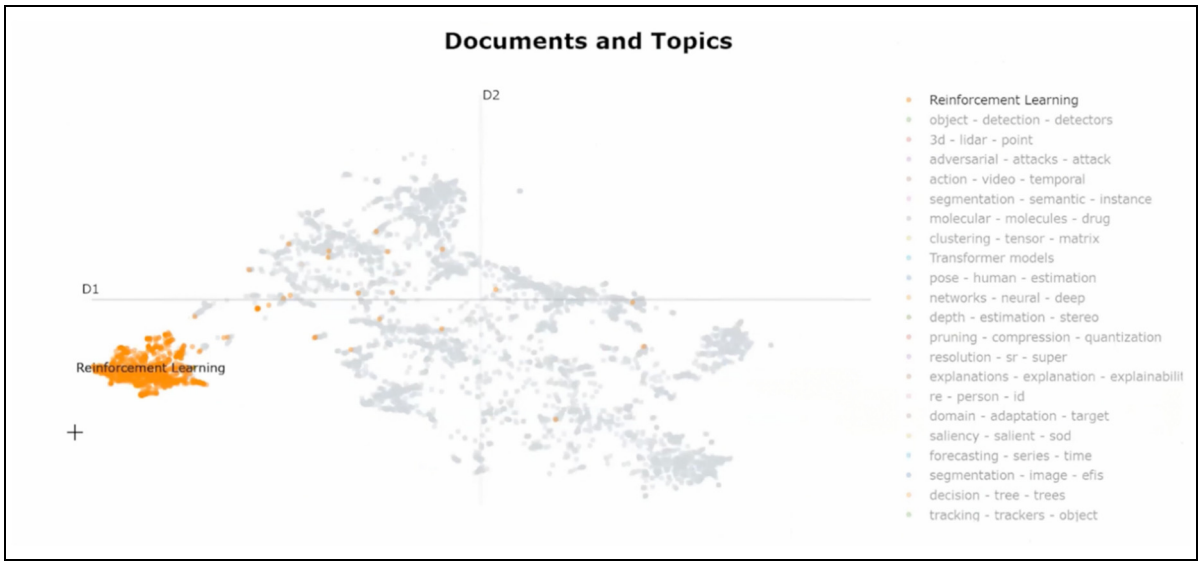


Figure 1.4 Visualisation document-sujets selon BERTopic
tirée de co:here, 1 novembre 2022

Les différences fondamentales entre ces deux méthodes résident dans leur approche pour la modélisation des thématiques. **LDA suppose que chaque document est composé de plusieurs thèmes**, tandis que BERTopic traite chaque document comme un vecteur unique pour l'analyse des thématiques, ce qui a pour conséquence qu**'un document appartient à un sujet**[4]. Cela signifie que LDA fournit des distributions de sujets latents qui sont pondérées en fonction de leur probabilité dans chaque document, tandis que BERTopic fournit des clusters de documents qui partagent des similitudes dans leur représentation vectorielle.

En termes de complexité, LDA peut être considéré comme étant plus simple que BERTopic en raison de la quantité de paramètres à ajuster. Même si BERTopic se dit être modulaire[5] dans sa mise en œuvre, c'est-à-dire que chacune des étapes illustrées à la Figure 1.2 peut être modifiée par d'autres algorithmes pouvant effectuer la tâche (ou même être éliminées), le choix

[4] Cette hypothèse va à l'encontre d'une évidence qu'un document peut aborder plusieurs sujets.
[5] https://maartengr.github.io/BERTopic/index.html#modularity

de chacun de ces algorithmes est accompagné de leurs propres jeux de paramètres qui doivent être fixés pour leur mise en œuvre.

En termes d'interprétabilité des résultats, LDA est souvent considéré comme plus facile à interpréter que BERTopic, car il fournit des distributions de sujets latents (matrice sujet-termes et matrice document-sujets) qui peuvent être associées à des mots spécifiques. Cependant, BERTopic fournit des clusters de documents qui peuvent être interprétés par l'analyse de leur contenu (représentation document-sujets). Il est également possible de récupérer les mots clés les plus importants pour chaque cluster en utilisant des techniques qui permettent d'évaluer l'importance d'un mot dans un document, par exemple c-TF-IDF, afin d'obtenir une représentation sujet-termes.

1.3 Évaluation de la performance d'un modèle de langue : de l'analyse qualitative aux mesures quantitatives vers l'introduction de la sagesse de la population

De manière générale, un praticien supposera que l'espace latent découvert par ce type de modèle de langue a une signification sémantique. Pour valider sa qualité, il peut réviser l'espace latent afin de vérifier s'il correspond bien à ses intuitions. Cependant, ce type d'évaluation qualitative demeure difficile à cause de son caractère subjectif et par de la nature générative de ces algorithmes non supervisé. Néanmoins, voici quelques approches qui sont proposées :

- Évaluer la composition des sujets (les listes de termes décrivant chaque sujet) via plusieurs générations en utilisant le même jeu de paramètres, mais en changeant la valeur d'amorçage (seed) (DiMaggio, Nag et Blei, 2013; Levy et Franklin, 2014).
- Examiner si les sujets identifiés sont reproductibles à travers plusieurs exécutions du modèle en faisant varier le nombre de sujets (K) (Jacobi, Van Atteveldt et Welbers, 2018; Levy et Franklin, 2014).
- Vérifier s'il est possible de reproduire les mêmes sujets en utilisant des échantillons plus petits du corpus (Biel et Gatica-Perez, 2014).

- Passer en revue les mots ayant les probabilités les plus élevées pour chaque sujet et essayer de trouver un terme général les décrivant.
- Sélectionner et lire des échantillons de documents appartenant respectivement à chaque sujet (Elgesem, Feinerer et Steskal, 2016; Jacobi, Van Atteveldt et Welbers, 2018).

Par cette énumération, nous pouvons déduire que l'analyse qualitative peut être longue et fastidieuse en plus de nécessiter une connaissance du domaine de la part du praticien.

Afin de contourner cette problématique, des chercheurs ont proposé différentes approches quantitatives. Pour ce faire, les chercheurs font appel à une variété de mesures internes utilisées lors de l'entrainement du modèle. Voici quelques approches possibles :

- Utiliser une heuristique qui calcule la fréquence de cooccurrence des termes les plus probables de décrire un sujet telle que celle proposée par Mimno *et al.* (2011).
- Utiliser l'information mutuelle qui vise à identifier lequel des termes les plus probables apporte l'information la plus significative à un sujet donné.
- Utiliser des mesures statistiques comme la vraisemblance ou la perplexité (Blei, Ng et Jordan, 2003). Pour leur application, un modèle doit être entrainé sur la majorité du corpus, par exemple 90% de tous les documents, puis validé sur l'ensemble restant, ici 10% des documents de la collection non inclus dans le processus d'entrainement. La qualité de l'ajustement du modèle (la vraisemblance) est estimée par la façon dont le modèle prédit la plus petite portion de documents retenue. Une plus grande vraisemblance correspond à une mesure de perplexité plus faible.
- Vérifier si les sujets sont suffisamment distincts les uns des autres en appliquant des techniques de regroupement hiérarchique (Marshall, 2013; Puschmann et Scheffler, 2016).

Bien que de telles mesures sont utiles pour évaluer le caractère prédictif du modèle, elles ne répondent pas aux objectifs plus exploratoires de la modélisation par sujet. Ainsi, d'autres études optent plutôt pour des stratégies utilisant une validation externe :

- Évaluation à l'aide d'experts (Levy et Franklin, 2014).
- Utilisation de systèmes de codage (Guo et al., 2016; Jacobi, Van Atteveldt et Welbers, 2018).
- Utilisation de schémas temporels afin de faire correspondre les sujets du modèle à des événements qui se sont produits dans le cadre temporel de l'étude (Evans, 2014; Newman et al., 2006).

Bref, il n'existe pas de procédure générale bien définie pour évaluer la qualité de ce type de modèle.

1.3.1 La sagesse de la population

Chang *et al.* (2009) ont proposé une méthode d'évaluation manuelle qui consiste à transformer la tâche de validation en un jeu pouvant être distribué à une large population. L'idée générale des différentes tâches est de demander aux participants d'identifier l'intrus parmi une liste de choix. Pour un sujet donné, ils sélectionnent, les neuf premiers termes et y insèrent un nouveau terme, l'intrus, provenant d'un autre sujet dans lequel il a une forte probabilité. Si les participants arrivent à identifier l'intrus, alors les auteurs qualifient ce sujet comme étant en accord, voire cohérent avec le jugement humain.

Dans cette étude, ces auteurs ont extrait les sujets latents de deux différents corpus (1000 articles provenant de Wikipédia et 8447 articles provenant du *New York Times*) contenant de l'information grand public à l'aide des modèles pLSA, *Correlated Topic Model* (CTM) et LDA. Pour des nombres de sujets (K) fixés à 50, 100 et 150, ils ont calculé le score de vraisemblance (discuté à la section précédente) pour chacune des modélisations et ont soumis ces sujets à l'évaluation manuelle de leur cohérence telle que décrite au paragraphe précédent

(identifier l'intrus). Ils ont constaté que les modèles identifiés comme ayant les sujets les plus cohérents selon l'évaluation manuelle ne sont pas toujours ceux qui présentent les meilleurs résultats de vraisemblance. Cette constatation a donc motivé ces auteurs à suggérer que les praticiens qui utilisent ces types de modèles devraient plutôt se concentrer à évaluer la cohérence des éléments de sortie du modèle pour la tâche à laquelle il est dédié (évaluation qualitative) plutôt que se consacrer à optimiser le score de la vraisemblance (évaluation quantitative).

Au vu de ces résultats, plusieurs chercheurs ont conduit des études portant sur le développement de nouvelles mesures quantitatives fondées sur l'interprétabilité sémantique des sujets découverts. Autrement dit, elles tentent d'évaluer la cohérence des *n* termes les plus probables des différents sujets latents inférés par l'algorithme de modélisation. Cet épisode a donné naissance à une famille de mesures portant le nom de *Coherence*. Nous pouvons citer les travaux de Newman *et al.* (2010) qui incorporent des données externes à leur mesure de cohérence et la méthode développée par Mimno *et al.* (2011) qui utilise seulement les données du corpus à l'étude. Ces mesures, évaluées à l'aide de participants, offrent des résultats qui semble corréler avec le jugement humain. Il convient de noter que dans la suite de ce mémoire, le terme « *Cohérence »* écrit en anglais et en italique fera référence aux heuristiques appartenant à cette famille de mesure et le que le terme « cohérence » fera référence à l'accord avec le jugement humain.

1.4 Conclusion

Étant donné ce qui précède et l'objectif général de ce projet de recherche (développer un système de recommandations vidéo fondé sur l'analyse des sous-titres via un modèle de langue), le système que nous proposons utilisera, dans un premier temps (et pour ce projet), une approche de filtrage sur le contenu qui pourrait éventuellement se diriger vers les approches de recommandations hybrides dans des recherches futures (non couvertes dans ce mémoire). Nous avons décidé d'écarter l'approche du filtrage collaboratif pour notre système

de recommandations dédié aux enfants autistes, car ceux-ci ont tendance à préférer des activités connues et prévisibles. De plus, nous ne disposons pas d'un historique de données vidéos pour lesquelles les enfants ont montré un intérêt, et que nous voulons développer une méthodologie généralisable à l'analyse des sous-titres. De plus, en choisissant l'approche par filtrage sur le contenu, nous évitons les inconvénients du démarrage à froid dans le cas de l'ajout d'un nouvel item et le cas du mouton gris (cf. Tableau 1.1).

Tel que mentionné lors de la description de l'approche par filtrage sur le contenu, usuellement, ce type de système de recommandations vidéo utilise les métadonnées associées aux vidéos. La popularité quant à l'utilisation des métadonnées réside dans la simplicité à les exploiter. En effet, elles nécessitent peu de prétraitement afin d'être utilisées par le processus du calcul des recommandations et ne requiert pas l'ajout d'une étape de modélisation (par exemple des sous-titres). Cependant, l'utilisation de ce type de données peut facilement générer de mauvaises recommandations dues à la qualité variable de ces descriptifs. De plus, nous partons de l'hypothèse soutenue (Basu et al., 2016; Mei et al., 2011; Zhu, Shyu et Wang, 2013) que l'utilisation des sous-titres dans un contexte de recommandations vidéo offre une source d'information ayant un potentiel descriptif prometteur quant à l'interprétation de la vidéo.

Afin de modéliser les sous-titres, nous avons opté pour la technique de la modélisation par sujet. À la suite de la description des différents algorithmes appartenant à cette technique et de la comparaison entre LDA et BERTopic, nous priorisons l'utilisation de LDA. Les principaux facteurs qui ont motivé ce choix sont l'interprétabilité des matrices de sorties, le nombre d'étapes ainsi que la quantité de paramètres à fixer afin de le mettre en œuvre comparativement à BERTopic. D'un point de vue personnel, BERTopic semble « cacher la poussière sous le tapis » en ce qui concerne la question fondamentale de la modélisation par sujet relative au nombre de sujets abordés dans le corpus. Bien que BERTopic tente d'estimer le nombre de sujets abordés, il y arrive en vectorisant le corpus à l'étude dans un espace prédéfini par un modèle de langue préentraîné (BERT) et en utilisant un algorithme de clustering comme HDBSCAN, qui soit dit en passant n'a pas besoin de cette valeur pour

identifier des regroupements mais qui demande de fixer d'autres paramètres. Nous jugeons que l'ensemble de cette méthodologie (BERTopic) nous éloigne du corpus à l'étude. Quant à LDA, même si celui-ci effectue son analyse directement sur le corpus auquel il est soumis, le choix d'une paramétrisation adéquate reste l'une des tâches les plus compliquées, car il n'y a pas de procédure statistique standard pour guider cette sélection. Jusqu'à présent, nous n'avons donné que peu de détails sur le processus génératif de LDA, de ses différents types d'instabilité ainsi que sur les procédures pour estimer une paramétrisation quasi optimale, mais nous dédions le chapitre 4 à cet approfondissement.

Finalement, en ce qui concerne l'évaluation du modèle, nous allons suivre le conseil donné par Chan *et al.* (2009) qui stipule de confronter cette modélisation à la tâche à laquelle elle est dédiée. Pour ce faire, nous allons développer une application web qui sera accessible à la communauté universitaire afin que des participants puissent évaluer cette modélisation dans un contexte de recommandions vidéo ainsi que par des tests d'intrusion.

CHAPITRE 2

MÉTHODOLOGIE GÉNÉRALE

Tel que mentionné en introduction, le programme de recherche dans lequel s'inscrit notre projet a pour objectif de développer un système de recommandations vidéo basé sur l'analyse multimodale des caractéristiques contenues dans les vidéos. L'élaboration d'un tel système est complexe, car il nécessite le développement de différentes composantes hétérogènes pour ensuite les combiner afin de concevoir l'engin de recommandations. Par souci de clarté, ce chapitre brossera une vue globale de la méthodologie pour ensuite dédier les subséquents chapitres à l'approfondissement des composantes spécifiques de ce projet de recherche. Le schéma suivant expose ces principales composantes ainsi que leurs interactions.

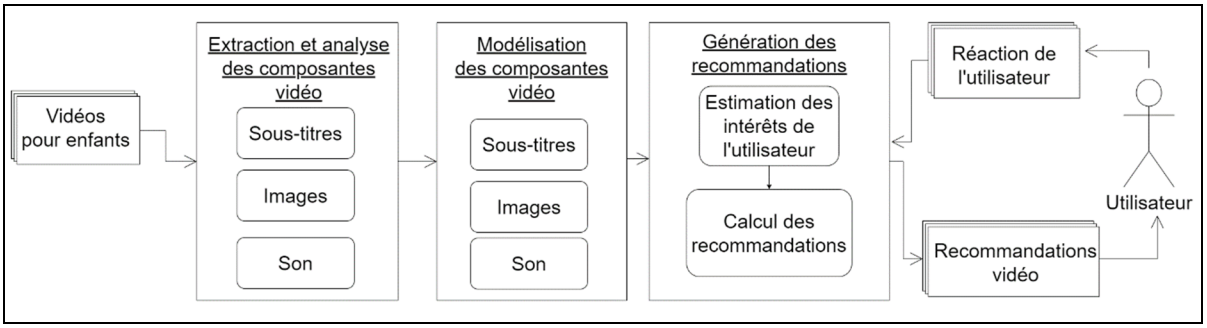


Figure 2.1 Principales composantes et interactions du système de recommandations

De manière générale, la boîte ***extraction et analyse des composantes vidéos*** contient les étapes d'extraction et de prétraitement des différentes composantes d'une vidéo (sous-titres, images, son). La boîte ***modélisation des composantes vidéo*** concerne les étapes de modélisation des différents types de signaux caractérisant les ***vidéos pour enfants.*** La boîte ***génération des recommandations*** combine les modélisations des différents signaux et les réactions de l'utilisateur afin d'y estimer le centre d'intérêt dans le but de générer une nouvelle liste de recommandations personnalisées.

Puisque notre projet de recherche se consacre à l'utilisation des sous-titres, nous simplifions le schéma précédent par la nouvelle représentation illustrée ci-dessous. Nous avons également modifié le titre de certaines composantes afin de refléter ce contexte.

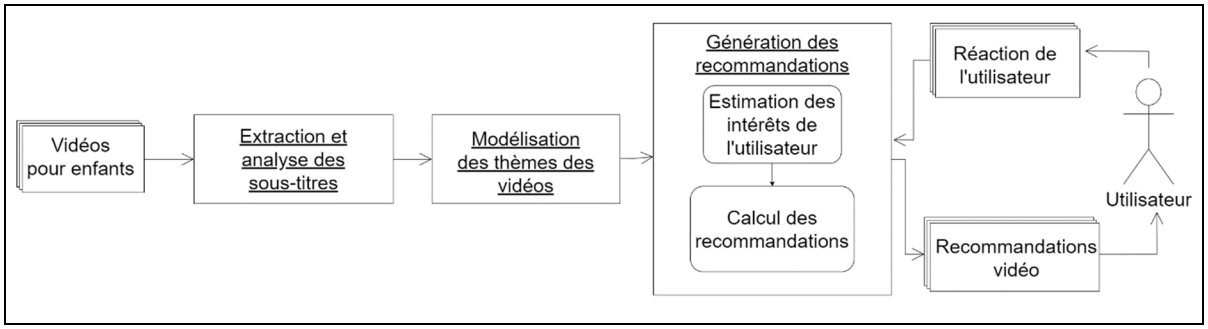


Figure 2.2 Schéma des principales composantes du système de recommandations basé sur l'analyse des sous-titres

Désormais, nous réfèrerons à cette nouvelle dénomination pour identifier les principales composantes du système de recommandations. De plus, les sous-sections suivantes décrivent de manière générale les différentes méthodologies qui ont permis l'élaboration de chacune de ces principales composantes.

2.1 Extraction et analyse des sous-titres

La réalisation de la composante d'extraction et d'analyse des sous-titres permettra de répondre au sous-objectif 1. Ce sous-objectif a pour but d'extraire les sous-titres de vidéos en ligne afin de concevoir le corpus à l'étude, de l'analyser et d'y apporter quelques modifications afin que de générer des sac-de-mots de différentes dimensions qui seront utilisés par la composante de ***modélisation des thèmes des vidéos***.

Les vidéos sélectionnées sont les épisodes de la première saison de *Peppa Pig* offerts sur la plateforme de YouTube. Cette série d'animation pour enfant a été sélectionnée pour satisfaire

les intérêts d'enfants d'âge préscolaire et, aussi, car elle a été suggérée par des parents d'enfants atteints du TSA.

L'utilisation de ce type de données, c'est-à-dire les sous-titres provenant de vidéos pour enfant, n'est pas un corpus usuellement étudié dans le domaine du traitement automatique des langues naturelles. De manière générale, les corpus étudiés proviennent de différentes sources anglophones telles que des sources journalistiques reconnues comme *Reuter* et le très connu jeu de données le *Twenty Newsgroups,* des résumés d'articles scientifiques publiés dans *The Journal of NIH Research* ou même, des sources d'informations populaires telles que *Stack Overflow*, *Twitter* et *Wikipédia*.

Par conséquent, l'usage de sous-titres aura pour effet de soulever quelques questions à caractère technique et légal. Techniquement, ces questions concerneront la sélection des épisodes offerts sur la plateforme de YouTube, la sélection d'un outil d'extraction automatique des sous-titres, l'impact de l'encodage des transcriptions sur la définition de la longueur d'un document et finalement et les étapes de prétraitement nécessaires pour les fins de modélisation. Légalement, la principale question concernera le respect du droit d'auteur dans notre contexte d'utilisation. Autrement dit, est-ce légal d'extraire les sous-titres de vidéos mises à la disposition du grand public sur YouTube ?

De façon plus formelle, l'élaboration de cette composante sera guidée par les trois principales questions présentées ci-dessous et suivra aux pages suivantes une description sommaire de la méthodologie pour y répondre.

- Question 1 : Quelle combinaison « outil d'extraction - chaîne YouTube » permet d'obtenir les sous-titres générés automatiquement qui sont les plus fidèles à la valeur témoin (script original) ?
- Question 2 : Quelles sont les durées minimales d'une séquence vidéo permettant de générer un document contenant une valeur minimale de termes significatifs ?

- Question 3 : Devons-nous considérer l'ajout d'un prétraitement spécialisé sur le nom des personnages ?

Question 1 : Quelle combinaison « outil d'extraction - chaîne YouTube » permet d'obtenir les sous-titres générés automatiquement qui sont les plus fidèles à la valeur témoin (script original) ?

La sélection de la chaîne YouTube ayant les sous-titres les plus fidèles à leur valeur témoin sera possible grâce à une analyse comparative des différentes combinaisons « outil d'extraction - chaîne YouTube ». Cette étape est importante puisque nous avons remarqué que la qualité des sous-titres recueillis diffère pour une même vidéo en fonction de l'URL d'une vidéo et de l'outil d'extraction utilisé. Cette différence peut s'expliquer par les différentes implémentations des outils d'extraction utilisés et par les différentes techniques mises à la disposition du détenteur de la vidéo pour générer les sous-titres.

Question 2 : Quelles sont les durées minimales d'une séquence vidéo permettant de générer un document contenant une valeur minimale de termes significatifs ?

L'idée de fragmenter une vidéo en plusieurs séquences est d'émettre à l'utilisateur des fragments vidéo qui abordent des sujets précis. Par exemple, une vidéo de cinq minutes peut aborder plusieurs thématiques différentes et, sans cette étape de segmentation, nous pourrions difficilement recommander une séquence précise de la vidéo. De plus, puisque nous avons recours à des outils d'extractions, la segmentation d'une vidéo est tributaire de l'encodage des transcriptions.

Pour ce faire, nous évaluons l'impact de ces subdivisions sur le nombre de termes significatifs et la durée afin de générer un document. Ces considérations nous permettent d'identifier des valeurs candidates de longueur d'un document.

Question 3 : Devons-nous considérer l'ajout d'un prétraitement spécialisé sur le nom des personnages ?

Après l'écoute de quelques épisodes de cette série pour enfant, nous avons remarqué que la nomenclature utilisée par l'auteur afin de nommer les différents personnages est basée sur la répétition de termes liés aux espèces animales. Cette accentuation, dans un contexte de modélisation par sujet, risque de perturber la stabilité de ce type de modèle à cause de leur fréquence anormalement élevée comparativement aux autres termes décrivant le vocabulaire employé.

Pour mitiger cet effet, nous proposons quelques modifications à cette nomenclature afin de créer différents sac-de-mots qui seront ensuite soumis à notre méthodologie de modélisation.

À l'issue de cette première étape d'extraction et d'analyse, nous avons une sélection de plusieurs sac-de-mots qui sont utilisés à l'étape suivante pour les fins de modélisation par sujets.

2.2 Modélisation des thèmes des vidéos

Cette composante permettra de répondre aux sous-objectifs 2 et 3 qui concernent la modélisation des thématiques contenues dans un corpus de texte. Afin de voir automatiquement émerger des thématiques contenues dans les sous-titres des vidéos, nous optons pour l'utilisation du modèle de langue LDA. Cependant, la difficulté quant à l'utilisation de ce modèle réside dans l'identification de ses paramètres ainsi que dans la validation de la stabilité de la solution finale, car il n'existe aucune paramétrisation universelle pouvant être réutilisée sur différents corpus soumis à ce type d'analyse. Généralement, pour vérifier la qualité d'un modèle issu d'une paramétrisation, le praticien s'adonnera à l'évaluation de la cohérence sémantique des différents sujets (regroupements de mots).

Ensuite, si celui-ci veut évaluer l'impact qu'aura de modifier un paramètre sur la modélisation des thématiques, celui-ci devra réévaluer la cohérence des différents sujets, ce qui devient rapidement une tâche longue et ardue. Afin de surpasser ces difficultés, nous proposons un cadriciel combinant un algorithme de recherche suivi d'une évaluation quantitative afin de chiffrer la stabilité de la paramétrisation quasi optimale identifiée.

L'élaboration de ce cadriciel permettra de répondre au sous-objectif 2 et est préalable à la modélisation des thématiques contenues dans les sous-titres des vidéos. En appliquant ce cadriciel sur les différents sac-de-mots issus de la composante ***extraction et analyse de sous-titres***, il sera possible de sélectionner la modélisation finale qui nous servira de base de calcul pour la génération des recommandations et de répondre au sous-objectif 3.

Les sous-sections suivantes détaillent ces deux processus en lien avec les sous-objectifs 2 et 3.

2.2.1 Optimisation du modèle de langue Latent Dirichlet Allocation (LDA)

Dans le contexte de notre projet, nous optons pour l'utilisation de l'algorithme LDA qui appartient à la famille de la modélisation par sujet. L'intérêt quant à son utilisation est qu'il permet, de manière non supervisée, d'extraire les sujets latents d'un corpus à l'étude. Cependant, la difficulté quant à son utilisation réside dans sa paramétrisation. Étant de nature générative et probabiliste, chaque modélisation utilisant les mêmes paramètres sur un même corpus produira différentes distributions. Une paramétrisation adéquate favorise la réplication de distributions similaires, et, dans ce cas, on pourra caractériser le modèle comme étant stable.

L'optimisation des paramètres est un domaine de recherche actif en apprentissage machine. Que ce soit dans le domaine de la recherche opérationnelle, des mathématiques appliquées, ou de l'analyse numérique, les algorithmes d'optimisation permettent d'identifier une paramétrisation quasi-optimale pour maximiser les performances de l'algorithme que l'on cherche à optimiser.

En se basant notamment sur les travaux de Panichella (2021) et Agrawal *et al.* (2018) (sur la recherche des paramètres) et Mantyla *et al.* (2018) et Rierger *et al.* (2020) (sur la stabilité des sujets latents découverts par LDA), il est possible de développer une nouvelle approche en combinant leurs méthodes. Notre approche est schématisée à la Figure 2.3 et est décrite ci-dessous. Notez que les rectangles représentent une succession d'étapes et les trapèzes représentent un élément de sortie.

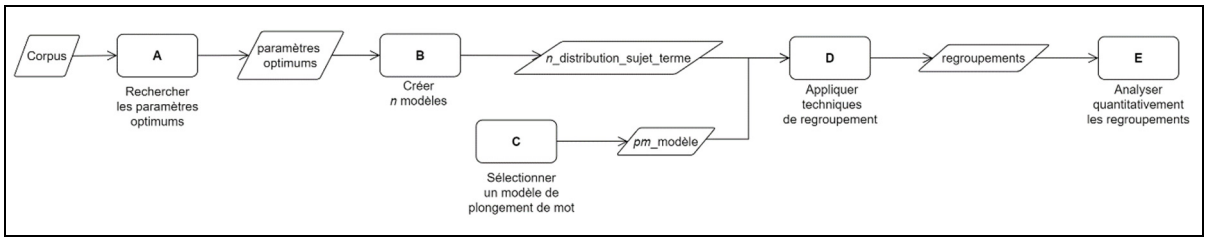


Figure 2.3 Optimisation d'un modèle de langue

Recherche des paramètres optimums (A)

Tel quel suggéré par Panichella (2021), nous utiliserons un Algorithme Génétique (AG) d'optimisation pour déterminer les paramètres optimums K, α et β de LDA et N qui est le nombre d'itérations de l'échantillonnage Gibbs.

Création de *n* modèles (B)

Avec les paramètres optimums identifiés, nous générons *n* modèles différents en modifiant l'ordre des documents du corpus (Agrawal, Fu et Menzies, 2018; Panichella, 2021). Nous aurons alors *n* matrices sujet-termes.

Modèles de plongement de mots (C)

Il a été démontré que l'espace vectoriel produit par un algorithme de plongement de mots tel que GloVe tend à regrouper les mots qui sont sémantiquement similaires (Pennington, Socher et Manning, 2014). L'idée est de tirer avantage du potentiel synonymique et de similarité sémantique (Mantyla, Claes et Farooq, 2018) dans l'espace vectoriel généré par GloVe pour

valider la cohésion des termes les plus probables constituant les différents sujets latents découverts.

Pour ce faire, dénotons l'espace vectoriel des mots générés par GloVe comme étant la matrice G constituée de t termes et v vecteurs de poids, $G = [t, v]$. En effectuant le produit matriciel entre la matrice sujet-termes $ST = [s, t]$ et G, nous obtenons la matrice sujet-vecteur $SV = [s, v]$ sur laquelle nous appliquons les techniques de regroupement et d'évaluation.

Technique de regroupement (D)

L'algorithme k-médoïde est utilisé pour regrouper en K sujets les *n* matrices SV.

Technique d'évaluation (E)

Afin de quantifier la stabilité du modèle, nous avons développé une métrique que nous appelons le score en stabilité. Cette mesure combine sous la forme d'une moyenne arithmétique, les résultats de différentes mesures d'évaluation (internes et externes) des regroupements identifiés à l'étape D.

2.2.2 Optimisation d'un modèle de langue sur un corpus constitué de transcriptions automatiques de vidéo pour enfant

La création du modèle de langue issu des sous-titres des vidéos pour enfant s'effectue à l'aide de la technique d'optimisation préalablement développée qui est appliquée sur les différents sac-de-mots produits à la composante ***extraction et analyse de sous-titres***.

Puisque les sac-de-mots utilisés diffèrent d'une part par les différentes techniques de substitution appliquées sur la nomenclature des personnages et d'autre part via les différentes longueurs de documents identifiées, nous comparons les résultats obtenus à l'étape d'analyse en stabilité à une analyse qualitative et quantitative des distributions document-sujets et sujet-termes et afin de sélectionner notre modèle final (i.e. la technique de substitution et la longueur

de documents). Cette dernière étape nous permet de valider le pouvoir d'identification de l'analyse en stabilité proposée à la section 2.2.1 précédente.

2.3 Génération des recommandations

Cette composante permettra de répondre aux sous-objectifs 4 et 5 qui concernent respectivement le développement de l'engin de recommandations, de la carte sémantique et l'évaluation du système dans son ensemble.

2.3.1 Génération et visualisation des recommandations vidéo et des centres d'intérêt de l'utilisateur

L'émission des recommandations vers les utilisateurs est possible grâce à un système informatique que nous avons déployé sur un serveur Web. D'un point de vue « développement Web », ce système respecte une architecture client-serveur. Par abus de langage, ce système informatique est communément identifié comme étant un « système de recommandations ». Conséquemment, il est constitué de deux entités distinctes, la partie client et la partie serveur.

Côté client, celui-ci a pour objectif d'offrir différentes interfaces qui permettent d'authentifier l'utilisateur et de visualiser les recommandations vidéo ainsi que les éléments permettant l'interaction avec la partie serveur. Dans ce mémoire, nous nous attarderons principalement sur le développement de l'interface que nous définissons comme la « carte sémantique ». Cette carte permet de présenter l'évolution du centre d'intérêt de l'utilisateur au sein des sujets latents découverts par LDA. La section 6.3.2 présente cette carte plus en détail.

Côté serveur, celui-ci a pour objectif d'estimer le centre d'intérêt de l'utilisateur afin de personnaliser l'émission des recommandations. La section 6.3.1 de ce mémoire présentent les calculs d'estimation du centre d'intérêt, sa mise à jour et le calcul des recommandations.

2.3.2 Évaluation du système par les utilisateurs

Pour évaluer le potentiel de notre solution, nous avons effectué une première expérimentation au sein d'une population neurotypique issue de la communauté universitaire. Les différents tests permettent d'évaluer la qualité des recommandations, l'interprétabilité de la carte sémantique ainsi que la qualité du modèle LDA via les tests d'intrusion développés par Chan *et al*. (2009) pour lesquels nous avons dû apporter quelques modifications afin de les adapter à notre contexte (voir section 6.3.3).

CHAPITRE 3

EXTRACTION ET ANALYSE DES SOUS-TITRES

Ce chapitre détaille la démarche qui permet d'atteindre le premier sous-objectif de ce projet de recherche. Ce sous-objectif concerne le développement d'un mécanisme d'extraction automatique des sous-titres à partir de vidéos en ligne et de leur analyse. Il est également la première composante de notre système de recommandations présenté à la Figure 2.2. Une fois complétée, cette composante permettra de concevoir les différents sac-de-mots qui seront utilisés par la composante de modélisation des thèmes des vidéos.

À la section 2.1 du chapitre précédent, l'atteinte de ce sous-objectif a été présentée sous la forme de trois questions. Ce chapitre détaillera la démarche qui nous a permis de répondre à chacune de ces questions.

3.1 Question 1 : Quelle combinaison « outil d'extraction - chaîne YouTube » permet d'obtenir les sous-titres générés automatiquement qui sont les plus fidèles à la valeur témoin ?

Pour la grande majorité des vidéos mises à la disposition du public sur la plateforme YouTube, il est possible de les visionner accompagnées de leurs sous-titres. Ceux-ci peuvent avoir été générés automatiquement à l'aide d'un algorithme de reconnaissance de la parole[6] ou ajoutés manuellement par le détenteur de la vidéo.

Quant aux transcriptions automatiquement générées, certaines divergences entre le signal audible et les sous-titres peuvent être remarquées. Cela est dû aux faiblesses des algorithmes de reconnaissance de la parole qui ne peuvent actuellement pas offrir une transcription exacte.

---

[6] https://support.google.com/youtube/answer/6373554?hl=en#:~:text=YouTube%20can%20use%20speech%20recognition,to%20provide%20professional%20captions%20first.

Des facteurs tels qu'une durée trop longue d'une vidéo, la langue, une faible qualité audio ou l'utilisation de mots spécifiques non reconnus ou mal interprétés peuvent affecter la qualité d'une transcription. Des périodes de silence en début de vidéo, la multiplicité des intervenants et le chevauchement des discours peuvent également perturber le logiciel de transcription pour ainsi introduire des erreurs.

Par la suite, un utilisateur voulant télécharger ces sous-titres par programmation, l'API de YouTube offre une fonction pour les obtenir[7]. Cependant, une fois les étapes d'authentification effectuées, la réponse du serveur demeure non concluante. Voici la réponse :

> *forbidden* 403[8] *« The permissions associated with the request are not sufficient to download the caption track. The request might not be properly authorized, or the video order might not have enabled third-party contributions for this caption. »*

Une explication possible est que cette fonction n'est disponible que pour le détenteur de la vidéo. Afin que d'autres utilisateurs puissent bénéficier de ce même privilège, le détenteur doit autoriser la contribution des tiers, ce que la plupart des détenteurs ne font pas. De plus, depuis le 28 septembre 2020, YouTube a discontinué cette option[9]. Cependant, une seconde route non documentée dans la documentation officielle de YouTube permet de contourner ce problème d'authentification. D'un point de vue communication avec un serveur, cette route d'API est la suivante :

https://www.youtube.com/**get_video_info**?video_id={videoId}

Cette route, qui ne nécessite aucune authentification, permet de retourner une chaîne de caractères encodée en pourcentage qui, si les sous-titres sont disponibles, contiendra le

---

[7] https://developers.google.com/youtube/v3/docs/captions
[8] https://developers.google.com/youtube/v3/docs/captions/download#errors
[9] https://support.google.com/youtube/thread/61967856?hl=en

paramètre *captionTracks* spécifiant une adresse URL pour obtenir les sous-titres. De plus, c'est cette route qui est exploitée par les différents outils offrant la fonctionnalité de téléchargement des sous-titres.

Toutefois, nous avons remarqué qu'en fonction de l'URL d'une vidéo et de l'outil d'extraction automatique de sous-titres utilisé, la qualité du texte recueilli diffère pour une même vidéo.

Afin de sélectionner l'outil et l'URL permettant d'obtenir les sous-titres les plus fidèles à la valeur témoin, cette première étape a pour but d'effectuer une analyse comparative quant à la qualité des sous-titres qu'il est possible d'extraire par programmation. Globalement, cette analyse consiste à calculer, pour l'ensemble des épisodes, le niveau de similarité entre la transcription et sa valeur témoin correspondante.

Les sous-sections suivantes décriront les notions de base qui nous ont permis de calculer la similarité entre les sous-titres et la valeur témoin. Suivront les détails concernant les différentes chaînes YouTube, la provenance des valeurs témoins et les différents outils d'extractions qui ont été nécessaires pour la mise en œuvre de la méthodologie développée. Finalement suivra une discussion sur les différents résultats obtenus qui nous permettra de répondre à cette première question concernant la sélection d'une chaîne YouTube et d'un outil d'extraction.

3.1.1 Calcul de la similarité entre les transcriptions et la valeur témoin

Mesurer la similarité entre des termes, des phrases, des paragraphes ou même entre des ensembles de textes est une tâche largement appliquée dans le domaine du traitement des langues naturelles. La littérature dans ce domaine est excessivement abondante et cela a pour conséquence que la quantité des métriques permettant d'évaluer la similarité évolue en ce sens. Cela dit, il existe une multitude de mesures permettant de quantifier la similarité. Globalement, l'ensemble des mesures dites « de similarité » se scindent en deux catégories : les mesures de similarité syntaxiques et les mesures de similarité sémantiques.

La similarité dite syntaxique[10] utilise des algorithmes qui opèrent sur des séquences de chaînes de caractères et/ou sur la composition de ces chaînes de caractères. Ce type de similarité est communément appelé *String-based similarity* (Gomaa et Fahmy, 2013; Wang et Dong, 2020). En fonction de l'unité utilisée, le caractère ou la chaîne de caractères (un terme), les différents algorithmes appartiennent respectivement aux catégories *character based* ou *term based*.

La similarité sémantique est un concept selon lequel un ensemble de documents ou de termes se voit attribuer une métrique basée sur la ressemblance de leur signification (contenu sémantique). Dans un contexte où l'applicabilité est algorithmique, le calcul de la similarité peut se faire soit en définissant une similarité topologique (par exemple, en définissant la distance entre les mots à l'aide d'une ontologie) ou soit en définissant une similarité statistique qui utilise un modèle d'espace vectoriel pour corréler les termes et les contextes à partir d'un corpus de texte approprié (co-occurrence). Ces mesures de similarité permettent d'évaluer la notion de concept et de sens. Deux concepts sont considérés comme sémantiquement similaires s'il y a une synonymie, une hyponymie ou une antonymie entre eux. Deux sens de mots sont considérés comme sémantiquement liés s'il existe au moins une relation lexico-sémantique entre eux. En somme, les différentes mesures de similarité sémantique se divisent en deux grandes familles, soit les approches statistiques basées sur le corpus ou les approches topologiques basées sur les domaines de connaissance. Les approches statistiques utilisent les informations contenues dans les textes analysés pour calculer le niveau de similarité. Citons par exemple les mesures comme LSA, pLSA, LDA, l'analyse sémantique explicite (ESA) et la distance normalisée de Google (NGD). Les approches topologiques basées sur les domaines de connaissance déterminent le niveau de similarité à l'aide d'information tirée de réseaux sémantiques. Mentionnions par exemple les réseaux sémantiques comme WordNet et WOLF dont leurs concepts sont respectivement issus de l'anglais et du français.

---

[10] L'expression « similarité syntaxique » est un abus de langage puisque le calcul est fondé sur des comparaisons de chaînes de caractères et non sur des critères relevant de la syntaxe des langues naturelles.

Par conséquent, il existe une multitude de techniques afin de calculer la similarité entre deux textes qui peuvent varier d'un domaine à l'autre. De manière générale, le calcul de la similarité s'effectue en trois étapes :

1. Sélectionner une mesure de similarité.
2. Convertir les éléments textuels dans le format approprié à la mesure sélectionnée.
3. Appliquer la mesure de similarité.

Comme indiqué par l'étape 2, la représentation des éléments textuels dépend du type de mesure choisie. Par exemple, les mesures de similarité sémantique de type statistique utilisent une représentation vectorielle pour représenter les différents documents qui constituent le corpus à l'étude. Ces documents sont alors représentés comme des vecteurs qui représentent les termes essentiels du corpus. Cet ensemble de vecteurs est communément appelé le sac-de-mots où les termes sont considérés comme indépendants et dont leur ordre est sans importance. Pour chacun des vecteurs, la valeur associée à chaque terme est appelée le poids du terme et est généralement calculée à l'aide d'une fonction de fréquence. La représentation d'un document sous cette forme vectorielle se déroule en deux étapes :

1. Extraire les termes pertinents du document (mise en jeton, suppression des ponctuations, suppression des mots vides, lemmatisation, etc.)
2. Calculer les poids des termes sélectionnés (méthode booléenne, *Terms-Frequency* (TF), *Term Frequency–Inverse Document Frequency* (TF-IDF), etc.)

Pour les mesures de similarité syntaxique (*character based* ou *term based*), une simple représentation sous la forme de chaîne de caractères est nécessaire, car leur fonctionnement est principalement basé sur le décompte du nombre minimal de caractères qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre.

Pour cette analyse comparative, nous avons fait le choix d'utiliser des mesures de similarités syntaxiques, car les textes en comparaison (transcription et valeur témoin) devraient contenir

les mêmes séquences de termes à l'exception des différences (erreurs) générées par les algorithmes de transcription automatique. Pour ce faire, nous avons sélectionné 2 mesures qui utilisent le caractère comme unité de calcul (*character based*) et une mesure qui utilise le terme (*term based)*. Ces dernières sont décrites ci-dessous :

- **Damerau-Levenshtein** définit la distance entre deux chaînes de caractères en comptant le nombre minimum d'opérations nécessaires pour transformer une chaîne à l'autre. Une opération est définie comme une insertion, une suppression, une substitution d'un seul caractère (opérations classiques de Levenshtein) ou une transposition de deux caractères adjacents (Bard, 2007).
- **Jaro** est basé sur le nombre et l'ordre des caractères communs entre deux chaînes de caractères (Jaro, 1989).
- **Jaccard** est calculé comme le nombre de termes partagés par rapport au nombre de tous les termes uniques dans les deux chaînes de caractères (Jaccard, 1901).

Finalement, l'avantage d'utiliser différentes mesures est d'agréger les résultats finaux, car les valeurs peuvent fortement varier en fonction des éléments mesurés.

3.1.2 Chaînes YouTube et outils d'extraction

Afin de mettre en œuvre cette analyse comparative, nous présentons au Tableau 3.1 les différents outils d'extraction qui seront utilisés. Par la suite, nous présenterons la référence aux scripts originaux des épisodes qui serviront de valeurs témoins au Tableau 3.2 et la liste de différentes chaînes YouTube offrant les épisodes de la première saison de *Peppa Pig* au

Tableau 3.3.

Tableau 3.1 Outils d'extraction automatique des sous-titres

Outil	Référence

YouTube-dl	https://github.com/ytdl-org/youtube-dl
PyTube	https://github.com/nficano/pytube
YouTube-transcript-api	https://github.com/jdepoix/youtube-transcript-api

D'un point de vue légal, l'utilisation d'outils qui permet l'extraction de vidéos ou de leurs sous-titres semble causer préjudice, car la RIAA (Recording Industry Association of America) qui est l'organisme américain responsable de la protection contre le piratage audiovisuel, a déposé le 23 octobre 2020 une demande de retirer tout code source en lien avec l'utilitaire *YouTube-dl* sur *GitHub*. A suivi ensuite un jeu du chat et de la souris dans lequel les partisans de ce logiciel ont commencé à créer d'innombrables copies du logiciel sous différents noms afin de le garder en ligne pour de multiples raisons dont son utilité dans le domaine du journalisme et de la recherche. Trois semaines plus tard, soit le 16 novembre 2020, *GitHub* a publiquement rétabli le dépôt officiel du logiciel en envoyant une contestation de l'avis stipulant que le logiciel ne violait pas les systèmes DRM (Digital Right Management) commerciaux.

N'étant pas juriste, mais consciencieux du droit d'auteur, nous croyons que les transcriptions sont des œuvres dérivées couvertes par le droit d'auteur même si elles ont été générées automatiquement et que nous sommes autorisés à les utiliser dans un contexte d'usage loyal[11] et sans but commercial. Cependant, demander accès ou poser toutes questions de ce genre génère des réponses complètement différentes en fonction de l'interlocuteur. Par exemple, l'accès aux sous-titres originaux anglais nous a été refusé de la part du diffuseur officiel (filmsseville.com) de cette série au Canada, car il se dit frileux quant à l'utilisation de ses marques (titre, dialogue, image) à l'extérieur d'un contexte commercial. Par conséquent, afin d'obtenir une valeur témoin, nous utilisons les données disponibles sur un *fandom* dédié à *Peppa Pig* dont l'URL est référencée dans le tableau ci-dessous. Il convient de noter que le

[11] https://support.google.com/youtube/answer/2797449?hl=fr#zippy=%2Cquentend-on-par-usage-loyal

terme *fandom* désigne une communauté d'adeptes partageant un même enthousiasme pour un sujet particulier, en l'occurrence ici *Peppa Pig*.

Tableau 3.2 Valeurs témoins

Nombre d'épisodes offerts pour la première saison de *Peppa Pig*	**Référence**
52	http://glamour-and-discourse.blogspot.com/p/peppa-pig-episode-transcripts.html

Étant liés à ces valeurs témoins, nous avons donc scruté la plateforme YouTube pour identifier différentes chaînes offrant les épisodes de la première saison de cette série pour enfant. Le tableau suivant liste ces différentes chaînes ainsi que le nombre d'épisodes qu'elles contiennent.

Tableau 3.3 Chaînes YouTube de la saison 1 de Peppa Pig

Chaîne YouTube	**Nombre épisodes**	**Lien de la chaîne YouTube**
PeppaPigSurprise	54	https://www.youtube.com/playlist?list=PLinVG_AbsqohsQ5sh1VHR5KjD0Qb0TWrx
PeppaPigSubtitled	52	https://www.youtube.com/playlist?list=PL9H5wl7jyq4RW3I86ECokGqzESs4SrY0Z
RexyKids	52	https://www.youtube.com/playlist?list=PLhBNaOObFw5kRNaYjuVVf2IdX2HU-Mruq
PeppaPigAsia	10	https://www.youtube.com/playlist?list=PLU00UZpxqtObvpONlPTvGE5Yv2lUFu9ha

Puisque ces chaînes sont détenues par de tierces personnes et que la gestion de leur contenu leur est propre, ces détenteurs ne sont en aucun cas tenus garants de la validité de l'appartenance des épisodes à une saison précise. Dans le tableau ci-dessus, on remarque un

manque de constance puisque le nombre d'épisodes offerts varie entre 10 et 54 alors que le nombre d'épisodes de cette saison est de 52[12].

Afin de vérifier si ces différentes chaînes offrent réellement les épisodes appartenant à la première saison de cette série pour enfant, nous validons les titres des épisodes avec une liste de référence créée à partir des informations offertes sur Wikipédia[13].

3.1.3 Méthodologie pour sélectionner la meilleure combinaison outil d'extraction et chaîne YouTube

Globalement, la méthodologie consiste à valider leur appartenance à la première saison de *Peppa Pig,* à collecter les données textuelles provenant de différentes sources sur le Web (Tableau 3.2 et

Tableau 3.3), à appliquer les prétraitements appropriés, à calculer le niveau de similarité entre la transcription à sa valeur témoin pour chaque épisode et finalement, à sélectionner la combinaison « outil d'extraction - chaîne YouTube » qui offre le plus haut niveau de similarité pour l'ensemble des épisodes.

De manière plus détaillée :

1. Validation des épisodes

   Cette étape de validation consiste, premièrement, à valider l'appartenance des épisodes listés dans les différentes chaînes aux épisodes réels de la première saison. Pour ce faire, nous comparons les titres des vidéos à la liste des épisodes offerte sur Wikipédia. Deuxièmement, pour chaque épisode de chaque chaîne, nous collectons les informations qui concernent la méthode de génération des sous-titres (automatique ou manuel) ainsi qu'à leur possibilité d'extraction à l'aide de l'utilitaire *YouTube TranscriptApi*.
2. L'extraction des données

---

[12] https://en.wikipedia.org/wiki/Peppa_Pig
[13] https://en.wikipedia.org/wiki/List_of_Peppa_Pig_episodes#Series_1_(2004)

L’extraction des sous-titres en provenance des différentes chaînes YouTube (voir

Tableau 3.3) s'effectue à l'aide des outils mentionnés au Tableau 3.1. Les données que nous qualifions comme valeur témoins proviennent d’un *fandom*, voir Tableau 3.2. Ces données étant mises en libre accès sur une page Web, leur extraction s’effectue à l’aide d’un outil de *Web scraper.*

3. Uniformisation et prétraitement des textes

Cette étape consiste, dans un premier temps, à extraire les données textuelles brutes. Pour ce faire, nous devons uniformiser les différentes sources de textes (transcription et valeur témoin) afin d'éliminer les éléments non textuels comme des balises HTML et d'horodatage. Ensuite, la notion de prétraitement fait référence aux étapes traditionnelles effectuées dans le domaine de la modélisation par sujet afin d'y extraire les termes pertinents dans le but de bâtir le vocabulaire. Le tableau suivant liste les différentes étapes dans leur ordre d'exécution.

Tableau 3.4 Étapes de prétraitement du corpus

Étape	**Technique**	**Librairie**
1	Mettre en jeton	*TreebankWordTokenizer* de *NLTK*
2	Mettre en minuscule	
3	Enlever la ponctuation	
4	Éliminer les nombres	
5	Éliminer les mots vides	*Stopwords* de *NLTK* et *Sklearn*
6	Éliminer les caractères uniques	
7	Appliquer la lemmatisation	*Spacy*

Notez qu'à l'étape 5 (éliminer les mots vides), deux librairies ont été combinées afin d'augmenter le nombre de termes considérés comme vides. La librairie *Sklearn* contient 318 termes comparativement à 179 pour *NLTK*. Cette dernière contient 60 termes non inclus dans *Sklearn*. En les combinant, nous obtenons une liste de 378 termes. De plus, une analyse des termes les plus fréquents a été appliquée afin de personnaliser la liste des mots vides du corpus à l'étude. Voici les termes qui y ont été ajoutés :

pron, oh, hmm, ah, ok, ca, hey, ha, wow, whoop, whoops, um, huh, uh, hi, ho, na, aah, mmhmm, mmm, uhhuh, ooh, ii, uhoh, bk, ok, mm, goo, haha, nt, aha, ahchoo, whoa, oops

4. Calculs des similarités entre les sous-titres et sa valeur témoin correspondante

   En utilisant les différents jeux de données issus de l'étape précédente, les calculs de similarité entre les différentes combinaisons « outil d'extraction - chaîne YouTube » et leur valeur témoin respective seront basés sur les trois mesures suivantes :

   - Damereau-Leveshtein (*character based*)
   - Jarro (character based)
   - Jaccard (*term based*)

5. Sélection finale de la combinaison « outil d'extraction - chaîne YouTube »

   La sélection de la combinaison offrant le plus haut niveau de similarité pour l'ensemble des épisodes sera détaillé lors de l'analyse des scores de similarité présenté ci-dessous (section 3.1.4.2).

3.1.4 Résultats et discussion

Dans cette section, nous allons présenter les résultats pertinents qui mènent à la sélection d'un outil d'extraction et d'une chaîne YouTube. Nous nous attarderons aux résultats obtenus aux étapes de validation des épisodes et de l'analyse des scores de similarité.

3.1.4.1 Validation des épisodes

Le tableau suivant dénombre pour les différentes chaînes YouTube sélectionnées : le nombre d'épisodes appartenant à la première saison de *Peppa Pig*, les numéros d'épisodes pour lesquels il est impossible d'extraire les sous-titres par programmation ainsi que le nombre d'épisodes pour lesquels les sous-titres ont été générés manuellement et automatiquement.

Tableau 3.5 Validation des épisodes

Chaîne YouTube	**Nombre d'épisodes appartenant à la première saison de *Peppa Pig***	**Numéro d'épisode sans sous-titres**	**Nombre d'épisodes avec sous-titres générés manuellement**	**Nombre d'épisodes avec sous-titres générés automatiquement**
PeppaPigSurprise	52	1,5,9	Aucun	49
PeppaPigSubtitled	52	1,5,9,14,48	Aucun	47
RexyKids	52	1,5,9,45	Aucun	48
PeppaPigAsia	10	Aucun	10	9

Comme nous pouvons le remarquer au Tableau 3.5 présenté ci-dessus, les chaînes YouTube *PeppaPigSurprise*, *PeppaPigSubtitled* et *RexyKids* offrent le visionnement des 52 épisodes appartenant à la première saison de *Peppa Pig*. De plus, aucune de ces différentes chaînes ne détient des épisodes pour lesquels les sous-titres ont été générés manuellement. Par conséquent, nous aurons accès aux sous-titres qui ont été générés à l'aide d'un algorithme de transcription automatique. Cependant, il est impossible d'extraire les sous-titres pour les épisodes 1, 5 et 9 de ces différentes chaînes. De plus, la chaîne *RexyKids* ne permet pas l'extraction des sous-titres de l'épisode 45 et la chaîne *PeppaPigSubtitled* ne permet pas l'extraction des sous-titres des épisodes 14 et 48.

En ce qui concerne la sélection de la chaîne *PeppaPigAsia*, son principal intérêt est qu'elle offre la possibilité d'extraire les sous-titres qui ont été générés manuellement et automatiquement. Même si cette chaîne ne détient que quelques épisodes, leur analyse permettra de vérifier si les transcriptions manuelles sont plus similaires à leur valeur témoin respective.

Au tableau suivant, nous présentons les différentes possibilités des combinaisons « outil d'extraction - chaîne YouTube ». Les cases marquées d'un X sont les combinaisons pour lesquelles il a été possible d'extraire les sous-titres et qui seront mises en comparaison pour l'analyse des scores de similarité.

Tableau 3.6 Possibilités de combinaisons
« outil d'extraction - chaîne YouTube »

Outil / **Chaîne**	*YouTube-dl*	*Pytube*	*YouTube Transcript Api (manual)*	*YouTube Transcript Api (automatic)*
PeppaPigAsia (automatique)	X			X
PeppaPigAsia (manuel)		X	X	
PeppaPigSubtitled (automatique)	X			X
PeppaPigSurprise (automatique)	X			X
RexyKids (automatique)	X			X

Par cette représentation, nous pouvons conclure que l'outil *Pytube* permet seulement d'extraire les sous-titres qui ont été générés manuellement en opposition à l'outil *YouTube-dl* qui permet seulement d'extraire les sous-titres générés automatiquement. Finalement, l'outil *YouTube-transcript* permet, selon l'API, d'extraire les sous-titres en fonction de l'API spécifié.

3.1.4.2 Analyse des scores de similarité

L'analyse des scores de similarité permet de sélectionner la combinaison « outil d'extraction - chaîne YouTube » gagnante, c'est-à-dire celle qui offre le plus haut niveau de similarité entre les sous-titres des différents épisodes à leur valeur témoin respective.
Comme préalablement mentionnés, nous avons sélectionné 3 différentes mesures de similarité, quatre chaînes YouTube et trois outils d'extractions différents. L'intérêt de calculer la similarité selon différentes mesures est de voir s'il y a une adéquation entre elles quant à l'identification de la combinaison gagnante. Les différentes combinaisons possibles permettant l'obtention des transcriptions à l'aide des différents outils sont présentées au Tableau 3.6 de la section précédente et identifient dix combinaisons possibles. Cependant, six combinaisons sont utilisées pour identifier la combinaison gagnante, car elles utilisent des chaînes offrant la majorité des épisodes de la première saison et que les sous-titres ont été générés automatiquement. Les quatre autres combinaisons utilisent la chaîne *PeppaPigAsia* qui ne détient que quelques épisodes de la première saison, mais son intérêt réside dans la possibilité d'extraire les sous-titres générés manuellement et automatiquement pour un même épisode. Cette possibilité de comparaison nous permet de confronter la qualité des sous-titres générée manuellement à ceux générés automatiquement ainsi que vérifier si les transcriptions manuelles sont plus similaires à la valeur témoin.

Afin de brosser un portrait général des résultats de similarité obtenus par les différentes combinaisons, nous présentons à la figure suivante la distribution des scores obtenus à l'aide d'une représentation de boîtes à moustache (*box plot*). Cette représentation est un moyen rapide de figurer le profil des résultats et est utile pour comparer des ensembles de résultats de tailles différentes. De plus, cette figure est constituée de trois lignes et trois colonnes où les lignes identifient les différentes mesures de similarité. La première colonne identifie les deux combinaisons utilisant la chaîne *PeppaPigAsia* où les sous-titres ont été générés manuellement. La seconde colonne identifie les deux autres combinaisons provenant de cette même chaîne, mais pour lesquels les sous-titres ont été générés automatiquement. La troisième colonne

identifie les six combinaisons où les sous-titres ont été générés automatiquement et qui contiennent la majorité des épisodes la première saison. À noter que nous avons également ajouté en trait pointillé la moyenne et l'écart-type afin de représenter l'intervalle « moyenne ± écart-type ».

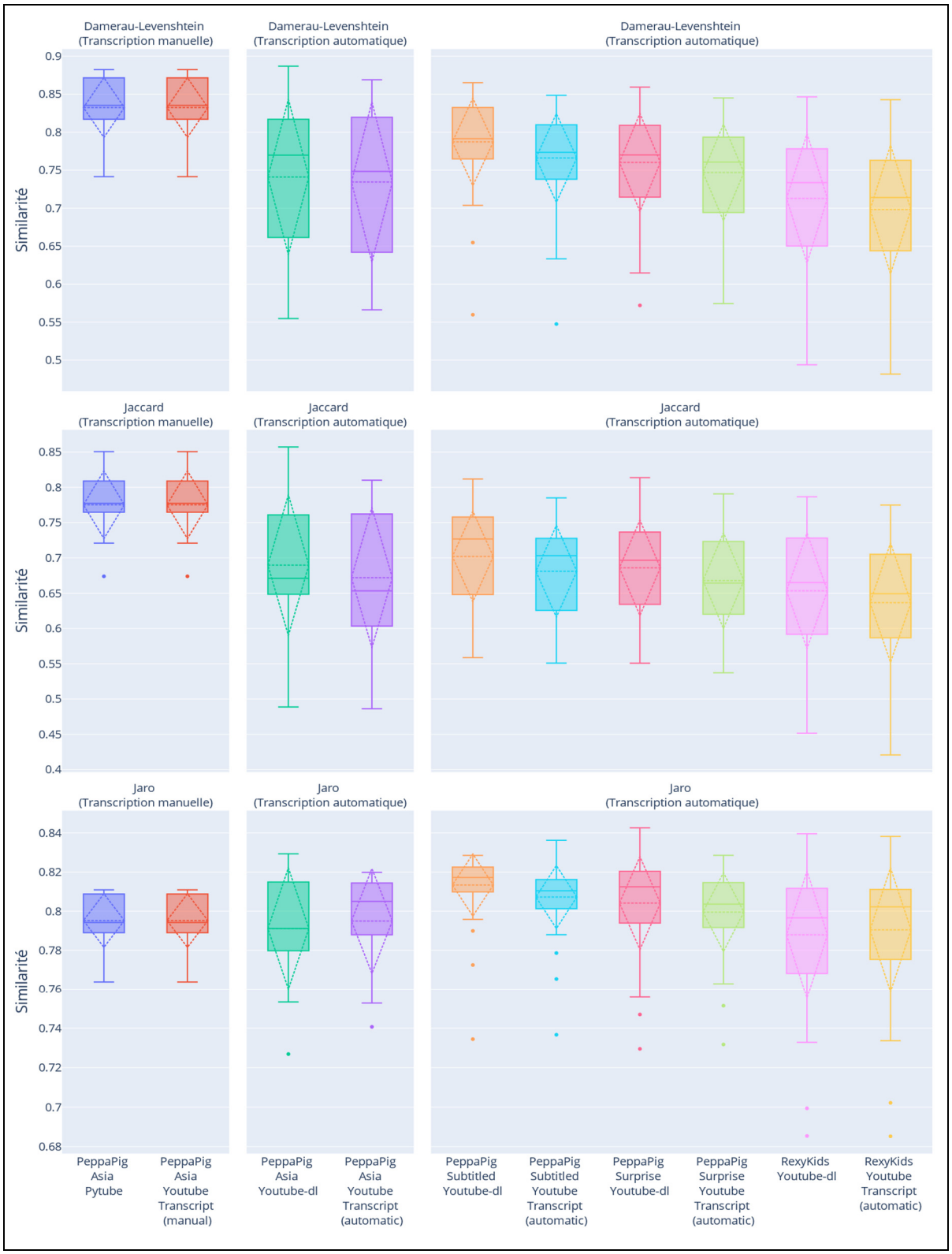

Damerau-Levenshtein (Transcription manuelle)
Damerau-Levenshtein (Transcription automatique)
Damerau-Levenshtein (Transcription automatique)
Jaccard (Transcription manuelle)
Jaccard (Transcription automatique)
Jaccard (Transcription automatique)
Jaro (Transcription manuelle)
Jaro (Transcription automatique)
Jaro (Transcription automatique)
Similarité
PeppaPig Asia Pytube
PeppaPig Asia Youtube Transcript (manual)
PeppaPig Asia Youtube-dl
PeppaPig Asia Youtube Transcript (automatic)
PeppaPig Subtitled Youtube-dl
PeppaPig Subtitled Youtube Transcript (automatic)
PeppaPig Surprise Youtube-dl
PeppaPig Surprise Youtube Transcript (automatic)
RexyKids Youtube-dl
RexyKids Youtube Transcript (automatic)

Figure 3.1 Distribution des scores de similarité obtenus pour les différentes combinaisons outil d'extraction et chaîne YouTube

Premièrement, comparons les résultats des deux premières colonnes afin de confronter la qualité des sous-titres générés manuellement (la première colonne) à ceux générés automatiquement (la deuxième colonne). Pour la première colonne, les différentes mesures de similarité accordent des scores moyens (voir traits pointillés) variant entre 0,79 et 0,84. De plus, les transcriptions obtenues par les outils *Pytube* et *Youtube-Transcript (manual)* obtiennent des distributions de scores identiques, cela nous permet donc de supposer que ces deux outils accèdent à la même source de sous-titres.

Pour la seconde colonne, les différentes mesures de similarité accordent des scores moyens variant entre 0,67 et 0,79. Cependant, les différentes mesures n'accordent pas des distributions de score de similarité identiques pour les outils *Youtube-dl* et *Youtube-Transcript (automatic)*. Selon les mesures de Damereau-Levenstein et Jaccard, ces mesures accordent des scores de similarités moyens et des quartiles 1 plus élevés avec l'utilisation de l'outil *Youtube-dl*. Selon la mesure de Jaro, celle-ci accorde un score moyen et un quartile 1 plus élevé avec l'utilisation de l'outil *Youtube-Transcript (automatic)*.

Si nous comparons les résultats obtenus entre ces deux colonnes, nous pouvons conclure que les sous-titres générés manuellement sont plus similaires à la valeur témoin que ceux générés par un algorithme de transcription automatique. Si tel n'avait pas été le cas, nous aurions mis en doute la qualité des transcriptions produites manuellement et peut-être même la validité des valeurs témoins.

Deuxièmement, regardons les résultats de la troisième colonne. En s'attardant au score moyen, nous pouvons distinguer que les scores de similarité accordés par la mesure de Jaccard évoluent entre 0,64 et 0,70. Selon la mesure de Damerau-Levenshtein, les scores de similarité évoluent entre 0,70 et 0,78. Selon la mesure de Jaro, les scores de similarité évoluent entre 0,79 et 0,81. Nous pouvons également identifier une tendance baissière de ces scores moyens pour toutes les mesures confondues à partir de la combinaison la plus à gauche (*PeppaPigSubtitled-Youtube-dl*) jusqu'à l'extrémité droite (*RexyKids- Youtube-Transcript (automatic)*).

De plus, un élément contre-intuitif qu'il possible de remarquer est la distribution de résultats obtenus entre la colonne 1 (sous-titres générés manuellement) et la colonne 3 (sous-titres générés automatiquement) pour la mesure de Jaro. En opposition aux deux autres mesures, Jaro accorde des scores de similarité égaux et supérieurs aux transcriptions qui ont été générées automatiquement.

Finalement, on peut identifier la chaîne *Peppa Pig Subtitled* et l'outil *YouTube-dl* comme potentiel gagnant dû à ses valeurs du premier quartile et moyennes plus élevées accordées par les différentes mesures de similarité. On peut également confirmer que pour une chaîne YouTube, les différents outils utilisés ne donnent pas la même qualité de sous-titres.

Cette première représentation nous a permis d'illustrer le profil général des scores de similarité et d'identifier une combinaison potentiellement gagnante. Afin de confirmer notre sélection, nous effectuons une seconde analyse. Celle-ci consiste à identifier parmi les différentes combinaisons outil-chaîne les trois résultats de similarité les plus élevés pour un même épisode. Pour chaque épisode, à la suite de l'identification de ce top trois, nous attribuons à la combinaison ayant le score le plus élevé la mention « 1 », à la combinaison ayant le second score plus élevé la mention « 2 », à la combinaison ayant le troisième score le plus élevé la mention « 3 » et la mention « valeur inférieure » aux autres combinaisons qui ont des scores ne figurant pas dans ce top trois. À noter que si des combinaisons partagent le même score de similarité, ces combinaisons se verront accorder la même mention (par exemple, la mention « 1 »). De plus, s'il n'y a pas de résultat de similarité pour un épisode, dû au fait que les sous-titres de cet épisode ne sont pas accessibles pour la combinaison outil-chaîne, cet épisode se verra attribuer la mention « Épisode-absent ».

Après l'application de cette méthode de classement sur l'ensemble des épisodes de la première saison, on dénombrera ces différentes mentions afin de les cumuler. C'est à l'aide du cumul de

ces différentes mentions que nous allons faire notre choix final. Par exemple, nous pourrions sélectionner la combinaison qui cumule le plus de mentions « 1 ».

Illustrons cette démarche à l'aide d'un exemple où nous avons cinq épisodes et quatre combinaisons outil-chaînes avec lesquelles il a été possible d'extraire les transcriptions pour chacun de ces épisodes. Nous calculons ensuite le score de similarité entre la valeur témoin les transcriptions avec de la mesure de Jaccard. Nous identifions ensuite les trois résultats de similarité les plus élevés parmi les 4 combinaisons outil-chaînes pour chacun des épisodes. Ces résultats sont présentés au tableau suivant.

Tableau 3.7 Exemple d'affectation des mentions

Décompte des mentions et valeur du score de similarité par épisode		**Combinaison Chaîne Youtube-outil d'extraction**			
		PeppaPigAsia Youtube-dl	*PeppaPigSubtitle Youtube-dl*	*PeppaPigSubtitle Pytube*	*RexyKids Youtube-dl*
Épisode 1	Score	0,90	0,60	0,65	0,30
	Mention	1	3	2	Valeur Inférieure
Épisode 2	Score	0,92	0,88	0,89	0,91
	Mention	1	Valeur Inférieure	3	2
Épisode 3	Score	0,50	0,90	0,70	0,65
	Mention	Valeur Inférieure	1	2	3
Épisode 4	Score	0,65	0,55	0,50	0,70
	Mention	2	3	Valeur Inférieure	1

Épisode 5	Score	0,80	0,65	0,60	0,50
	Mention	1	2	3	Valeur Inférieure

En cumulant les différentes mentions pour chacune des combinaisons outil-chaînes, nous pouvons résumer ces résultats à l'aide du tableau suivant.

Tableau 3.8 Exemple du cumul des mentions

Combinaison « outil d'extraction - chaîne YouTube »	**Cumul des mentions**				
	1	**2**	**3**	**Valeur Inférieure**	**Épisode absent**
PeppaPigAsia-Youtube-dl	**3**	**1**	**0**	**1**	**0**
PeppaPigSubtitle-Youtube-dl	1	1	2	1	0
PeppaPigSubtitle-Pytube	0	2	2	1	0
RexyKids-Youtube-dl	1	1	1	2	0

À la suite du dénombrement des différentes mentions, nous pouvons identifier la combinaison *PeppaPigAsia-Youtube-dl* comme étant, en général et selon la mesure de Jaccard, la plus fidèle aux valeurs témoins, car c'est elle qui cumule le plus de mentions « 1 ».

Cependant, cette démarche nous permet d'identifier une combinaison par mesure de similarité. Nous devrons donc identifier la meilleure combinaison parmi les différentes mesures de similarité. Une façon de représenter visuellement des cumuls de résultats est d'utiliser les diagrammes à bandes. À la figure suivante, nous présentons les résultats pour l'ensemble des épisodes sous le même format précédemment illustré à la Figure 3.1 (trois lignes et trois colonnes). La couleur or correspond à la mention « 1 », la couleur argent correspond à la mention « 2 », la couleur bronze correspond à la mention « 3 », la couleur rose correspond à la mention « Valeur inférieure » et la couleur bleue correspond à la mention « Épisode absent ».

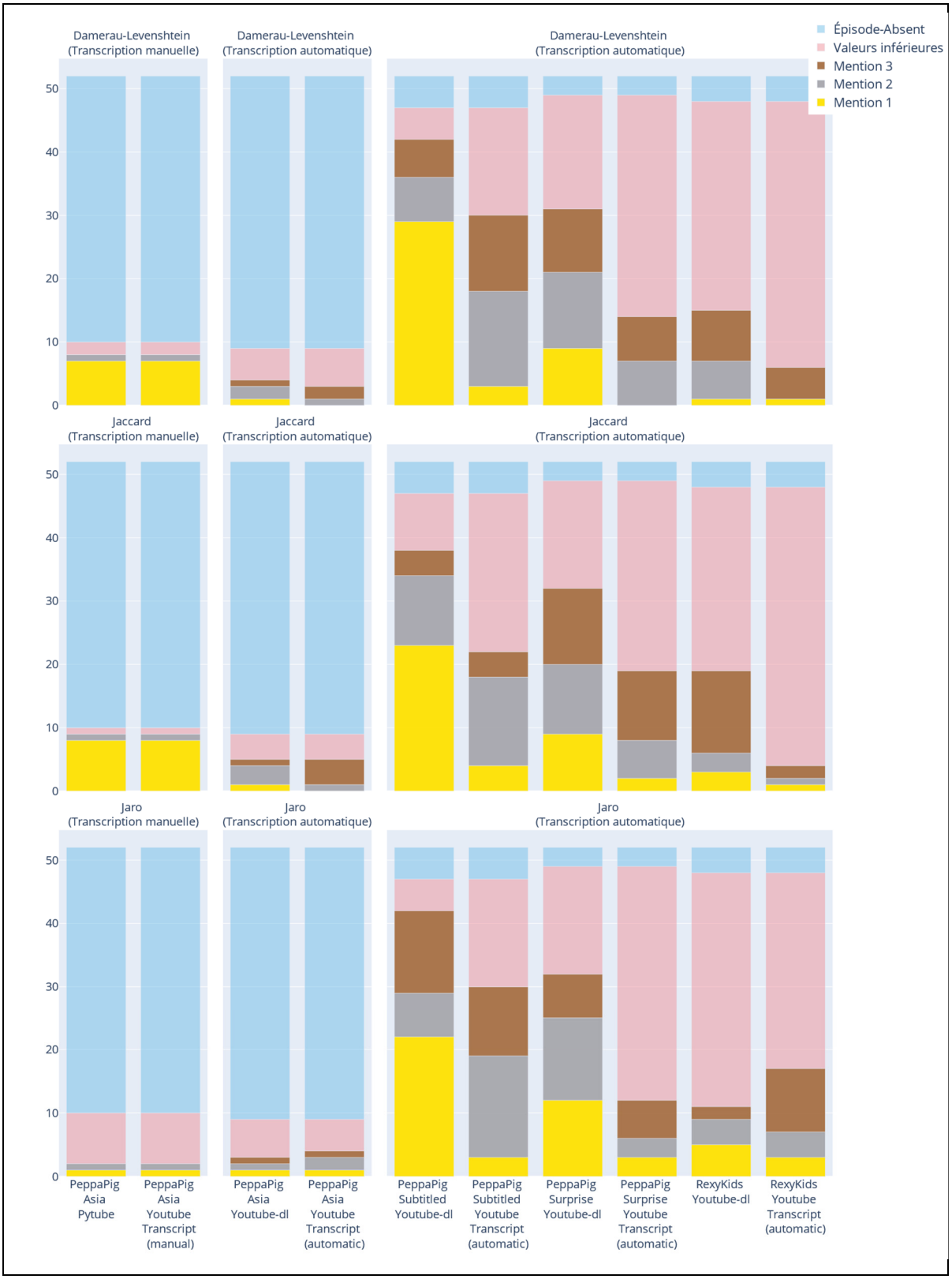

Damerau-Levenshtein (Transcription manuelle)
Damerau-Levenshtein (Transcription automatique)
Damerau-Levenshtein (Transcription automatique)
Jaccard (Transcription manuelle)
Jaccard (Transcription automatique)
Jaccard (Transcription automatique)
Jaro (Transcription manuelle)
Jaro (Transcription automatique)
Jaro (Transcription automatique)
Épisode-Absent
Valeurs inférieures
Mention 3
Mention 2
Mention 1
0
10
20
30
40
50
PeppaPig Asia Pytube
PeppaPig Asia Youtube Transcript (manual)
PeppaPig Asia Youtube-dl
PeppaPig Asia Youtube Transcript (automatic)
PeppaPig Subtitled Youtube-dl
PeppaPig Subtitled Youtube Transcript (automatic)
PeppaPig Surprise Youtube-dl
PeppaPig Surprise Youtube Transcript (automatic)
RexyKids Youtube-dl
RexyKids Youtube Transcript (automatic)

Figure 3.2 Cumul des mentions

Cette méthode de comparaison met en compétition les transcriptions obtenues par épisode en fonction des différentes combinaisons outil d'extraction et chaîne YouTube. Nous pouvons remarquer, en regardant les résultats de la colonne de gauche, que ce ne sont pas toutes les transcriptions manuelles qui obtiennent la mention « 1 » et que, selon la mesure de similarité utilisée, le cumul des différentes mentions varie. Pour ce qui a trait aux sous-titres qui ont été générés par un algorithme de transcription automatique, les résultats de la colonne de droite, les différentes mesures de similarité s'accordent à identifier la chaîne *PeppaPigSubtitled* et l'outil *YouTube-dl* comme la combinaison qui cumule le plus de mentions « 1 », « 2 » et « 3 ». Nous identifions donc cette combinaison comme étant la combinaison gagnante.

3.1.5 Réponse

Finalement, le but de cette première étape était de sélectionner une chaîne YouTube et un outil d'extraction automatique des sous-titres afin d'obtenir les sous-titres les plus fidèles à la valeur témoin. Cette démarche nous a permis de sélectionner la chaîne *PeppaPig Subtitled* et l'outil *YouTube-dl* qui nous permet de construire un corpus de 47 épisodes afin de poursuivre aux prochaines étapes.

3.2 Question 2 : Quelles sont les durées minimales d'une séquence vidéo permettant de générer un document contenant une valeur minimale de termes significatifs ?

Les algorithmes appartenant au domaine de la modélisation par sujet sont appliqués sur un ensemble de documents. Un document peut représenter un paragraphe, une page, un chapitre ou même l'entièreté d'un livre. Si nous nous attardons aux nombres de termes qui constituent chacun de ces documents, nous pouvons aisément déduire qu'il diffère grandement. Nous définissons donc la longueur d'un document (LD) comme étant le décompte du nombre de termes d'un document.

L'objectif de cette étape est d'identifier des valeurs candidates de LD issu de la segmentation des épisodes sur des intervalles de temps relativement courts. Cette étape est nécessaire afin d'être en mesure d'émettre des recommandations sur de courtes séquences vidéo et puisque, d'un point de vue modélisation de sujet, des documents de courte longueur devraient favoriser la probabilité d'appartenance d'un document à un seul sujet.

Pour ce faire, nous évaluons sous la forme d'une analyse quantitative, l'impact de faire varier le nombre de répliques sur la LD. Cette évaluation nous aide à sélectionner quelques valeurs candidates de LD (trois) que nous utilisons pour la création des sac-de-mots.

Les sections suivantes présentent le type de format des transcriptions automatiques, la méthodologie générale qui permet de générer les valeurs candidates de LD et les résultats statistiques décrivant ces différentes transformations suivies d'une discussion qui nous permettra de répondre à la question qui identifie cette section.

3.2.1 Format des transcriptions

La génération des valeurs candidates de LD est tributaire de la structure du fichier de transcription. Le format privilégié dans cette étude est le WebVTT (*Web Video Text Tracks*). Ce type de fichier contient des informations supplémentaires sur une vidéo telle que, des sous-titres, des descriptions, des métadonnées, mais ne contient pas de données vidéo. L'unité contenant les informations relatives aux sous-titres est communément nommée la « *cue* » dans ce type d'encodage, mais nous le traduisons par le terme « réplique » en faisant allusion aux échanges entre les acteurs dans une pièce de théâtre. Le tableau suivant illustre le format général de cette unité.

Tableau 3.9 Patron général d'une réplique

Représentation	Description

[idstring] [hh:]mm:ss.msmsms -->[hh:]mm:ss.msmsms Texte (la réplique)	Identifiant de la réplique Temps début --> Temps Fin Transcription

Ce type de représentation permet d'obtenir le moment auquel une communication orale a eu lieu dans la vidéo. Nous allons donc baser la construction d'un document sur de multiples enchaînements de répliques. Il est important de mentionner que la durée des répliques n'est pas constante et que les périodes sans paroles ne sont pas transcrites.

3.2.2 Méthodologie

Les détails de la méthodologie sont présentés ci-dessous et se scindent en trois étapes distinctes :

1. Sélection du corpus

   Le corpus utilisé est celui issu de l'étape d'identification de « outil d'extraction - chaîne YouTube » offrant la transcription la plus fidèle à la valeur témoin, soit : « *YouTube-dl - PeppaPig Subtitled* ». Ce corpus est constitué de 47 épisodes.

2. Génération de document avec et sans prétraitement sur le corpus

   Pour les fins d'analyse de notre corpus, nous générons les documents avec et sans l'application des étapes de prétraitements mentionnées au Tableau 3.4

3. Calcul pour la génération de document en variant la définition de la longueur d'un document

   Puisque la durée et la quantité de termes constituant une réplique ne sont pas constantes, nous avons développé une approche utilisant la réplique comme unité de base pour créer différentes longueurs de document. L'approche, dans sa plus simple expression, consiste à enchaîner des répliques en respectant leur ordre chronologique pour lesquelles nous ajoutons la contrainte qu'un document doit contenir un minimum de cinq termes. Dans le

cas où un document contient moins de cinq termes, il est automatiquement ajouté au document suivant.

Pour les fins de cette analyse, nous avons fait varier le nombre de répliques à enchaîner pour constituer un document. Les différents nombres de répliques à enchaîner sont représentés dans la liste ci-dessous où la valeur 1 signifie qu'un document est constitué d'une réplique et 9999 est l'enchaînement de toutes les répliques contenues dans l'épisode afin de produire 1 document.

1,4,5,7,9,10,11,12,13,14,15,20,25,30,9999

3.2.3 Résultats et discussion

Comme précédemment discuté, le nombre de répliques à enchaîner pour construire un document a un impact sur le nombre de termes qui constitue un document (LD) et sur la durée de la séquence vidéo. Dans cette section, nous décrivons cet impact à l'aide de quelques statistiques que nous rapportons aux tableaux suivants. Ces statistiques concernent le nombre total de documents que ce découpage a permis de générer sur l'ensemble des répliques constituant le corpus, la durée de la séquence vidéo et le nombre de termes moyens par document, le débit de mot moyen et le nombre total de termes constituant le corpus.

Tableau 3.10 Génération de longueur de document sans l'application de prétraitement

Nombre de répliques enchaînées pour concevoir un document	**Nombre total de documents**	**Durée moyenne de la séquence vidéo (seconde)**	**Nombre moyen de termes par document**	**Débit de mot moyen (mots/seconde)**	**Total de terme dans le corpus**
1	2466	3,6 ± 3.12	7,86 ± 1.53	2,18	19393
4	699	16,85 ± 7.49	27,74 ± 4.81	1,65	19393
5	561	21,46 ± 8.45	34,57 ± 6.01	1,61	19393
7	408	30,01 ± 10.53	47,53 ± 9.30	1,58	19393
9	326	38,21 ± 14.06	59,49 ± 14.02	1,56	19393
10	290	42,91 ± 13.77	66,87 ± 13.69	1,56	19393
11	268	46,72 ± 16.62	72,36 ± 17.58	1,55	19393
12	245	51,34 ± 16.95	79,16 ± 17.54	1,54	19393
13	233	53,85 ± 19.44	83,23 ± 22.79	1,55	19393
14	215	58,47 ± 19.74	90,20 ± 24.08	1,54	19393
15	202	62,63 ± 21.06	96,00 ± 25.47	1,53	19393
20	157	80,80 ± 28.70	123,52 ± 37.68	1,53	19393
25	132	96,50 ± 37.12	146,92 ± 51.35	1,52	19393
30	112	114,06 ± 47.35	173,15 ± 66.54	1,52	19393
9999	47	27,23 ± 7.30	412,62 ± 71.04	1,50	19393

Tableau 3.11 Génération de longueur de document avec l'application de prétraitement

Nombre de répliques enchaînées pour concevoir un document	**Nombre total de documents**	**Durée moyenne de la séquence vidéo (seconde)**	**Nombre moyen de termes par document**	**Débit de mot moyen (mots/seconde)**	**Total de terme dans le corpus**
1	1367	7,75 ± 5.10	6,46 ± 1.33	0,83	8828
4	689	17,11 ± 7.47	12,81 ± 3.00	0,75	8828
5	553	21,79 ± 8.21	15,96 ± 3.5	0,73	8828
7	**403**	**30,5 ± 10.85**	**21,91 ± 4.99**	**0,72**	**8828**
9	**320**	**38,95 ± 13.73**	**27,59 ± 6.52**	**0,71**	**8828**
10	**287**	**43,38 ± 13.17**	**30,76 ± 6.96**	**0,71**	**8828**
11	264	47,44 ± 15.95	33,44 ± 8.25	0,70	8828
12	242	52,00 ± 16.57	36,48 ± 8.42	0,70	8828
13	230	54,57 ± 18.62	38,39 ± 10.69	0,70	8828
14	212	59,53 ± 19.65	41,64 ± 11.19	0,70	8828
15	201	62,95 ± 20.67	43,92 ± 12.27	0,70	8828
20	156	81,33 ± 28.08	56,59 ± 17.67	0,70	8828
25	131	97,24 ± 36.36	67,39 ± 23.79	0,69	8828
30	111	115,10 ± 46.33	79,53 ± 30.33	0,69	8828
9999	47	274,23 ± 7.30	187,83 ± 30.43	0,68	8828

Les statistiques rapportées dans ces deux tableaux diffèrent par le fait que nous avons appliqué ou non les étapes de prétraitements de texte afin de concevoir les documents. L'idée de comparer ces résultats est qu'il nous permet d'identifier quelques caractéristiques sur le corpus à l'étude ainsi que des impacts de notre méthode de création de documents.

Dans cet ordre d'idée, nous pouvons remarquer que l'application du prétraitement fait diminuer d'environ de moitié le nombre total de termes dans le corpus. Cela signifie qu'environ 1 mot sur 2 appartient à la liste des mots vides.

Si on s'attarde au débit de mots, nous pouvons remarquer que plus le nombre de répliques enchaînées augmente, plus le débit diminue. Cette variation s'explique par le fait que plus on enchaîne des répliques, plus on cumule des périodes sans paroles.

Une autre conséquence notable concernant notre méthode pour concevoir un document est que plus on diminue le nombre de répliques à enchaîner, plus on perdra des séquences visuelles sans paroles.

Cela dit, afin d'identifier des valeurs candidates du nombre de répliques enchaînées pour concevoir un document, nous considérons deux statistiques qui sont pour nous importantes à cause du contexte de recommandations vidéo. La première statistique concerne la durée d'une séquence vidéo et la seconde concerne le nombre de mots par document. Il est évident que ces deux statistiques sont dépendantes l'une de l'autre à cause de notre mode de génération de ces documents, mais nous voulons des séquences vidéo ni trop longues, ni trop courtes. Selon les résultats présentés au tableau 3.11, nous identifions qu'une durée moyenne idéale se situe entre 30 et 45. Cette plage de durée nous permettrait non seulement de concevoir un répertoire de vidéos plus volumineux, mais aussi de permettre aux utilisateurs de saisir le contexte de la vidéo lors des tests. Par exemple, si nous optons pour une durée moyenne de 30 secondes, nous disposerons d'un répertoire de 403 vidéos, tandis que pour une durée moyenne de 45 secondes, nous en aurons 287 à notre disposition.

Pour la seconde statistique, nous nous sommes questionnés quant à l'applicabilité de LDA sur des documents contenant peu de termes. C'est-à-dire que pour le bon fonctionnement de cet algorithme, un document doit contenir assez de mots pour qu'il puisse capturer le contexte discuté dans la séquence vidéo. En revanche, si les documents contiennent trop de termes, cela diminue les possibilités de recommandations. Par exemple, depuis l'émergence des médias sociaux, les textes courts sont devenus une source populaire d'information sur lesquels plusieurs recherches ont tenté d'appliquer des techniques de modélisation par sujet, dont LDA. Normalement, ces techniques sont appliquées sur des textes plus longs. Ces recherches ont identifié que l'application de ces techniques performe moins bien sur ce type de texte en raison du manque d'information (de contexte) sur les cooccurrences de mots dans les courts textes (Cheng et al., 2014; Lin et al., 2014). Afin de pallier ce manque de contexte, certaines recherches ont proposé de tirer avantage des mots-dièses (*hashtags*) auxquels ces textes se rapportent (Mehrotra et al., 2013) ou même de créer de nouveaux modèles spécialisés sur des documents contenant peu de termes (Cheng et al., 2014).

Dans l'étude de Qiang *et al.* (2020), ces auteurs ont évalué la performance de plusieurs modèles (9) dont LDA sur plusieurs jeux de données (six) couramment utilisés dans cette orientation de recherche. L'évaluation des performances consiste à évaluer la distribution document-sujets dans un contexte de classification ainsi que d'évaluer la distribution sujet-termes à l'aide de la mesure de cohérence proposée par Newman (Newman et al., 2010). En ce qui concerne les jeux de données, le nombre de termes moyens par document, après l'application des étapes de prétraitement, se situe entre 6 et 15. Les résultats obtenus pour l'évaluation des performances (classification et cohérence) n'admettent pas une sous-performance de LDA comparativement aux autres modèles. Cependant, il est notable que LDA performe moins bien sur les jeux de données où le nombre de termes moyens est de six ou sept comparativement aux jeux de données où le nombre de termes moyens est plus élevé.

En vertu de ces informations, nous établissons comme conditions de sélection qu'un document doit comporter au plus de 15 termes après l'étape de prétraitement et que la durée moyenne d'une séquence vidéo doit être comprise entre 30 et 45 secondes. Ainsi, nous sommes en mesure de prendre en compte les lignes qui présentent un nombre de répliques enchaînées allant de 7 à 10, tel que présenté au tableau 3.11.

3.2.4 Réponse

Finalement, le but de cette étape était d'identifier des durées minimales pour concevoir des documents basés sur l'enchaînement de répliques. Selon l'analyse de nos résultats et les contraintes que nous nous sommes imposées, nous avons sélectionné les nombres de répliques enchainées 7, 9 et 10.

3.3 Question 3 : Devons-nous considérer l'ajout d'un prétraitement spécialisé sur le nom des personnages ?

L'idée d'appliquer un traitement particulier aux termes représentant les personnages découle du constat que nous avons fait quant à la nomenclature employée par l'auteur pour les nommer. Prenons pour exemple la liste des représentants de la famille « Pig » :

- Peppa Pig
- George Pig
- Daddy Pig
- Mummy Pig
- Granny Pig
- Grandpa Pig
- Uncle Pig
- Auntie Pig
- Chloé Pig

Nous pouvons remarquer que chaque prénom est suivi de l'espèce animale et que certains prénoms réfèrent à la position hiérarchique familiale. Ces observations quant à la façon de nommer les personnages s'appliquent aussi aux autres espèces animales (lapin, chien, mouton, etc.) illustrées dans la télésérie[14].

Ce type de répétition peut s'expliquer par le fait que cette série est dédiée à un public de jeune âge et que ce type de mise en évidence favorise un certain caractère éducatif. Cependant, nous devons considérer que cette accentuation impactera l'algorithme de modélisation.

Toujours avec l'exemple de la famille « *Pig* », nous avons calculé la fréquence d'apparition des noms de ces personnages au sein de notre corpus. Nous présentons au Tableau 3.12, les fréquences des bigrammes associés aux représentants de cette famille, exemple « *Grandpa Pig* » et au Tableau 3.13, les fréquences des termes uniques (unigramme) qui ont permis de composer le nom de ces personnages, exemple « *Grandpa*» et « *Pig* ». À noter que les termes utilisés comme prénom peuvent être utilisés avec d'autres espèces animales, par exemple « *Grandpa Dog* », ce qui augmente la fréquence d'un terme comme « *Grandpa* ». Nous présentons également au Tableau 3.14 le nombre de termes uniques et le total de termes contenus dans le corpus à l'étude.

[14] https://peppapig.fandom.com/wiki/List_of_Characters

Tableau 3.12 Fréquence des bigrammes constituant le nom des personnages de la famille « Pig »

Personnage (bigramme)	**Fréquence**
Daddy Pig	147
Mummy Pig	112
Peppa Pig	41
Grandpa Pig	29
Granny Pig	12
Uncle Pig	11
George Pig	0
Auntie Pig	3
Chloé Pig	1
Total de terme	**Somme des fréquences**
9	356

Tableau 3.13 Fréquence des unigrammes utilisée pour identifier les représentants de la famille « Pig »

Personnage (unigramme)	**Fréquence**
George	445
Pig	400
Peppa	384
Daddy	322
Mummy	211
Grandpa	48
Auntie	4
Chloé	19
Uncle	15
Granny	30
Total de terme	**Somme des fréquences**
10	1878

Tableau 3.14 Statistiques sur le vocabulaire

Total de termes uniques	**Total de termes contenu dans le corpus**
1182	8828

À l'aide des fréquences calculées, nous remarquons que :

- Les fréquences des unigrammes utilisées pour identifier les représentants de la famille « *Pig* » représentent environ 21% (1878 / 8828 = 0,213) des fréquences des termes contenus dans le corpus.

- L'utilisation du terme « *Pig* » équivaut à 4,5% (400 / 8828) des fréquences des termes contenus dans le corpus.
- Le terme « *Pig* » représente 0,084% (1 / 1184) du vocabulaire.
- Environ 9 fois sur 10 (356/400) l'utilisation du terme « *Pig* » est employée pour composer le nom d'un représentant de cette famille

Du point de vue de modélisation par sujet, la notion de fréquence des termes dans un corpus a un impact sur les distributions finales. Les termes ayant des fréquences anormalement élevées comparativement aux autres termes qui constituent le vocabulaire auront tendance à être dominants dans plusieurs sujets latents. À titre d'exemple, dans les étapes du prétraitement traditionnel d'un corpus, nous considérons la suppression des termes appartenant à une liste de mots vides. Dans cette quête à éliminer les termes aux prédominances extrêmes (élevée ou faible), nous pouvons introduire la notion d'émondage relatif ou le *relative pruning* en anglais. En d'autres mots, cela signifie de supprimer les termes ayant des fréquences élevées et faibles. Il est recommandé de supprimer tous les termes qui apparaissent dans plus de 99% ou moins de 0,5% de tous les documents (Maier et al., 2018). Cette étape a pour avantage de réduire la taille du vocabulaire et aura pour conséquence d'améliorer les performances et de stabiliser l'inférence stochastique de LDA (Denny et Spirling, 2018).

Considérant la fréquence des termes uniques utilisés pour désigner les représentants de la famille « *Pig* » et le fait qu'il y a plus d'une centaine de personnages, nous pouvons aisément assumer qu'une importante portion du vocabulaire (plus de 21%) serait utilisée pour identifier un personnage. Influencés par ce constat et la considération de l'émondage relatif, cela nous a menées à nous questionner sur les avantages et les désavantages d'appliquer un traitement particulier sur les noms des personnages.

Dans notre contexte, supprimer le prénom d'un personnage comme « *Peppa* » nous ferait perdre toutes relations avec les sujets latents pouvant lui être associés. À l'inverse, ce terme risque d'être dominant dans plusieurs sujets latents. Peu importe les combinaisons que nous

pouvons imaginer, ce type de modification sur le corpus impactera la modélisation finale. Par ces considérations, nous proposons donc de comparer l'impact de l'application ou non de différentes techniques de substitution sur la modélisation du corpus. Pour ce faire, nous proposons cinq différentes techniques de substitution présentées au tableau suivant.

Tableau 3.15 Technique de substitution des personnages dans le corpus

Identifiant de la technique de substitution	**Technique de substitution**	**Exemple**
Ts-1	Aucune modification	"suzy sheep" : "suzy sheep"
Ts-2	Personnage vers espèce animal	"danny dog" : "dog"
Ts-3	Suppression du nom du personnage	"peppa pig" : ""
Ts-4	Concaténation	"rebecca rabbit" : "rebecca_rabbit_character"
Ts-5	Nom unique pour tous les personnages	"suzy sheep" : "peppa_pig_character", "rebecca rabbit" : "peppa_pig_character"

3.4 Conclusion

Ce chapitre a permis d'élucider notre démarche afin de générer le corpus de sous-titres qui est nécessaire à la poursuite de notre projet sous la forme de trois questions. Dans un premier temps, nous avons cherché à identifier la combinaison « outil-chaine YouTube » offrant les sous-titres les plus fidèles à notre valeur témoin. Selon notre démarche et les résultats obtenus, nous avons identifié la chaîne *PeppaPig Subtitled* et l'outil *YouTube-dl*.

Nous avons ensuite cherché à identifier des intervalles de temps qui permettront de segmenter un épisode en plusieurs courtes séquences. Nous avons identifié que des séquences comprises entre 30 à 45 secondes et qui contiennent en moyenne entre 22 et 31 termes significatifs nous sembleraient raisonnables afin de faire émerger la dominance d'un sujet latent par séquence vidéo (voir tableau 3.11).

À la suite d'une écoute attentive de cette série pour enfant, nous avons remarqué que l'auteur utilise une nomenclature particulière pour nommer les personnages de la série qui favorise la répétition de termes liés aux animaux. Afin de minimiser les perturbations que ces fréquences élevées de répétition de termes auraient sur l'algorithme de modélisation par sujet, nous avons proposé 5 techniques de substitution des noms des personnages.

Finalement, cette démarche nous a permis d'identifier trois définitions de longueurs de document différentes et cinq techniques de substitution des noms de personnages. Au total, 15 sac-de-mots différents seront analysés afin d'identifier la sélection du modèle final.

CHAPITRE 4

NEW HEURISTICS FOR STABLE LDA PARAMETER SEARCH

Simon-Olivier Harel [a], Erick Velazquez-Godinez [b], Sylvie Ratté [c]

[a,c] Département de Génie logiciel, École de Technologie Supérieure, 1100 Notre-Dame Ouest, Montréal, Québec, Canada, H3C 1K3
[b] Centre de recherche en informatique de Montréal (CRIM), 405 Av. Ogilvy 101, Montréal, Québec, Canada, H3N 1M3

Article soumis pour publication à la revue *Journal of Machine Learning Research*, mars 2023

4.1 Contexte de l'article dans ce projet de recherche

Ce chapitre présente la démarche qui permet d'atteindre le deuxième sous-objectif de ce projet de recherche qui concerne le développement d'une approche de recherche automatique des paramètres quasi optimaux de LDA et d'en vérifier sa stabilité. Ce sous-objectif fait partie de la seconde composante de notre système de recommandations présenté à la Figure 2.2 et son accomplissement permettra de modéliser les sous-titres.

4.2 Abstact

The Latent Dirichlet Allocation (LDA) algorithm automatically extracts latent topics from a textual corpus, but configuring its parameters can be difficult and time-consuming. Optimization algorithms can help determine the best parameters but not necessarily the optimal ones. This research proposes a framework that estimates near-optimal parameters for the corpus under analysis and provides a way to justify the final model selection. To achieve this goal, we evaluated different combinations of fitness functions (traditional and novel) to guide the Genetic Algorithms (GA) to identify the quasi-optimal LDA parameters. A stability

analysis is conducted to evaluate the quality of the parameters. The results show that different GA and fitness function combinations can identify the expected number of latent topics and extract more distinct topics than traditional ones. However, using only the combinations of different GA and fitness functions does not automatically produce the most stable results. We show that, instead of blindly relying on metaheuristic procedures, a thorough stability analysis is necessary to evaluate the quality of the parameters and obtain an optimal model. Finally, the choice of the fitness function should be based on the parameter tuning runtime and its ability to discover latent topics.

Keywords: Topic modeling, Latent Dirichlet allocation, Parameter optimization, Stability, Replication

4.3 Introduction

Making sense of or finding information in unstructured data, such as textual data, is difficult and challenging (Rajaraman, 2016). Finding, categorizing, and classifying the information meaningfully remains difficult as long as no structure is imposed. To see information emerge from textual data, we can rely on Natural Language Processing (NLP) techniques such as topic modeling to achieve this task. One of the common techniques for finding related topics within texts is the Latent Dirichlet Allocation (LDA) algorithm (Blei, Ng et Jordan, 2003). Even in this era of deep learning techniques, the use of this probabilistic model remains relevant in the context of statistical analysis of a corpus because of its interpretability and grounded generative process (Boyd-Graber, Hu et Mimno, 2017).

However, the general problem when using LDA is selecting the values of the parameters [K, α, β, N] when applying it to a specific corpus in a situation where no universal parameterization exists (Agrawal, Fu et Menzies, 2018; Hughes, Kim et Sudderth, 2015; Panichella, 2021). Many heuristics have been proposed to find the nearest optimal parameters for a given task. However, most of them focus on the number of topics, denoted by K, while using fixed values (implementation related) for α and β (Arun et al., 2010b; Greene, O'Callaghan et Cunningham,

2014) or on fixing K while estimating α and β (Terragni et al., 2021). Researchers have recently proposed techniques based on metaheuristics to tune all the LDA parameters (Agrawal, Fu et Menzies, 2018; Panichella et al., 2013; Panichella, 2021; Yarnguy et Kanarkard, 2018). Due to the very nature of this algorithm, researchers have pointed out the lack of stability that can lead to systematic errors and the difficulty of replicating results without proper parameters.

The motivation of our research is two folded. First, we want to continue discussing finding the nearest optimal LDA parameters started by Panichella (2021) and Agrawal *et al.* (2018). Second, we propose a framework to find such optimal parameters. Our approach will focus on testing additional surrogate metrics used to calibrate LDA in an unsupervised fashion and on validating the parameters found by evaluating their stability based on replicated runs, word embedding, and clustering techniques (Greene, O'Callaghan et Cunningham, 2014; Panichella, 2021). More precisely, we use the Genetic Algorithms (GA) combined with five different fitness functions (Silhouette, Coherence, R-square, Raw score, 2PJ) to determine a set of LDA parameters. We then evaluate the quality of the obtained parameters via a stability analysis based on replicated runs, convert the set of topics with word embedding to cluster them, and then quantify this analysis by a stability metric, which we will define as the final score.

Finally, we validate our proposed methodology and show the importance of the stability analysis with the following questions:

Q1) Do the different combinations of metaheuristic and fitness functions give similar results?

Q2) Are the run id of the maximum value of the fitness function and the run id of the maximum value of the stability analysis the same?

Q3) How do we explain the relation between a low number of topics and a high final score, and vice-versa?

Q4) Does hierarchical clustering find similar latent topics as Silhouette and Coherence?

Q5) Is it more advantageous to train a new GloVe than to use a pretrained GloVe?

Q6) How do we select the final parameters?

This paper is structured as follows. We start with a literature review in Section 4.4, discussing essential theoretical concepts related to our research. Section 4.5 introduces the methodology, datasets, and evaluation method. Then, in Section 4.6, we present and discuss the results of our experiments. In the last section, we state our conclusions and future work.

4.4 Literature review

4.4.1 Latent Dirichlet Allocation

The Latent Dirichlet Allocation (LDA) algorithm was first introduced by Blei *et al.* (2003). LDA is a generative and probabilistic technique in the topic modeling branch of the Natural Language Processing (NLP) discipline. It seeks to capture the heterogeneity of ideas (latent semantic structure) prevailing in a set of documents (corpus). The corpus must be transformed into a document-term matrix (DTM) in which the frequencies of words in documents are counted. This assumption means that the words in a document are exchangeable, so their order is unimportant. LDA aims to model documents as a discrete distribution over K latent topics, where every topic is modeled as a discrete distribution over the terms constituting the DTM (vocabulary).

Referring to the LDA graphical model in Figure 4.1, the nodes represent the random variables, and the edges represent the dependence between those random variables. The shaded node represents the observed random variable (i.e., the words within the documents), the unshaded nodes represent the hidden random variables, and the rectangular boxes indicate replication. Specifically, the underlying latent semantic structure is expressed by the topics ϕ (distributions over terms), the per-document topic proportion θ, and the per-word topic assignment z. However, ϕ, θ and z are unobserved, and this algorithm aims to determine them from the observed variables. Note that this representation is called Smoothed LDA model. In our context, we prefer the Smoothed LDA model instead of the original formulation, whose

weakness is that it does not place a prior on each ϕ_k. By this modification, it can be qualified as a fully Bayesian model.

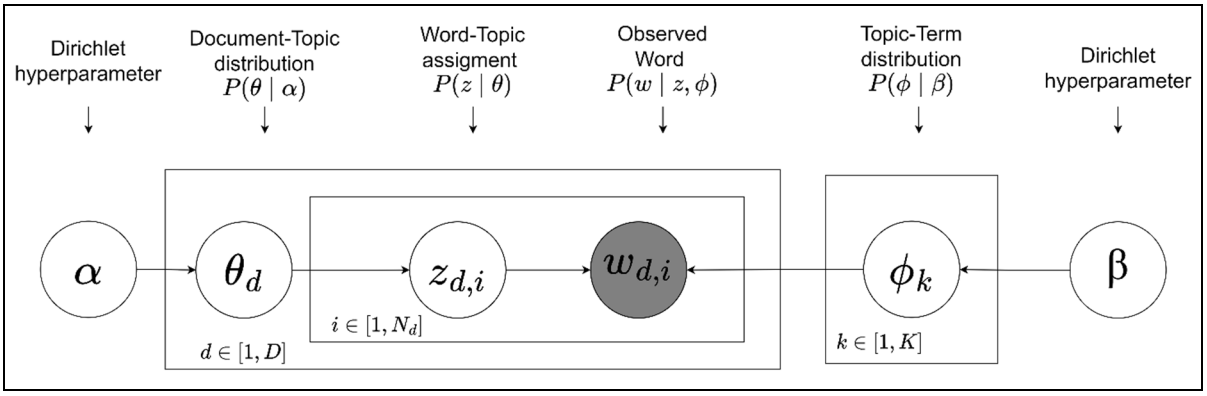


Figure 4.1 Smoothed LDA in plate notation (adapted from Blei *et al* (2003))

Because the LDA structure allows the observed variable to interact with the hidden variables (see Figure 4.1), the latent variables' posterior distribution can be inferred, given the observed documents. This interaction is exhibited by the generative process, in which the random process is assumed to have produced the observed data.

The generative process is described as follows (Blei, Ng et Jordan, 2003):

Algorithm 4.1 LDA generative process

for every topic $k = \{1, \ldots, K\}$,
  Draw a distribution over words $\phi_k \sim Dir(\beta)$
end for
for each document *d*,
  Draw a vector of topic proportions $\theta_d \sim Dir(\alpha)$ (i.e., per-document topic proportion)
  for each word *i* within document *d*,
    Draw a topic assignment $z_{d,i} \sim Mult(\theta_d), where\ z_{d,i} \in \{1, \ldots, K\}$ (i.e. per-word topic assignment)
    Draw a word $w_{d,i} \sim Mult(\phi_{z_{d,i}}), where\ w_{d,i} \in \{1, \ldots, V\}$
  end for
end for

Finally, the goal of this generative process is to estimate the following two matrices:

Θ: Representing the distribution of topics over documents. It's a $K \times D$ matrix, where the generic entry $\Theta(i,j)$ represents the probability of the j-th document being relevant to the i-th topic.

Φ: Representing the distribution of words over topics. It is a $V \times K$ matrix, where the generic entry $\Phi(i,j)$ represents the probability of the i-th word belonging to the j-th topic.

4.4.1.1 LDA inference

Multiple mathematical methods approximate the intractable integral arising in a Bayesian inference. The two common techniques to infer LDA for a given corpus are variational expectation-maximization (VEM) (Minka et Lafferty, 2012) and Gibbs sampling (Griffiths et Steyvers, 2004). In this paper, we will rely on the Gibbs sampling proposed by Panichella (2021) to estimate the θ and ϕ distributions. Panichella (2021) justifies that choice because prior studies used it and showed it has a faster and better convergence towards the global optimum than VEM.

4.4.1.2 The four LDA parameters to set

When using LDA and Gibbs sampling, the practitioner must set:

- K: the number of topics to generate from the corpus.
- α: the hyperparameter that influences the document-topic density.
- For a symmetric distribution, with a higher α, the documents will comprise more topics, and with a lower α, the documents will contain fewer topics.
- For an asymmetric distribution, a higher α results in a more specific topic distribution per document.
- β: the hyperparameter that influences the topic-word density.
- For a symmetric distribution, with a higher β, topics will comprise most of the words in the corpus, and with a lower β, they will consist of a few words.

- For an asymmetric distribution, the β will result in a more specific word distribution per topic.
- N: the number of Gibbs iterations. This parameter is specific to the Gibbs sampling generative model.

Regarding the parameters α and β, typically, LDA implementations use symmetric Dirichlet priors over Θ and Φ with fixed concentration parameters α and β, respectively. Therefore, there is an implicit assumption that they have little practical effect. However, those priors may either be symmetric or asymmetric. A symmetric distribution refers to one which is fixed to a uniform distribution. Conversely, an asymmetric distribution is not fixed to a uniform distribution. Wallach *et al.* (2009a) tested the four combinations of symmetric and asymmetric concentration parameters combinations. They found that an asymmetric α has substantial advantages over a symmetric prior, while an asymmetric β provides no real benefit (Wallach, Mimno et McCallum, 2009).

4.4.2 LDA instabilities

When researchers talk about model instability, they refer to the fact that repeated runs with the same data set and the same parameters can lead to different results. Due to the probabilistic nature of LDA, this model may exhibit this feature and therefore be referred to as an unstable model. Researchers classify this type of instability as type-I (Agrawal, Fu et Menzies, 2018; Hughes, Kim et Sudderth, 2015). In contrast, high stability corresponds to the high reliability of results based on stable models in the sense of improved reproducibility.

Studies (Agrawal, Fu et Menzies, 2018; Binkley et al., 2016; Hughes, Kim et Sudderth, 2015; Mantyla, Claes et Farooq, 2018; Rieger, Rahnenführer et Jentsch, 2020) suggest different strategies to improve LDA stability, including using random seeds (because the Gibbs method uses a random number generator with a starting seed), applying multiple Gibbs restarts (to

avoid converging toward local optima), or running LDA numerous times and aggregating the results (topic-terms matrix) of separate runs using clustering.

The second type of instability, type-II instability, was observed by Agrawal *et al.* (2018), who observed that different document orderings could lead to different topics when training LDA. This instability can be addressed by running LDA multiple times with various document orderings. In this study, we ran LDA numerous times and shuffled the document ordering before each LDA generation to consider these two types of instability. More details are provided in Section 4.5.

4.4.3 Estimating LDA parameters based on optimization methods

In Panichella (2021), his objective was to shed light on the impact of metaheuristic and fitness function combinations on the software engineering task. It should be noted that a software engineering task is a small corpus document containing a small amount of information (words), which is very specific. The authors compare the performance of seven metaheuristics and three fitness functions based on their ability to identify duplicate bug reports and the time needed to tune LDA. They also keep the default metaheuristic parameters. Their results suggest no clear winner among the combinations of the different search-based approaches (GA, SA, SCH) and the Silhouette or Coherence fitness functions when tuning LDA. They observe that multiple metaheuristics could achieve equally good performances. However, using the Raw score fitness function leads to more calculation time before convergence. Also, they do not recommend using the CMA-ES, RS, PSO, or DE as these metaheuristics achieve significantly less accurate results.

On the other hand, Agrawal *et al.* (2018) compare the combination of GA with Silhouette (Panichella et al., 2013) to the combination of DE with Raw score, and in their case, DE with Raw score performs better. Moreover, they argue that Panichella *et al.* (2013) did not consider the instability of type II, which could have led to the instability of LDA, when they proposed their method of LDA-GA with Silhouette.

This discrepancy between these researchers could be explained by the fact that each has a different research objective from an optimization point of view. Panichella *et al.* (2013) use a fitness function based on the document-topic matrix (Silhouette), while Agrawal *et al.* (2018) created their measure (Raw score), which uses the top nine words in the topic-word matrix.

This parameter optimization calculated by the metaheuristic procedures could lead to the belief that the resulting model is optimal. However, to validate the optimality of the model, Mantyla *et al.* (2018) propose a complementary method to evaluate the type-I stability of the final model. Their approach is further described in Section 4.5.2.2. The following table presents studies that apply metaheuristics to optimize LDA parameters in an unsupervised manner. In Table 4.1, we present some of the research conducted. As we can see, the different authors do not all use the same data set to discuss the quality of the proposed solution (a combination of a metaheuristic and a fitness function).

Table 4.1 Authors who used metaheuristic search to optimize LDA parameters

Author	**Metaheuristic**	**Fitness function**	**Dataset**
Yarnguy *et al.*, (2018)	Ant colony optimization	Perplexity	UCI (KOS, NIPS, ENRON)
Agrawal *et al.*, (2018)	Differential evolution (DE) Genetic algorithms (GA)	Raw score Silhouette	Pits, citemap, Stackoverflow, software document
Panichella *et al.*, (2013)	Genetic algorithms (GA)	Silhouette	Bench4BL (software document)
Panichella (2021)	Genetic algorithms (GA),	Silhouette Coherence	Bench4BL (software document)

	Differential evolution (DE), Random search (RS), Simulated Annealing (SA), Particle Swarm Optimization (PSO), Stochastic Hill Climbing (SHC), CMA-ES	Raw score	

4.4.4 Fitness functions

The combination of metaheuristic and fitness functions can guide the optimization process to find the nearest optimal parameters.

The following sections describe the fitness functions discussed in the present study.

4.4.4.1 Topic coherence

Several coherence measures have been proposed to evaluate the coherence of a set of words. We can mention two popular ones: the UCI measure proposed by Newman *et al.* (2010) and the UMass measure proposed by Mimno *et al.* (2011). Both measures calculate the coherence of a topic as the sum of the pairwise distributional similarity for a top M most probable words in the topic. However, the main difference between those two measures is that the UCI measure calculates the word co-occurrence frequencies over an external corpus; meanwhile, the UMass calculates the co-occurrence over the corpus used to train the model. For this research, we will use the one proposed by Mimno *et al.* (2011).

They defined:

- k as topic id
- $V^{(k)} = (v_1^{(k)}, \dots, v_M^{(k)})$ as a list of the M most probable words in topic k
- $D(v_l^{(k)})$ counts the number of documents containing $v_l^{(k)}$
- $D\left(v_m^{(k)}, v_l^{(k)}\right)$ counts the number of documents containing $v_m^{(k)}$ and $v_l^{(k)}$

$$C(t; V^{(k)}) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} log \frac{D\left(v_m^{(k)}, v_l^{(k)}\right) + \epsilon}{D(v_l^{(k)})} \quad (4.1)$$

Originally, the ϵ (smoothing factor) was set to 1 to avoid taking the logarithm of zero, but Stevens *et al.* (2012) found that the measure gives better results if this parameter is smaller. Because we rely on Matthew Denny's implementation for this measure[15], the epsilon value is set to 1.

The overall coherence for an LDA model (with parameters [K, N, α, β]) can then be computed as the arithmetic mean of the topic coherence scores of the K-generated topics. This final score takes a value between] -inf, 0 [where the value close to zero indicates a better score.

4.4.4.2 Silhouette coefficient

The silhouette coefficient for a cluster C is defined as follows:

$$s(d_i) = \frac{b(d_i) - a(d_i)}{max(a(d_i), b(d_i))} \quad (4.2)$$

where:

- s(di) denotes the silhouette coefficient for the document di in the corpus;
- a(di) measures the maximum distance of the document di to the other documents in the same cluster (cluster cohesion);

[15] https://github.com/matthewjdenny/SpeedReader

- b(di) measures the minimum distance of document di to another document in a different cluster (cluster separation).

The overall silhouette coefficient can be measured as the arithmetic mean of the coefficients $s(d_i)$ for all documents in the corpus (Panichella et al., 2013; Panichella, 2021).

$$s(C) = \frac{1}{D}\sum_{i=1}^{D} s(d_i) \tag{4.3}$$

s(C) takes a value between [-1, + 1], where a value close to 1 indicates a better score because (on average) the separation is larger than the cohesion of the clusters.

4.4.4.3 Raw score

Agrawal *et al.* (2018) proposed an alternative surrogate metric, namely, the Raw score, to measure the quality of the LDA parameters. They modify the Jaccard measure to do cross-run similarity of topics. They assess topic model stability via the median number overlaps of the top nine words. To this end, they execute n = 10 times the calculation of LDA over different random shuffles of the data documents, where they are in tf-idf form[16]. Then, they compute the overlap score of the previously agglomerated topics. To avoid sampling bias, they execute the previous steps n = 10 times to compute the median of the 10 overlap scores.

The Raw score takes a value between [0, 1], where a value close to 1 indicates a better score.

[16] The DTM uses a term frequency (tf) scheme or a term frequency with the inverse document frequency (tf-idf) scheme. The tf shows the term counts in the documents and the tf-idf shows the weighted counts. The weights are based on how important a term is in the corpus. For example, the weight of a given term will be based on the number of occurrence in a given document but also the numbers of occurrence over the whole corpus.

4.4.4.4 Perplexity

The perplexity measure is widely used in information theory (Asuncion et al., 2012; Wallach et al., 2009; Yarnguy et Kanarkard, 2018). It measures the ability of a model to predict a sample. The smaller the value, the better the model. However, researchers caution that this measure does not remain constant with different topics and corpus sizes. The perplexity depends on its implementation and the type of data sets used (Zhao et al., 2015). For these reasons, we will not use this measure.

4.4.4.5 R-square

Jones, T. (2019) (Jones, 2019) proposes adapting the R-squared goodness-of-fit measure to topic models while keeping the same interpretation as the one used in a broader class of statistical models. For example, when applied to a linear model, this measure explains the proportion of variability in the data; the results are bound between 0 and 1, where a higher value indicates a better model fit.

The adaptation of R-square to topic modelling follows the same principle, and it is mathematically described as follows:

$$f_d = E(w_d) = n_d \odot \theta_d \cdot \Phi \tag{4.4}$$

$$\overline{w} = \frac{1}{D} \sum_{d=1}^{D} w_d \tag{4.5}$$

$$SS_{tot} = \sum_{d=1}^{D} dist(w_d, \overline{w})^2 \tag{4.6}$$

$$SS_{resid} = \sum_{d=1}^{D} dist(w_d, f_d)^2 \tag{4.7}$$

$$R^2 \equiv 1 - \frac{SS_{resid}}{SS_{tot}} \tag{4.8}$$

where:

- f_d is the document's fitted value and represents the outcomes of a multinomial random variable under the model;
- $\odot$ denotes the elementwise multiplication;
- n_d is the number of terms in the document;
- $dist()$ is the Euclidian distance;
- SS_{tot} is the total sum of squares;
- SS_{resid} is the residual sum of squares.

We rely on the Tommy Jones implementation of R-square[17].

4.4.5 Measuring the stability of LDA topics based on replicated runs

Recent research on LDA stability states that a high stability value in running LDA multiple times with fixed parameters means many topics with representatives in each LDA run can be identified. Although LDA is run multiple times with fixed parameters, its results can still be unstable due to different seeds and random restart strategies.

Panichella (2021) and Panichella *et al.* (2013) do not perform independent runs of LDA with the same parameters found by the optimization process. Rather, they run the metaheuristic/fitness function combination 30 and 100 times in their first and second papers, respectively. They use the resulting LDA parameters at the end of each run to generate the corresponding LDA model. Since they address the stability problem using the same seed and random restart strategies, their method ensures that the same results are obtained even when LDA is rerun several times with the same respective parameters.

[17] https://cloud.r-project.org/web/packages/textmineR/index.html

For Agrawal *et al.* (2018), the replicated runs process is incorporated into the fitness function they developed, namely, the Raw score defined in Section 4.4.4.3. Therefore, stability is implicitly addressed by the fitness function.

Mantyla *et al.* (2018) propose a complementary method to evaluate the type-I stability of the final model. Their approach to providing information about the topic's stability must be used at the end of the optimization process. They perform 20 replicated runs of LDA with fixed parameters. Then, they extract the top 10 most probable words of each topic. These will be used to cluster the words from these 20 topic-word matrices via the k-medoid algorithm using the word embedding produced by the GloVe algorithm built on their corpus. Finally, they evaluate the similarity of the clustered topics with the suggested Rank Biased Overlap (RBO) measure (Webber, Moffat et Zobel, 2010). The advantage of this method is that it offers the possibility of further investigation of the clusters as needed. A shortfall in their approach is that they do not consider type-II instability. Although RBO seems useful thanks to its implementation of a more flexible form of a Jaccard coefficient (since it considers the word ranking), it requires parameter optimization.

Rieger *et al.* (2020) propose to evaluate the stability of LDA with clustering techniques applied to replicate LDA runs. They introduce a new similarity measure, S-CLOP (Similarity of multiple sets by Clustering with LOcal Pruning), based on a pruning algorithm applied to the hierarchical clustering results with complete linkage. Starting with a broad set of topics from all the LDA runs (replications), the ideal goal will be that the pruning algorithm results in clusters, each containing one topic for every replication of the modeling procedure. This increase in reliability is achieved by running the LDA several times (they suggest 50 times).

Finally, this concludes the related work that will help elaborate our methodological solution.

4.5 Methodology and implementation

The implementation of our experiment relies on R packages. We base our study on Panichella (2021)'s and Mantyla *et al.* (2018) previous works, respectively, at http://doi.org/10.5281/zenodo.4016590 and https://github.com/M3SOulu/Measuring-LDA-Topic-Stability. We use and adapt their code based to our study needs. The R scripts used in our experiment are publicly available at this-url-will-be-disclosed-in-the-published-document.

The methodology of this research can be broken down into three independent steps:

1. The first step consists in preprocessing the corpus to build the document term matrix (DTM).
2. The second step focuses on finding the nearest optimal LDA parameters.
3. The third step focuses on evaluating the stability of the LDA model generated by the previously found parameters.

4.5.1 Corpus preprocessing

The corpus preprocessing aims to extract the relevant words from each document constituting the corpus and eliminates any potential noise. For evaluation purposes, we used four different corpora published[18] by the researcher Derek Greene (Greene, O'Callaghan et Cunningham, 2014) which are presented in the following table.

Table 4.2 Corpora information

Corpus	DTM dimensions		Description
	Documents	Terms	
20 Newsgroups -ol-6-1	2191	8499	Natural Classes (topics): 6 Document topical categories and counting: crypt (991), electronics (240), med (240),

[18] Dereck Greene resources website: http://mlg.ucd.ie/howmanytopics/index.html

			politics (240), religion (240), space (240) o* - * - *: dataset with clusters that overlap considerably. *l - * - *: dataset with unbalanced clusters where one cluster contains 60% of the documents.
20 Newsgroups -sb-6-1	3000	10673	Natural Classes (topics): 6 Document topical categories and counting: autos (500), Christian (500), crypt (500), forsale (500), hockey (500), windows (500) s* - * - *: dataset with reasonably compact and well-separated clusters. *b - * - *: dataset with balanced clusters containing 500 documents each.

Table 4.3 Corpora information (continued)

Corpus	DTM dimensions		Description
	Documents	Terms	
BBC-Sport	737	3848	Natural Classes (topics): 5 Document topical categories and counting: athletics (101), cricket (124), football (265), rugby (147), tennis (100)
BBC-News	2225	8850	Natural Classes (topics): 5 Document topical categories and counting: business (510), entertainment (386), politics (417), sport (511), tech (401)

The corpora based on 20 Newsgroups are artificially constructed to allow smaller documents and human-annotated "ground truths" by the topical categories. The annotation was done to derive smaller datasets for which the correct value of the number of topics (K) is known. The corpora from BBC were studied by D. Greene in these research works (Greene et Cunningham, 2006; Greene, O'Callaghan et Cunningham, 2014) and concern the news and sports sections. Using these corpora is advantageous as they offer a research result benchmark for comparison with our methodology results.

Regarding text preprocessing, we used the final DTM proposed by Greene *et al.* (2014) for the two corpora related to 20 Newsgroups, where stop-word removal, stemming, and terms occurring in less than three documents were eliminated. We used the same steps for BBC, but here, we applied lemmatization instead of the stemming approach. This modification was possible because we had access to the two original corpora. Using lemmatization instead of stemming is motivated by the word embedding technique in the stability step. The resulting preprocessed documents were then converted into a DTM. We used the term frequency (tf) scheme instead of the term frequency with the inverse document frequency (tf-idf) scheme applied by Agrawal *et al.* (2018) and Panichella (2021). The LDA is a word generation model

that implies a word is formed from a multinomial distribution. In the LDA context, we believe using the term frequencies (tf) instead of the inverse document frequency (tf-idf) makes more sense. It makes no sense to suggest that, for example, 32% of a word (tf-idf weight) is created by some distribution. However, for another context (e.g., in a search-engine context), it could make more sense to use the inverse document frequency (tf-idf).

4.5.2 Finding the nearest optimal parameters

This section will discuss finding the optimal LDA parameters [K, α, β, N]. In this process, we will use different fitness functions. In addition to the previously discussed, we developed a new fitness function. Therefore, we will present our new proposed fitness function and then discuss the methodology related to the run execution of the possible GA-fitness function combinations.

4.5.2.1 Proposal for a new fitness function: Phi-Phi-Prime-mean-Jaccard (2PJ)

Panichella (2021) suggests that future work should improve the surrogate metrics (fitness functions) used to calibrate/tune LDA in an unsupervised fashion. We propose a new approach for a fitness function. The idea is to apply the average Jaccard similarity over the Φ (topic-words) and Φ' (word-topics) distributions produced by LDA. The calculation of Φ' is based on the Φ matrix, in which the resulting rows of Φ are $P(w_{d,i}|k)$. By Bayes' theorem, we can calculate its prime function as follows:

$$\Phi' = P(k|w_{d,i}) = \frac{P(w_{d,i}|k) \times P(k)}{P(w_{d,i})} \quad (4.9)$$

If we assume that documents are equally distributed, the probability of document *i* is

$$P(d_i) = \frac{1}{D}. \quad (4.10)$$

The weighted probability of a topic:

$$P(k) = \sum_{i=1}^{D} P(k|\, d_i) \times P(d_i) \quad (4.11)$$

The probability of a word:

$$P(w_{d,i}) = \sum_{k=1}^{K} P(w_{d,i}|k) \times P(k) \quad (4.12)$$

For each row of Φ and the Φ' transposed, we can now select the top M words to calculate their similarity with the Jaccard coefficient. Finally, we compute the arithmetic average over the K topics to approximate the overall Jaccard's similarity.

We refer to that measure herein as 2PJ for Phi-Phi-Prime-mean-Jaccard. The 2PJ takes a value between [0, 1], where a value close to 1 indicates a better score.

4.5.2.2 Methodology related to the run execution of the possible GA-fitness combinations

As suggested by Panichella (2021), we used the metaheuristic genetic algorithms (GA) (Scrucca, 2013) to find the nearest optimal LDA parameters [K, α, β, N] based on Gibbs sampling. We used the TextmineR library (Jones, Doane et Jones, 2016) for the LDA implementation because it allowed us to modify the symmetry assumption on prior distributions. As Wallach *et al.* (2009a) suggested, we forced α to be asymmetric and β to be symmetric. We experimented with the five following fitness functions to guide the GA in the optimization process. We use four previously discussed fitness functions (Silhouette, Coherence, Raw score, and R-square) and the new proposed one, 2PJ.

We executed several independent runs for each combination to allow a fair comparison between the possible combinations of GA-fitness functions. The number of LDA models generated during the optimization process is based on the design of the fitness functions. As a reminder, for n execution of the Raw score, n^2 LDA models will be generated, versus one for

the other fitness functions. Given that ratio, we executed only one independent run for the GA-Raw score and 25 for the other combinations.

The followings are the Genetic Algorithms' parameter settings and fitness functions:

- For the GA, we used the same parameters as described in Panichella (2021):
- population size of 10 LDA configurations;
- number of GA iterations to 5;
- binary simulated crossover with probability pc=0.9;
- polynomial mutation with mutation probability pm=0.25; and
- mutation index ηm=20.
- For Silhouette, we use the Euclidean distance to compute the distance between documents in the topics space.
- For 2PJ, Coherence, and Raw score, we use the top 10 words.
- For Raw score, we use 7 overlap words and n = 5.
- For R-square, no parameters are needed.

In this step, we only address the type-II instability problem by applying a random shuffle on the DTM (documents order) before fitting the LDA model. This applies to all the fitness functions except for the Raw score. The two types of instabilities are implicitly considered for this latter due to their design.

Finally, that procedure is applied to all four datasets, with the LDA parameter boundaries set as follows:

Table 4.4 List of LDA parameters and their ranges

Parameter	Range	Description
K	[5, 20]	Number of topics to extract
α	[0.0000001, 1]	Distribution of topics over documents

β	[0.0000001, 1]	Distribution of topics over words
N	[200, 300]	Number of Gibbs iterations

The range of topics to extract (K) was selected based on the fact that there should be a minimum of 5 topics, as shown in Table 4.2. Recall from Section 4.4.1.2 that a higher α results in a more specific topic distribution per document. The α range is in accordance with the small number of topics identified by Greene *et al.* (2014). Since β represents the topic-word density, we preferred a lower β (i.e., few words per topic). We used the same Gibbs iterations (N) range as Panichella (2021).

4.5.3 Stability evaluation

The stability analysis, inspired by Mantyla *et al.* (2018), consists of four independent steps applied sequentially (replication of the model, word embedding integration, word clustering technique, and quantitative clustering analysis). They are described as follows:

4.5.3.1 Replication of the model

We generated the LDA models 20 times and aggregated all topic-word matrices with the optimal parameters identified. To be consistent with the two types of instability, we apply random shuffling on the DTM (type-II) before each model generation, and the 20 independent runs handle type-I.

4.5.3.2 Word embedding integration

It has been shown that the vector space produced by a word embedding algorithm such as GloVe tends to cluster similar words (Pennington, Socher et Manning, 2014). Our idea is to take advantage of the potential synonymic and semantic similarity of the vector space generated by GloVe to validate the cohesion of the most significant terms constituting the different latent topics discovered (Mantyla, Claes et Farooq, 2018) over the 20 independent

runs. To this end, let's denote the vector space of words generated by GloVe as the matrix G consisting of t terms and w weight vectors (G = [V, w]). The matrix product between the topic-word matrix produced by LDA, T = [K, V] and G, gives the topic-weight matrix TV = [K,w], which will be applied to the clustering and evaluation techniques.

We used the GloVe model trained on Common Crawl data offered by Stanford5[19]. We also created two new models trained on the BBC corpora (sport and news) separately and one other by combining the two corpora. The GloVe models are used to see if they impact the stability analysis. No preprocessing was applied to avoid losing semantic information in the BBC corpora. Moreover, as the GloVe paper (Pennington, Socher et Manning, 2014) suggested, Table 4.5 shows the used parameters.

Table 4.5 GloVe parameters

Parameter	**Value**
Skip-grams window	10
Skip-grams window context	Symmetric
X max	100
Learning rate	0.05
Alpha	0.75
Lambda	0

The following table shows basic information about the dimensions of the vectors:

Table 4.6 GloVe basic information

GloVe model	**Total words**	**Dimension**
GloVe-Common-crawl	2,196,018	300

[19] https://github.com/stanfordnlp/GloVe

GloVe-BBC-sport	14,158	300
GloVe-BBC-news	32,286	300
GloVe-BBC-sport-news	33,863	300

The stability analysis was performed using the four different GloVe models. The following table shows the percentage of each DTM vocabulary included in the terms constituting the GloVe vector representations.

Table 4.7 Terms intersection between GloVe and DTM (in %)

	GloVe-Common-Crawl	**GloVe-BBC-Sport**	**GloVe-BBC-News**	**GloVe-BBC-Sport-News**
20 Newsgroups -ol-6-1	72.44	**23.92**	**37.91**	**38.41**
20 Newsgroups -sb-6-1	71.20	**20.63**	**33.69**	**34.31**
BBC-Sport	92.83	96.10	92.13	98.67
BBC-News	95.40	54.80	96.96	97.08

The lowest percentages associated with the 20 Newsgroups can be explained because their vocabulary is reduced to their stem form.

4.5.3.3 Word clustering technique

The k-medoid algorithm will cluster the topic-weight matrices TV into K topics.

4.5.3.4 Quantitative clustering analysis

We can analyze the results over the K clusters or within each respective cluster at this stage. To measure the quality of the clusters, we used the internal criterion, Silhouette. It is used for cluster validation because it considers how similar each object is within its cluster (cohesion)

and how different it is from the other clusters (separation). A silhouette coefficient is calculated for each cluster. Then the overall silhouette coefficient is measured as the arithmetic mean of the coefficients over all the clusters. The range of values of the Silhouette coefficient goes from -1 to 1. However, for comparison purposes, we recentered the scores between 0 and 1.

We also calculated the Jaccard coefficient based on the number of term intersections between two sets of topics within each cluster. A mean Jaccard is also calculated over all the clusters. It takes values between 0 and 1. We combined the mean Jaccard and the mean Silhouette into one metric (using the arithmetic mean) and denoted it as ***mean-sil-jacc***.

To describe the topic stability, i.e., whether a set of topics within the same cluster relates to the same list of terms, we applied pairwise comparisons between each topic-word matrix in the cluster where we considered the top 10 words. As a cluster measurement, we used the Kendall coefficient to consider the ability of the model to generate the same topic-word sequences (order effect) and to summarize it as an arithmetic mean centered between 0 and 1.

As an external measure, we calculated a similarity distance based on the number of topics in the current cluster compared to the expected value (the number of replicated runs). We called it n-topic-per-clusters and calculated it as follows:

$$max\left(1 - \left|\frac{all\ topics\ in\ the\ cluster\ under\ analysis\ -\ number\ of\ replicated\ runs}{number\ of\ replicated\ runs}\right|, 0\right) \quad (4.13)$$

Finally, we calculated the **average arithmetic mean of the Kendall-centered, mean-sil-jacc, and n-topic-per-clusters metrics to create the final score measure**.

Using this final metric, we modify the methodology proposed by Mantyla *et al.* (2018), suggesting using Rank Biased Overlap (RBO) to compare ranked lists. We agree with Rieger *et al.* (2020) that although the RBO measure seems useful because it implements a more flexible form of a Jaccard coefficient, it is time-consuming due to the required parameter optimization.

4.6 Results and discussion

In this section, we present the results obtained during our experimentation. We conduct our discussion considering the two main steps of our methodology: parameters tuning and stability evaluation. This section will allow us to answer the research questions we stated in this paper's introduction.

4.6.1 Parameter tuning

In this section, we analyze the number of topics found and the duration for each combination of metaheuristic-fitness functions from a statistical point of view. As a reminder, the dispersion measures are based on 25 independent runs per combination, except for the Raw score fitness function, which uses only 1 run. This is equivalent to the number of LDA models evaluated during the optimization process. The statistical results are presented in Table 4.8, followed by their discussion.

Table 4.8 Summary of the combinations of GA and fitness functions per corpus

	Median number of topics found	**Mean number of topics found**	**Variance coefficient mean topic found (%)**	**Mean duration (seconds)**	**Variance coefficient Mean duration (%)**	**Expected number of topics (D. Greene)**
20 Newsgroups - sb-6-1 corpus						
Coherence	7	7	16.27	296	12.94	6
2PJ	6	6	28.46	280	14.21	6
R-square	17	16	19.94	375	13.05	6
Raw score	10	10	NA[†]	245[‡]	NA[†]	6

Silhouette	7	7	23.08	533	11.71	6
†Statistics calculated in one execution.						
‡Total run time divided by 25 to allow easier comparison.						

Table 4.9 Summary of the combinations of GA and fitness functions per corpus (continued)

	Median number of topics found	Mean number of topics found	Variance coefficient mean topic found (%)	Mean duration (seconds)	Variance coefficient Mean duration (%)	Expected number of topics (D. Greene)
20 Newsgroups - ol-6-1 corpus						
Coherence	6	6	28.17	251	11.59	6
2PJ	18	18	11.79	330	11.58	6
R-square	19	18	7.93	341	8.65	6
Raw score	13	13	NA†	296‡	NA†	6
Silhouette	7	8	39.47	474	17.70	6
BBC-Sport corpus						
Coherence	9	9	30.74	86	11.27	5
2PJ	18	17	18.75	111	14.92	5
R-square	19	19	6.73	113	8.44	5
Raw score	11	11	NA†	89‡	NA†	5
Silhouette	7	7	26.08	157	14.27	5
BBC-News corpus						
Coherence	7	7	28.07	322	12.54	5
2PJ	15	14	34.57	371	20.55	5
R-square	19	19	7.64	433	12.16	5
Raw score	12	12	NA†	343‡	NA†	5
Silhouette	7	8	30.94	588	20.97	5

†Statistics calculated in one execution.

‡Total run time divided by 25 to allow easier comparison.

Number of topics

For all corpora, regarding the number of topics found by the optimization process (see Table 4.8, columns median and mean number of topics found), we observed that the Coherence and Silhouette fitness functions found smaller numbers of topics than the three other fitness functions. The higher values may suggest that these fitness functions are more likely to generate hierarchical clustering. To confirm this assumption, we used a hierarchical clustering analysis approach. Based on Rieger *et al.* (2020), we used hierarchical clustering with complete linkage. To compute the topic similarities, we used the Hellinger distance.

The following graphs illustrate this clustering approach in one executed run of the BBCNews corpus for those three fitness functions: 2PJ, R-square, and Raw score. Based on the expected number of topics reported in Table 4.8, we pruned the resulting dendrogram in five to see if they could be retrieved. The results are presented below, showing clear evidence that we can retrieve the expected number of topics.

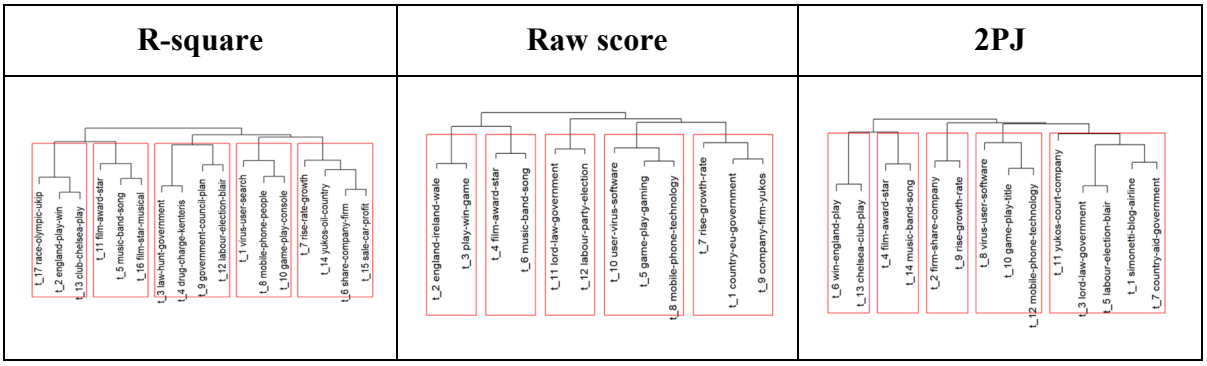


Figure 4.2 Hierarchical clustering results on BBC-News data set

This observation between the two groups of fitness functions is interesting because the emergence of latent topics found by Silhouette and Coherence seems to be on a more general level. In comparison, the other fitness functions find the general topics, and furthermore, they have more fine-grained latent information on the analyzed corpus.

The mean and median (of the number of topics) show similar results, indicating that the distributions are not skewed. Furthermore, if we compare the expected number of topics found by D. Greene, the variance coefficient for Silhouette and Coherence indicates that our results agree with the expected number of topics.

Duration

The execution time was calculated for the entire process of the optimization task, and the mean duration time was calculated over the 25 independent runs. The parallel computation (synchronization, communication, thread creation/destruction) influences the estimation of the durations used by the GA implementation on the server. Moreover, the LDA calculation time is influenced by the vocabulary length (see Table 4.2), the number of documents (see Table 4.2), and the number of topics dictated by the solution under optimization (see Table 4.4).

As reported in Table 4.8, we can see that the fitness functions that allow more topics take more time than the Coherence measure. However, what is counter-intuitive is that the Silhouette measurement is slower overall. Furthermore, the fact that Coherence is faster than Silhouette does not coincide with Panichella (2021) since their results show that Silhouette should be faster.

Following this analysis, we can now answer the first question:

Q1) Do the different combinations of metaheuristic and fitness functions give similar results?
We believe that the combinations between the GA and the different fitness functions provide similar results to those expected by Greene *et al.* (2014)'s studies. Although we observed that the R-square, 2PJ, and Raw score fitness functions find more topics. The hierarchical analysis (Figure 4.2) of these clusters shows that it is possible to find more general latent topics such as the ones discovered by traditional methods such as Silhouette and Coherence. Finally, in the

context of this analysis, we have just demonstrated the potential of using new fitness functions to discover latent topics by LDA.

4.6.2 Stability

As mentioned in Section 4.5.3, the stability analysis will help us determine the best run to select among the 25 runs of the optimization process. This evaluation consists of generating 20 LDA models with the parameters found and then projecting the topic-word matrices into a vector representation using the GloVe models. Then we cluster these word vectors using the k-medoid algorithm and evaluate them using intra-clustering and extra-clustering metrics. We define the final score as a final measure of the stability analysis by combining the calculation of inter- and extra-clustering characteristics. The final score metric will help us determine the more stable run id, leading to selecting the final parameters.

Q2) Are the run id of the maximum value of the fitness function and the run id of the maximum value of the stability analysis the same?
To answer this question, we compared the fitness score (the final value obtained by the optimization process) to the final score obtained by the stability analysis. The range of the fitness score is driven by the nature of the fitness function used. In comparison, the final score calculated will always be between 0 and 1.

The following tables were obtained by selecting the two highest values sorted by the fitness score or the stability score obtained by each GloVe model. For this comparison, we use only Silhouette and Coherence.

Table 4.10 Top two runs on the BBC-News corpus using the Silhouette fitness function

			Final score (%)			
			GloVe models			
Run id	**K**	**Fitness score**	**Common-Crawl**	**BBC-Sport**	**BBC-News**	**BBC-Sport-News**
Sorted by Fitness score						
	6	**0.69**	79.49	78.51	79.76	79.47
7	6	**0.68**	81.16	80.56	81.51	81.14
Sorted by Common-Crawl						
	5	0.53	**95.24**	95.00	95.25	95.18
17	6	0.57	**83.94**	83.13	84.0	83.4
Sorted by BBC-Sport						
23	5	0.53	95.24	**95.00**	95.25	95.18
17	6	0.57	83.94	**83.13**	84.0	83.4
Sorted by BBC-News						
23	5	0.53	95.24	95.00	**95.25**	95.18
17	6	0.57	83.94	83.13	**84.0**	83.4
Sorted by BBC-Sport-News						
23	5	0.53	95.24	95.00	95.25	**95.18**
17	6	0.57	83.94	83.13	84.0	**83.4**

Table 4.11 Top two runs on the 20 Newsgroups-sb-6-1 corpus using the Coherence fitness function

			Final Score (%)			
			GloVe models			
Run id	**K**	**Fitness score**	**Common-Crawl**	**BBC-Sport**	**BBC-News**	**BBC-Sport-News**
Sorted by Fitness						
	7	**-51.47**	78.16	76.38	77.55	77.03
12	7	**-51.48**	79.32	80.84	80.9	80.84
Sorted by Common-Crawl						
	6	-56.25	**95.57**	94.68	94.81	94.92
19	6	-54.43	**93.91**	93.34	93.02	93.10
Sorted by BBC-Sport						
1	6	-56.25	95.57	**94.68**	94.81	94.92
19	6	-54.43	93.91	**93.34**	93.02	93.10
Sorted by BBC-News						
1	6	-56.25	95.57	94.68	**94.81**	94.92
19	6	-54.43	93.91	93.34	**93.02**	93.10
Sorted by BBC-Sport-News						
1	6	-56.25	95.57	94.68	94.81	**94.92**
19	6	-54.43	93.91	93.34	93.02	**93.10**

Looking at the run id and the number of topics in Table 4.10 and Table 4.11, we can see the non-concordance between the highest fitness and final scores. Therefore, we can deduce that the optimization process does not automatically lead to the most stable LDA model. Furthermore, for the selected corpus (BBC-News and 20 Newsgroups-sb-6-1), we can note that the two highest selected values all have the same run id (run id 23 in Table 4.10 and run id 1 in Table 4.11), independently of the GloVe models used. The advantage of using the final score value instead of the fitness score is that using the former provides a more stable model.

Furthermore, we ensured that all underlying metrics constituting the final score agree (i.e., not only one or two scores push the final score high). Table 4.12 and Table 4.13 compare the underlying metrics of the best run selected by the fitness function versus the selection based on the final score (run id prefixed by * in Table 4.10 and Table 4.11).

Table 4.12 Final score underlying metrics comparison for the best run on the BBC-News corpus using the Silhouette fitness function

Run id	**Mean-Kendall-centered**	**mean-sil-jacc**	**n-topic-per-clusters**	**Final score**
14	0.777316	0.748714	0.866667	79.76
23	0.933253	0.924393	1	95.25

Table 4.13 Final score underlying metrics comparison for the best run on the 20 Newsgroups-SB-6-1 corpus using the Coherence fitness function

Run id	**Mean-Kendall-centered**	**mean-sil-jacc**	**n-topic-per-clusters**	**Final score**
2	0.78	0.72	0.84	78.16
1	0.95	0.92	1.0	95.57

Finally, we can conclude that the best solution found by the optimization process does not point out the most stable model and that applying the stability analysis seems to improve the reliability/replicability. At this point, we can select the most stable parameters to generate the LDA for each fitness function used. Based on this selective approach, the results presented in the following table are based on the maximum value among all runs as a function of the fitness score and final score of the four different GloVe models.

Table 4.14 Best run for all corpus and fitness functions

	Coherence			**Silhouette**			**R-square**			**2PJ**			**Raw score**		
	Run id	Final score	K	Run id	Final score	K	Run id	Final score	K	Run id	Final score	K	Run id	Final score	K
BBC-News corpus															
Fitness score	18	-	5	14	-	6	1	-	20	8	-	18	1	-	12
GloVe-Common-Crawl	18	95.38	5	23	95.24	5	23	69.97	20	7	90.15	5	1	72.40	12
GloVe-BBC-Sport	22	95.57	5	23	95.00	5	23	71.26	20	7	89.73	5	1	70.47	12
***GloVe-BBC-News**	**22**	**95.28**	**5**	**23**	**95.25**	**5**	**13**	**71.49**	**17**	**7**	**90.06**	**5**	**1**	**72.65**	**12**
GloVe-BBC-Sport-News	22	95.28	5	23	95.18	5	13	71.57	17	7	90.03	5	1	72.68	12
BBC-Sport corpus															
Fitness score	10	-	5	15	-	7	18	-	20	1	-	7	1	-	11
GloVe-Common-Crawl	15	80.32	6	4	80.27	6	16	69.24	16	20	72.23	14	1	76.84	11
***GloVe-BBC-Sport**	**15**	**80.78**	**6**	**4**	**79.44**	**6**	**16**	**70.81**	**16**	**15**	**74.85**	**14**	**1**	**77.50**	**11**
GloVe-BBC-News	15	80.80	6	4	80.25	6	16	69.32	16	15	74.48	14	1	76.97	11

GloVe-BBC-Sport-News	15	80.76	6	4	80.25	6	16	70.39	16	15	74.94	14	1	77.07	11

Table 4.15 Best run for all corpus and fitness functions (continued)

	Coherence			**Silhouette**			**R-square**			**2PJ**			**Raw score**		
	Run id	Final score	K	Run id	Final score	K	Run id	Final score	K	Run id	Final score	K	Run id	Final score	K
20 Newsgroups - *ol-6-1 corpus*															
Fitness score	7	-	5	10	-	6	11	-	18	9	-	17	1	-	13
***GloVe-Common-Crawl**	**11**	**71.77**	**5**	**22**	**69.39**	**6**	**6**	**65.28**	**19**	**16**	**69.18**	**12**	**1**	**67.40**	**13**
GloVe-BBC-Sport	1	72.55	5	22	69.74	6	1	63.85	19	16	68.92	12	1	65.60	13
GloVe-BBC-News	1	71.65	5	2	67.90	7	20	64.57	20	16	68.08	12	1	62.61	13
GloVe-BBC-Sport-News	1	71.74	5	2	67.67	7	20	64.10	20	16	67.65	12	1	62.82	13
20 Newsgroups - *sb-6-1 corpus*															
Fitness score	2	-	7	19	-	7	19	-	20	14	-	5	1	-	10
***GloVe-Common-Crawl**	**1**	**95.57**	**6**	**1**	**91.75**	**6**	**3**	**77.42**	**8**	**12**	**92.43**	**5**	**1**	**74.55**	**10**
GloVe-BBC-Sport	1	94.68	6	1	91.04	6	3	76.63	8	18	91.58	6	1	73.52	10

GloVe-BBC-News	1	94.81	6	1	91.27	6	3	76.19	8	4	91.51	6	1	73.00	10
GloVe-BBC-Sport-News	1	94.92	6	1	91.38	6	3	76.12	8	4	91.65	6	1	72.73	10

A remarkable feature of the results presented in Table 4.14 is the relationship between the number of topics discovered (K) and the final score results. This observation leads to the following question:

Q3) How do we explain the relation between a low number of topics and a high final score, and vice-versa?

Let us compare the different final scores obtained by the different fitness functions for a given data set. Notably, the number of topics discovered influences the value of the final score. As we can observe in Table 4.14, the higher the number of topics, the lower the final score, and vice-versa. This relationship can be explained by the difficulty of the clustering algorithm to correctly group the hierarchical elements that then influence the different metrics constituting the final score. Specifically, the relation between a low number of topics and a high final score happens when the topics discovered are more general. Then, the projection of the terms constituting the groups of topics (via work embedding) facilitates the task of the clustering algorithm (k-medoid) because their positioning in the word vector space does not impose any overlap. Conversely, when the number of topics discovered comprises hierarchical refinement, the projection of the terms into the word vector space does not distinguish between the parent and child groupings of the different hierarchies discovered. This feature adds difficulty to the clustering algorithm.

Q4) Does hierarchical clustering find similar latent topics as Silhouette and Coherence?
We cannot rely on the final score value to compare the two groups of fitness functions: the ones discovering topics at a less low level of granularity (Silhouette and Coherence) and the ones discovering the highest level of granularity (R-square, 2PJ, and Raw score). First, the final score is influenced by the number of topics discovered. Second, the Silhouette and Coherence fitness functions tend to discover fewer latent topics (described as a basic understanding of the different corpora) than the other three. On the other hand, their hierarchical characteristics can be exploited to convert many topics into smaller ones. When two models (a reference model and a comparison model) have the same number of topics, it is easier to compute a similarity value between the groups of terms discovered.

The following approach is used to enable this comparison:

- Compute the dendrogram of the comparison models, i.e., the models generated by R-square, 2PJ, and Raw score fitness functions.
- Prune the dendrogram to obtain the number of clusters equal to the number of topics of the reference model, i.e., the models generated by the Silhouette and Coherence fitness functions.
- Extract the 10 most important terms from each branch constituting each cluster, e.g., if there are three topics in a cluster and only 10 words in the reference model, we select 30 words for the comparison model.
- For the list of terms constituting each cluster in the comparison model (e.g., 30 words), we determine its most similar list of terms in the reference model (e.g., 10 words) by doing a pairwise comparison using the Jaccard similarity measure. Then, an association is carried out based on the highest similarity value. For each association, calculate the asymmetric similarity as follows:

$$\frac{|list\ of\ terms\ in\ comparison\ model \cap list\ of\ terms\ in\ reference\ model|}{|\ list\ of\ terms\ in\ comparison\ model|} \quad (4.14)$$

- Finally, calculate the arithmetic mean of each association similarity value over all the clusters to obtain the global similarity score.

We used the LDA parameters obtained by each * prefixed row from Table 4.14 to produce the following results. Their selection is based on the higher percentage of vocabulary intersection between the GloVe model and the data set under analysis (see Table 4.7), except for the BBC-Sport corpus used by the BBC-Sport GloVe model. The results below show the arithmetic mean of each association similarity value. The columns are for the reference models, while the rows are for the comparison model.

Table 4.16 Percentage of word similarity for BBC-Sport corpus

	Coherence	**Silhouette**
R-square	0.73	0.80
Raw score	0.70	0.78
2PJ	0.68	0.68

Table 4.17 Percentage of word similarity for BBC-News corpus

	Coherence	**Silhouette**
R-square	0.80	0.82
Raw score	0.96	0.98
2PJ	0.96	0.96

Table 4.18 Percentage of word similarity for 20 Newgroup-ol-6-1 corpus

	Coherence	**Silhouette**
R-square	0.60	0.67
Raw score	0.45	0.55
2PJ	0.63	0.68

Table 4.19 Percentage of word similarity for 20 Newgroup-sb-6-1 corpus

	Coherence	**Silhouette**
R-square	0.87	0.83
Raw score	0.80	0.73
2PJ	0.80	0.87

If we look at the similarity results for cases where the reference models are Silhouette and Coherence, we can conclude that there is a similarity among the categories of fitness functions. Generally, we can note that the fitness functions R-square, Raw score, and 2PJ are more similar to Silhouette. To conclude, there is a correspondence between the methods that hierarchically describe the corpora and the so-called general interpretation methods (Silhouette, Coherence).

Q5) Is it more advantageous to train a new GloVe instead of using a pretrained GloVe?
Since all our data sets comprise newspaper corpus, we could expect no significant differences in the results obtained when comparing the final score among different (new and pretrained) GloVe models used for the stability evaluation step. It is also interesting to note that despite the percentages of intersection shown in Table 4.7, the results obtained for the best run id selection are largely similar (see Table 4.14).

The choice between selecting a pre-trained model (Nguyen et al., 2016) and a model trained on one's corpus will depend on the context of the corpus. We used the model trained on Common Crawl data because our context was related to newspapers, which can be described as common knowledge. We believe that the most important thing is that GloVe has a bias (i.e., a knowledge of the domain in which we want to use it).

Q6) How do we select the final parameters?
Following the above analyses, selecting the best set of parameters for a corpus will depend on the depth of analysis the user would like to acquire (i.e., more or fewer topics). We have demonstrated the distinction between two main categories of fitness functions and shown their

power of interpretation. Selecting the final parameters will undoubtedly require a stability analysis; it will simply require selecting the maximum final score value.

4.7 Conclusion

The main objective of this paper was to pursue research to find near-optimal LDA parameters using Genetic Algorithms, as suggested by Panichella (2021). We tested five different fitness functions, including two traditional ones (Silhouette and Coherence), two less-known ones (R-square and Raw score), and one we developed (2PJ). LDA suffers from instability due to its generative and probabilistic nature (a point that several authors have mentioned). To overcome this shortcoming, we modified the order of the DTMs before each generation of the LDA model (addressing instability type-II). We regenerated a model with the same parameters several times to finally evaluate its replicability (addressing instability type-I).

Comparing the results obtained using different combinations of fitness functions showed an interesting advantage in discovering latent themes in a text. Traditional methods such as Silhouette and Coherence tend to discover more general topics than the R-square, 2PJ, and Raw score fitness functions, which discover many more latent topics. Using a hierarchical representation, we observed that these fitness functions are equivalent to traditional methods. Thus, using them offers a higher and more interesting descriptive potential for automatically analyzing topics in a text. Therefore, using the data sets proposed and studied by Greene *et al.* (2014), we showed that it is possible to obtain the same results in terms of the number of topics (K) when using the different GA-fitness function combinations (and after applying the pruning algorithm for the non-traditional fitness function).

Using an optimization algorithm to determine near-optimal parameters automatically may seem like a magic solution to a given problem. However, when using LDA, we have demonstrated the importance of adding a stability analysis step to evaluate the parameters' quality.

Regarding future work, the proposed methodology offers several avenues for improvement through its two distinct steps: the search for optimal parameters and stability analysis.

Regarding the search for parameters, we used the GA algorithm proposed by Panichella (2021). However, they also proposed other algorithms, such as SHC and SA, but they have the disadvantage of increasing the computation time. As for the implementation of GA in the R package, discussed in the article by (Scrucca, 2016), the author shows the advantage of applying a local search on the last iteration performed by GA to improve the search of parameters. A shortcoming observed is that the local search is not distributed/parallelized in this implementation, which increases computation time. We implemented a distributed version, but the results did not improve the quality of the parameters obtained but increased (threefold) the computation time of the full iterations of GA alone.

On the other hand, the package offers two different approaches for parallel computations: the master-slave (or global parallel GA approach) and the island parallel GA. Scrucca (2016) noted that using island-GA tends to be faster and offers superior results. This is due to the separation into islands, where each island can search very different regions of the search space, reinforcing the exploratory attitude of evolutionary algorithms.

We applied 20 independent LDA runs for the stability analysis, after which we applied the clustering algorithm. As mentioned in Mantyla *et al.,* (2018) study, it would be interesting to see the impact of increasing the number of runs. We used different GloVe models to project the word clusters into a vector space, but the current trend is for larger word embedding models such as Bert, Fastext, and GPT. We also created a final score metric to evaluate the quality of the clusters obtained. A modification or addition of new metrics could be added.

It might be interesting to compare the results of the topic-word matrices for the parameters discovered by the different combinations of GA-fitness functions to the ETM model (Dieng, Ruiz et Blei, 2020). This latter model performs better regarding the quality of discovered topics

and predictive performance than LDA and the LDA prototype model proposed by Rieger *et al.* (2020). As a final suggestion, a package composed of all these methodologies could be translated from R to Python since it seems to be a more fashionable programming language.

4.7.1 Acknowledgements

The research presented in this paper was financially supported by MITACS (Mathematics of Information Technology and Complex Systems) IT25026.

The authors thank the Fondation Les Petits Trésors, the Laurent Mottron laboratory from the University of Montreal, the LiNCS laboratory of Ecole de Technologie Supérieure, and IBM Canada Center for Advanced Studies (CAS) for their support.

CHAPITRE 5

OPTIMISATION D'UN MODÈLE DE LANGUE BASÉ SUR LES SOUS-TITRES EXTRAITS DES VIDÉOS EN LIGNE

Ce chapitre détaille la démarche qui permet d'atteindre le troisième sous-objectif de ce projet de recherche qui a pour but de sélectionner le modèle de langue optimal qui nous servira de base pour le calcul des recommandations. Ce chapitre se traduit également comme faisant partie de la deuxième composante de notre système de recommandation présenté à la Figure 2.2.

Au chapitre 3, nous avons identifié trois valeurs candidates pour segmenter les sous-titres des épisodes en plus de proposer cinq techniques de substitution à appliquer sur la nomenclature qui a servi à nommer les principaux personnages de la télésérie. Ces deux considérations nous ont permis de créer un total de 15 sac-de-mots différents. Afin d'en extraire les sujets latents, nous avons appliqué le cadriciel proposé au chapitre 4. De plus, ce cadriciel permet d'évaluer quantitativement la paramétrisation d'un modèle par l'analyse en stabilité. Bien que cette analyse permette de comparer différentes modélisations issues d'un même sac-de-mots à l'aide du score en stabilité, elle ne permet pas de comparer différentes modélisations issues de sac-de-mots différents.

Afin de recourir à d'autres éléments de comparaison qui nous permettront de mieux aiguiller notre sélection finale, nous avons développé une approche mettant en relation le partage des termes et des documents les plus probables de décrire les distributions sujet-termes et document-sujets. De plus, cette approche nous permet d'évaluer, outre que par la comparaison des scores en stabilité, l'impact des différentes modifications apportées à la construction des sac-de-mots sur la modélisation issue de LDA.

Les prochaines sections de ce chapitre préciseront la méthodologie appliquée et détailleront quelques statistiques sur les différents sac-de-mots utilisés ainsi que sur les résultats obtenus afin de conclure quant à la sélection du modèle final.

5.1 Méthodologie

Comme préalablement mentionné, nous nous appuyons sur la méthodologie décrite au chapitre 4 afin d'identifier une paramétrisation quasi optimale de LDA pour les différents sac-de-mots. Cette méthodologie se scinde en 3 étapes distinctes auxquelles nous en ajoutons une 4^{e}. Cette étape additionnelle est nécessaire, car les trois premières étapes sont spécifiques à l'optimisation d'un modèle pour un jeu de données (sac-de-mots), et que nous voulons être en mesure de comparer les différents modèles issus des différents sac-de-mots entre eux.

Ces 4 étapes distinctes sont les suivantes :

- Étape 1 : Appliquer les étapes de prétraitement sur le corpus afin de construire les différents sac- de-mots
- Étape 2 : Optimiser le modèle LDA pour chacun des sac-de-mots
- Étape 3 : Évaluer la stabilité des modèles
- Étape 4 : Évaluer le niveau de partage d'éléments communs entre les différents sujets au sein des distributions finales

À la suite de ces étapes, nous allons être en mesure de commencer la discussion sur la sélection du modèle final. La discussion ainsi que la sélection du modèle final se trouvent à la section 5.2.

5.1.1 Étape 1 : Appliquer les étapes de prétraitement sur le corpus afin de construire les différents sac-de-mots

Les données utilisées sont les différents sac-de-mots identifiés au chapitre 3 et sont résumées au Tableau 5.1 suivant. Au total, nous avons évalué 15 sac-de-mots différents. Ces derniers sont tous basés sur les sous-titres de la série pour enfant, mais sur lesquels nous avons fait varier le nombre de répliques à enchaîner pour concevoir un document, en plus d'appliquer différentes techniques de substitution relative à la nomenclature utilisée pour nommer les personnages principaux (voir Tableau 3.15).

Les étapes de prétraitement appliquées sur le corpus sont celles mentionnées au Tableau 3.4 présenté à la section 3.1.3. Par la suite, nous avons appliqué la technique de dénombrement *term-frequency (TF)* afin de générer nos sac-de-mots. De plus, afin de favoriser la stabilité de LDA, nous avons appliqué la technique de l'émondage relatif sur les termes ayant une fréquence d'apparition inférieure à deux.

Tableau 5.1 Résumé des différents sac-de-mots à l'étude

Identifiant de la technique de substitution pour nommer les personnages principaux	**Nombre de répliques enchaînées pour concevoir un document**					
	7		**9**		**10**	
	Dimension du sac-de-mots		**Dimension du sac-de-mots**		**Dimension du sac-de-mots**	
	Nombre de documents	Nombre de termes constituant le vocabulaire	Nombre de documents	Nombre de termes constituant le vocabulaire	Nombre de documents	Nombre de termes constituant le vocabulaire
Ts-1	403	436	320	436	287	436
Ts-2	403	426	320	426	287	426
Ts-3	402[†]	425	320	425	287	425

Ts-4	403	436	320	436	287	436
Ts-5	403	427	320	427	287	427

†Tel qu'indiqué par cette valeur (402) différente des autres dans de cette colonne (403), le nombre de documents peut être affecté par la contrainte du nombre de termes minimaux pour concevoir un document (voir section 3.2.2 pour plus de détails).

5.1.2 Étape 2 : Optimiser le modèle LDA pour chacun des sac-de-mots

L'optimisation des paramètres de LDA est guidée par la combinaison de l'Algorithme Génétique et de la fonction objective *R-Square*. Cette fonction objective a été sélectionnée pour son potentiel à découvrir un plus grand nombre de thèmes sous-jacents au corpus à l'étude comparativement aux mesures traditionnellement utilisées (*Coherence* et *Silhouette*), comme le montre l'article du chapitre 4 (Harel, Velazquez-Godinez et Ratté, 2023, (soumis)).

Les tableaux suivants montrent les paramètres utilisés pour l'Algorithme Génétique (AG) et les différents intervalles de recherches associés aux paramètres à optimiser de LDA. De plus, nous justifions l'intervalle de recherche (entre 0 et 1) du paramètre α par notre méthode de création des documents qui devrait favoriser l'association d'un nombre minimal de sujets par document. Nous justifions l'intervalle de recherche (entre 0 et 1) du paramètre β par notre sélection de la fonction objective. Comme mentionné ci-dessus, cette dernière favorise la découverte d'un plus grand nombre de sujets (K) et donc l'obtention de sujets plus spécialisés. Puisque nous avons peu de termes dans le vocabulaire et que nous avons sélectionné la fonction objective *R-Square*, cela nous parait adéquat qu'un regroupement de mots soit décrit par peu de termes significatifs. Finalement, à cause du caractère stochastique de ce processus d'optimisation, nous effectuons 50 exécutions indépendantes qui nous permettent d'évaluer la distribution du nombre de sujets découverts.

Tableau 5.2 Paramétrisation de l'Algorithme Génétique

Paramètre	**Valeur**
Taille de la population *(GA population size)*	10
Nombre d'itérations *(number of GA iteration)*	5
Probabilité de croisement (pc) *(binary simulated crossover with probability)*	0,9
Probabilité de mutation (pm) (*polynomial mutation with mutation probability*)	0,25
Index de mutation (ηm) *(mutation index)*	20

Tableau 5.3 Intervalle de recherche des paramètres de LDA

Paramètre	**Intervalle de recherche**
α	0,0000001-1.0
β	0,0000001-1.0
K	25-45
N (nombre d'itérations du *Gibbs sampling*)	200-300

Tableau 5.4 Exécution indépendante

Paramètre	**Valeur**
Nombre d'exécutions	50

En somme, nous nous retrouvons avec 50 jeux de paramètres (K, α, β et N) par sac-de-mots. Pour déterminer quel jeu de paramètres nous devons utiliser, nous allons effectuer l'analyse de stabilité décrite à la section suivante.

5.1.3 Étape 3 : Évaluer la stabilité des modèles

Cette étape permet d'évaluer quantitativement la stabilité d'un modèle LDA inféré à partir des paramètres quasi optimaux découverts lors de la recherche heuristique. La méthodologie appliquée est la même que celle proposée à l'étape 3 de la méthodologie décrite dans l'article présenté au chapitre 4. C'est-à-dire que chaque paramétrisation préalablement identifiée est utilisée afin d'instancier un modèle. Puisque l'évaluation de la stabilité du modèle consiste à déterminer si ces paramètres permettent une reproductibilité de la distribution de la matrice sujet-termes, nous utilisons ces paramètres pour créer 25 matrices sujet-termes. Afin de décrire les sujets latents issus des distributions sujet-termes, nous avons sélectionné les 10 termes les plus probables de chacun des sujets. Ensuite, nous avons utilisé le modèle de plongement de mots GloVe préentraîné offert par Stanford afin de projeter les termes dans un espace vectoriel. Finalement, nous avons utilisé le modèle k-médoïdes pour identifier les regroupements de mots afin d'appliquer les différentes analyses qui permettent le calcul du score en stabilité.

À la fin de cette troisième étape, nous aurons calculé, pour chacun des jeux de paramètres identifiés par la recherche heuristique (50) et pour chacun des différents sac-de-mots (15), un total de 750 scores en stabilité.

5.1.4 Étape 4 : Évaluer le niveau de partage d'éléments communs entre les différents sujets au sein des distributions finales

Dans un contexte de maximisation, la décision quant à la sélection entre une valeur plus élevée et une autre est un problème trivial. A priori, nous pourrions sélectionner le modèle final en sélectionnant celui qui a la valeur en stabilité la plus élevée parmi les 750 modèles. Cependant, l'analyse en stabilité se concentre à évaluer si la paramétrisation permet une reproductibilité des ensembles de mots les plus probables décrivant chaque sujet, mais ne considère pas la seconde distribution que LDA génère, c'est-à-dire la distribution document-sujets.

L'idée de cette quatrième étape est de mettre en relation ces deux distributions qui sont définies par LDA. Cela nous créera un autre critère de sélection de la modélisation finale et si possible, permettra de visualiser l'impact des différentes modifications apportées pour la génération des différents sac-de-mots. Cette approche consistera donc à calculer le niveau de partage des éléments communs pour les éléments les plus probables décrivant la matrice sujet-termes et dans la matrice document-sujets. Par la suite, nous agglomérerons ces deux types de niveaux de partage.

Pour ce faire, nous avons considéré une approche qui, dans un premier temps, calcule le niveau de partage des éléments communs entre les différents sujets (les termes). Prenons pour exemple une distribution sujet-termes contenant quatre sujets, où chacun des sujets est décrit par le top 4 des termes les plus probables tel que représenté au tableau suivant.

Tableau 5.5 Exemple des 4 termes les plus probables de décrire 4 différents sujets

	Terme 1	**Terme 2**	**Terme 3**	**Terme 4**
Sujet-1	a	b	c	d
Sujet-2	e	f	a	b
Sujet-3	a	y	z	h
Sujet-4	g	h	c	b

Le calcul des éléments communs entre les différents sujets s'effectue en comparant les termes d'une colonne entre eux. Pour chacune des colonnes, nous avons quatre possibilités (un terme par sujet). Supposons qu'on s'intéresse au top un (les termes de la colonne Terme 1), nous pouvons remarquer que le terme « a » apparait deux fois et que ce terme est partagé entre les sujet-1 et sujet-3. Nous dirons alors que le terme « a » est répliqué une seule fois. Nous pouvons alors calculer le ratio entre le nombre de termes répliqués par rapport au nombre maximum de répétitions. Ce nombre maximum est calculé comme étant le nombre n sélectionné de termes décrivant le top n multiplié par le nombre de sujets moins un (i.e. n * (K-1)). Pour cet exemple, ce ratio est de 1/3. À noter que le dénominateur correspond au nombre maximum de répétitions

et non au nombre d'éléments dans la matrice afin d'avoir des résultats variant entre 0 et 1 pour tous les modèles sous analyse.

Si on s'intéresse au top quatre, cela produit un ensemble de 16 termes et un nombre maximum de répétitions de 12 (i.e. 4 * (4-1)). Dans notre exemple, nous pouvons remarquer que les termes « a », « b », « c » et « h » sont répliqués. Par exemple, les termes « c » et « h » apparaissent 2 fois donc ces termes ont été répliqués une autre fois dans un autre sujet. Pour les termes « a » et « b », ceux-ci apparaissent 3 fois, donc ces termes ont été répliqués dans deux autres sujets. Si on calcule le ratio entre le nombre de termes répliqués par rapport au nombre de possibilités, nous obtenons une valeur de 6/12. Le tableau suivant présente les différents résultats après avoir fait varier le top n des termes les plus probables de 1 à 4.

Tableau 5.6 Calcul du ratio du nombre de termes répliqués

Top n	**Nombre de termes répliqués**	**Nombre maximum de répétitions**	**Ratio**
1	1	3	0,33
2	1	6	0,17
3	3	9	0,33
4	6	12	0,50

L'intuition quant au calcul de ce ratio est qu'il nous permet de quantifier le niveau de partage des éléments communs entre les différents sujets en fonction du top n. Un modèle idéal serait que pour un top n élevé, disons quarte pour rester dans le contexte de cet exemple, le ratio soit nul. Cela signifierait alors que chaque regroupement de mots décrivant chaque sujet est décrit par des termes spécifiques.

Dans un deuxième temps, nous appliquons cette même procédure sur la distribution document-sujets, mais en nous attardant aux documents ayant les probabilités les plus élevées d'appartenir à un sujet. Supposons les résultats présentés au tableau ci-dessous.

Tableau 5.7 Calcul du ratio du nombre de documents répliqués

Top n	Nombre de documents répliqués	Nombre maximum de répétitions	Ratio
1	1	3	0,33
2	3	6	0,50
3	4	9	0,44
4	7	12	0,58

Un modèle idéal serait que pour un top n élevé, le ratio soit nul. Cela signifierait alors que pour chaque sujet, il y a des documents spécifiques.

Dans un troisième temps, afin de mettre en relation les différents ratios calculés, nous les représentons dans un espace à deux dimensions tel que présenté à l'image ci-dessous et calculons les surfaces engendrées.

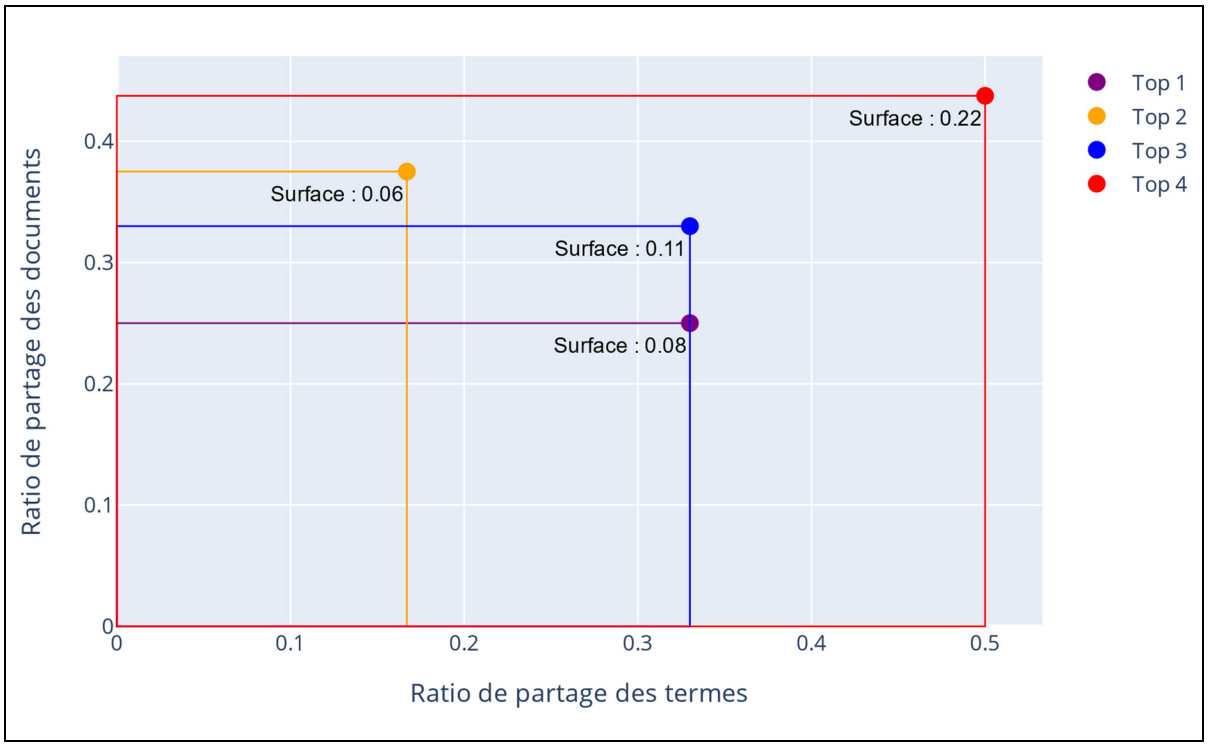


Figure 5.1 Exemple de représentation des ratios de partage

L'aire de la surface est utilisée pour quantifier le niveau de partage des éléments communs pour cette modélisation et sera utilisée pour comparer différentes modélisations entre elles. L'idée sera de comparer la valeur de surface engendrée des différentes modélisations en faisant varier n (du top n). Nous chercherons donc à identifier une modélisation qui minimise cette surface. De plus, nous croyons que cette approche nous permettra d'observer l'effet sur les distributions finales des différentes modifications apportées sur les sac-de-mots issus de la modélisation faite par LDA.

Finalement, grâce à l'analyse de stabilité discutée à la section précédente et à l'analyse des éléments en commun discutée dans cette section, nous pouvons débuter la discussion sur la sélection du modèle final.

5.2 Résultats et discussion

Cette section aborde les différents résultats obtenus à la suite de l'application de la méthodologie préalablement décrite. La discussion sera orientée vers les trois critères de sélection utilisés qui guideront le choix du modèle final. Ces trois critères sont les suivants :

1. Maximiser le nombre de sujets (K) ;
2. Minimiser les ratios de partages ;
3. Maximiser le score en stabilité.

Ces critères ont pour but de sélectionner un modèle ayant plusieurs sujets spécialisés. Ces critères sont en lien avec le choix de la fonction objective, nos intervalles de valeurs pour les paramètres de LDA et avec les valeurs candidates du nombre de répliques enchaînées.

Les premiers résultats présentés sont ceux obtenus après les étapes de recherche des paramètres quasi optimaux (étape 2) et d'évaluation de la stabilité des modèles (étape 3). La figure suivante illustre la distribution du nombre de sujets K identifiés par la recherche heuristique des 50 exécutions pour les différents sac-de-mots (15).

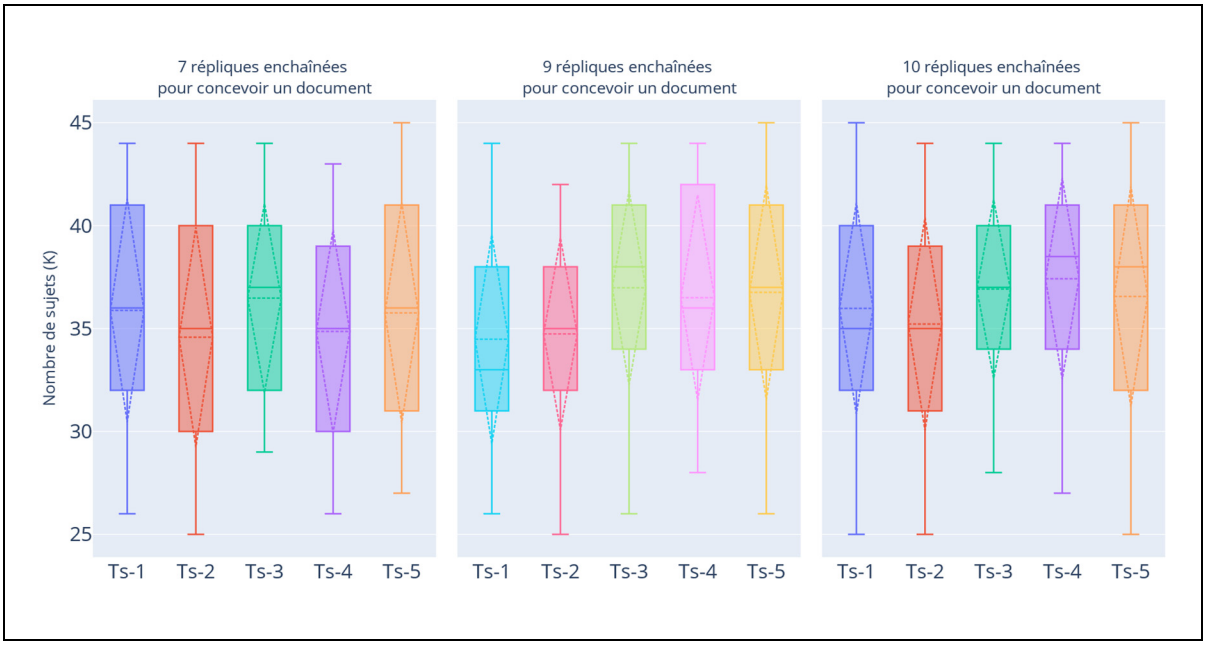


Figure 5.2 Distribution du nombre de sujets (K) identifiés par la recherche heuristique pour les différents sac-de-mots

Par cette représentation, nous pouvons remarquer que le nombre de sujets moyens, identifiés par les traits pointillés horizontaux, évolue entre 34 et 37 pour les différents sac-de-mots. Cela nous laisse croire que l'intervalle de recherche associé à ce paramètre (25-45) est suffisant.

La figure suivante illustre la distribution des résultats des scores en stabilité issue de la modélisation des différents sac-de-mots.

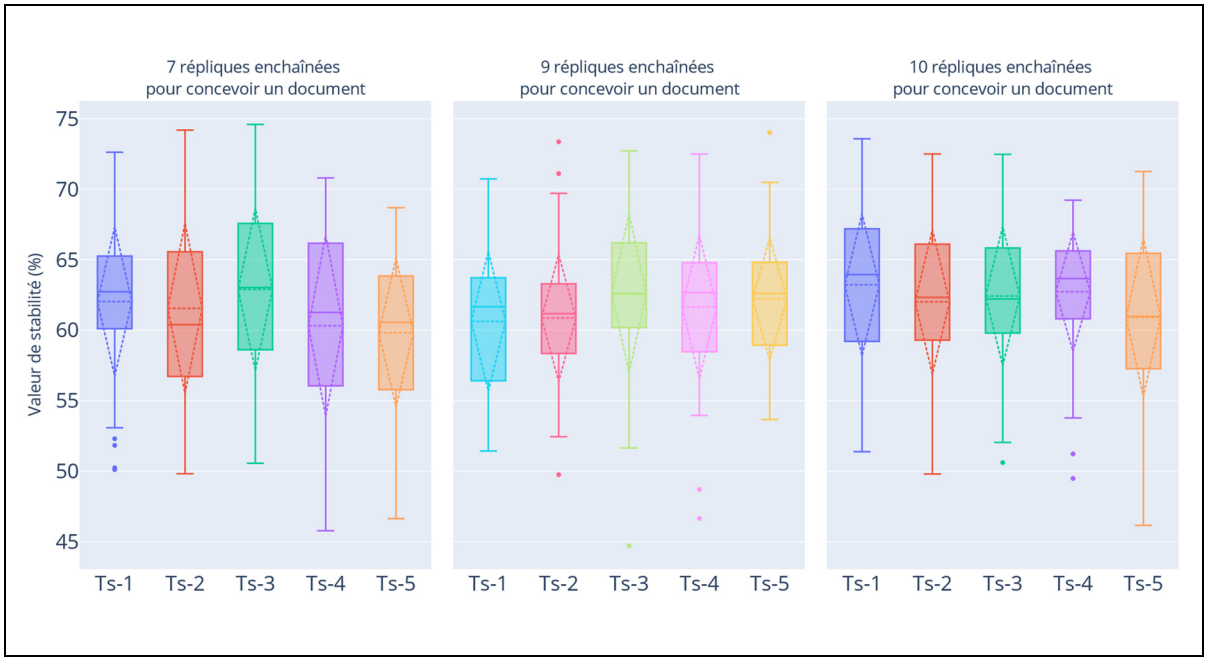


Figure 5.3 Distribution des scores en stabilité des différentes modélisations

Par cette représentation, nous pouvons remarquer que les scores en stabilité moyens, identifiés par les traits pointillés horizontaux, évoluent entre 60 et 64 pourcents pour la modélisation des différents sac-de-mots.

Basées sur les critères du nombre de sujets et de scores en stabilité, les modélisations sous Ts-3 semblent se démarquer légèrement des autres (Ts-1-2-4-5). En effet, nous observons que

dans ces 2 figures, pour les deux premiers cadrans (nombre de répliques de 7 et 9), les modèles sous Ts-3 ont les moyennes les plus élevées. Aussi, nous observons que les modèles sous Ts-3 ont les quartiles 1 les plus élevés ou les deuxièmes plus élevés parmi les 15 sac-de-mots.

Ces deux représentations nous ont permis de brosser un profil général des résultats de la distribution du nombre de sujets et des scores en stabilité. Cependant, elles ne nous permettent pas d'identifier clairement les impacts des différentes modifications apportées sur la création des sac-de-mots. Puisque nous cherchons à identifier des modélisations qui ont des scores en stabilité élevés, nous avons sélectionné les trois meilleurs résultats en stabilité pour les différents sac-de-mots. Nous identifions par la couleur jaune le score le plus élevé, en rouge le second score le plus élevé et en noir le troisième score le plus élevé tel qu'illustré à la figure ci-dessous.

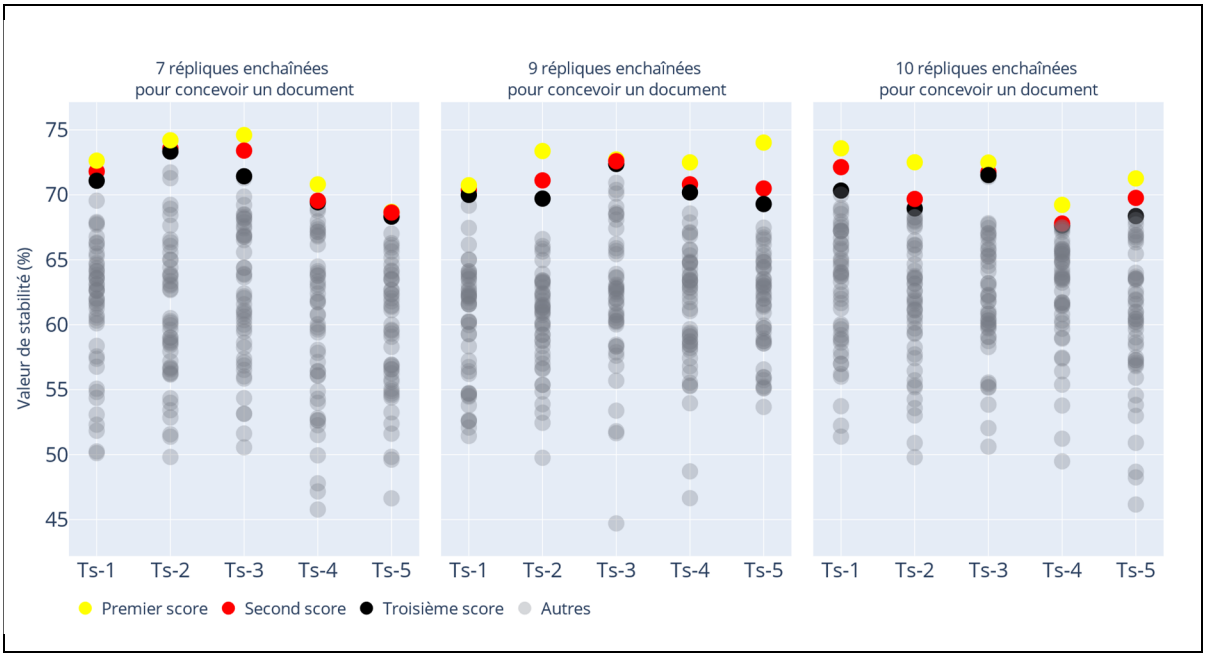


Figure 5.4 Identification des trois scores en stabilité les plus élevés

Ensuite, nous avons identifié le nombre de sujets K associés à ces valeurs de stabilité élevées.

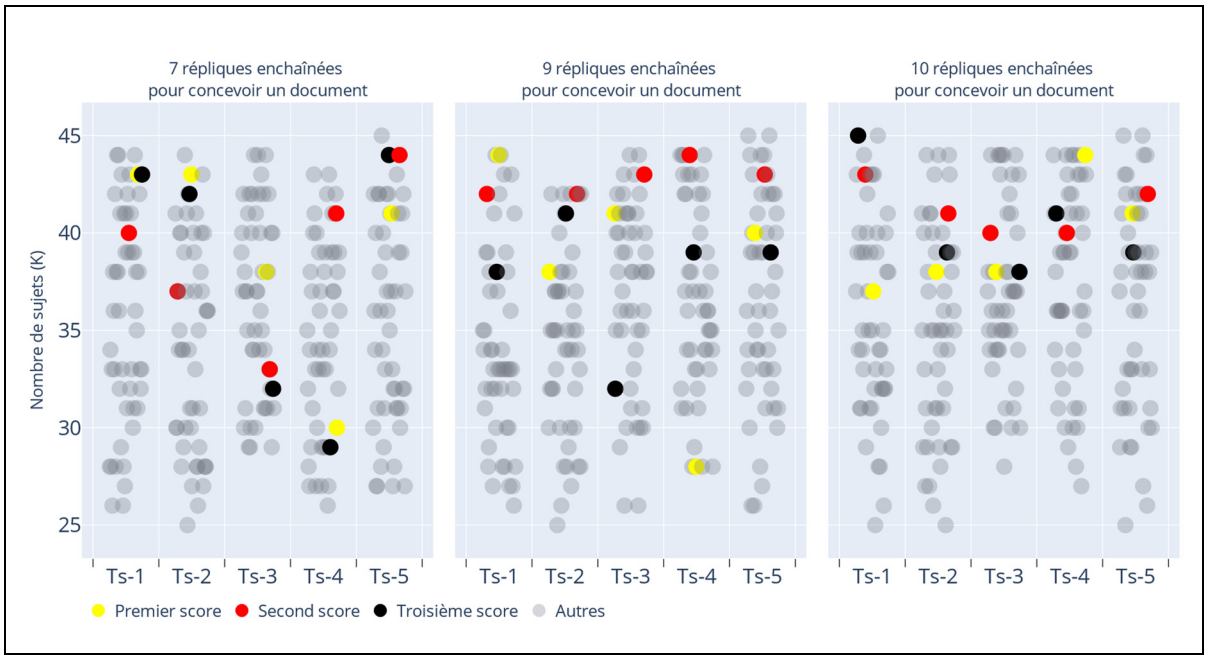


Figure 5.5 Identification du nombre de sujets (K) associés aux scores en stabilité les plus élevés

Par ces représentations, nous pouvons remarquer qu'après le filtrage sur les trois meilleurs scores en stabilité, l'ensemble des plus hautes valeurs évoluent entre 67 et 75 % (Figure 5.4) et que ces valeurs sont associées à un nombre de sujets évoluant entre 28 et 45 avec une majorité au-dessus de 35 (Figure 5.5). Nous pouvons également remarquer que l'augmentation du nombre de répliques à enchaîner pour concevoir un document fait augmenter les valeurs K des modèles les plus stables. Par exemple, dans le cadran de gauche de la Figure 5.5 (7 répliques enchaînées pour concevoir un document), on remarque 4 pastilles sous la barre des 35, dans le cadran du centre (9 répliques enchaînées pour concevoir un document), on remarque deux pastilles sous la barre de 35 et, dans le cadran de droite (10 répliques enchaînées pour concevoir un document), elles sont toutes au-dessus. Par cette représentation nous poserons comme critère qu'un modèle doit contenir un minimum de 35 sujets.

Passons aux résultats de l'étape 4 où l'idée est de quantifier le niveau de partage des éléments communs dans l'ensemble de mots les plus probables décrivant chaque sujet ainsi que le niveau

de partage des documents appartenant à chaque sujet. Tel que décrit dans cette méthodologie, nous devons sélectionner un top n des éléments les plus probables. Pour des fins d'illustration, prenons le sac-de-mots issu des caractéristiques Ts-5 et 7 répliques enchaînées pour concevoir un document. Les 50 modélisations sont présentées au graphique suivant par des triangles. Les triangles de couleurs jaune, rouge et noir réfèrent aux modèles ayant les scores en stabilité les plus élevés. Nous représentons la valeur moyenne des ratios par le point jaune. Ce centroïde nous donne la position des niveaux de partages (termes et documents) moyens des 50 modélisations pour un top 10 des éléments les plus probables. Cette représentation nous permet de dire qu'en moyenne pour ce sac-de-mots, le ratio de partage des termes est des 0,2867 et que le ratio de partage des documents est de 0,0958, ce qui représente une surface moyenne de 0,0275 (0,2867 x 0,0958).

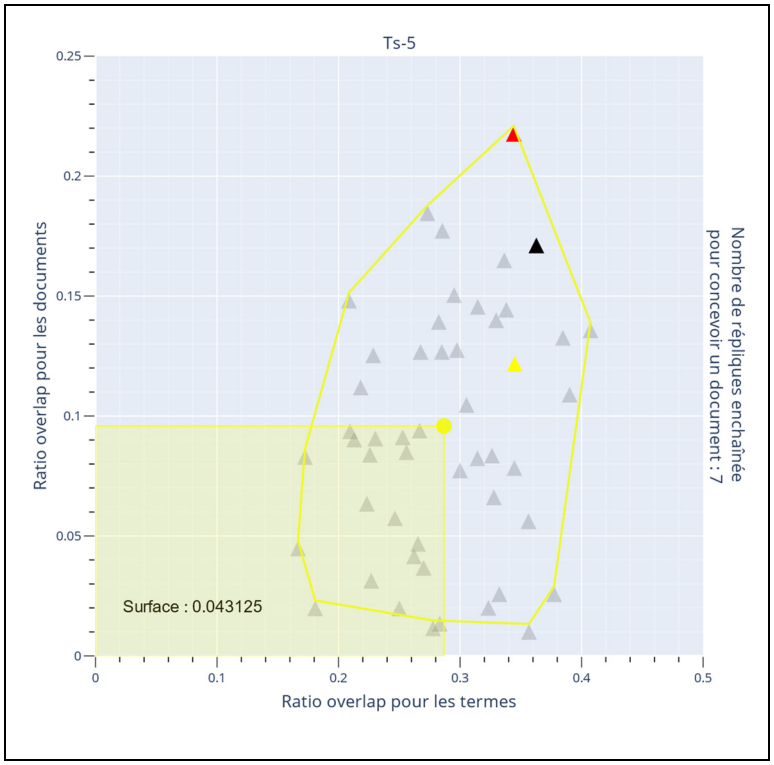


Figure 5.6 Ratios de partage pour 50 modèles à top 10

Le graphique suivant montre l'évolution de cette forme en faisant passer le top n de trois à dix pour les 15 différents sac-de-mots. Plus la valeur moyenne d'un modèle tend vers le coin inférieur gauche, plus la surface est petite et plus le modèle sera favorisé. L'analyse de cette représentation nous permettra de sélectionner une technique de substitution (Ts).

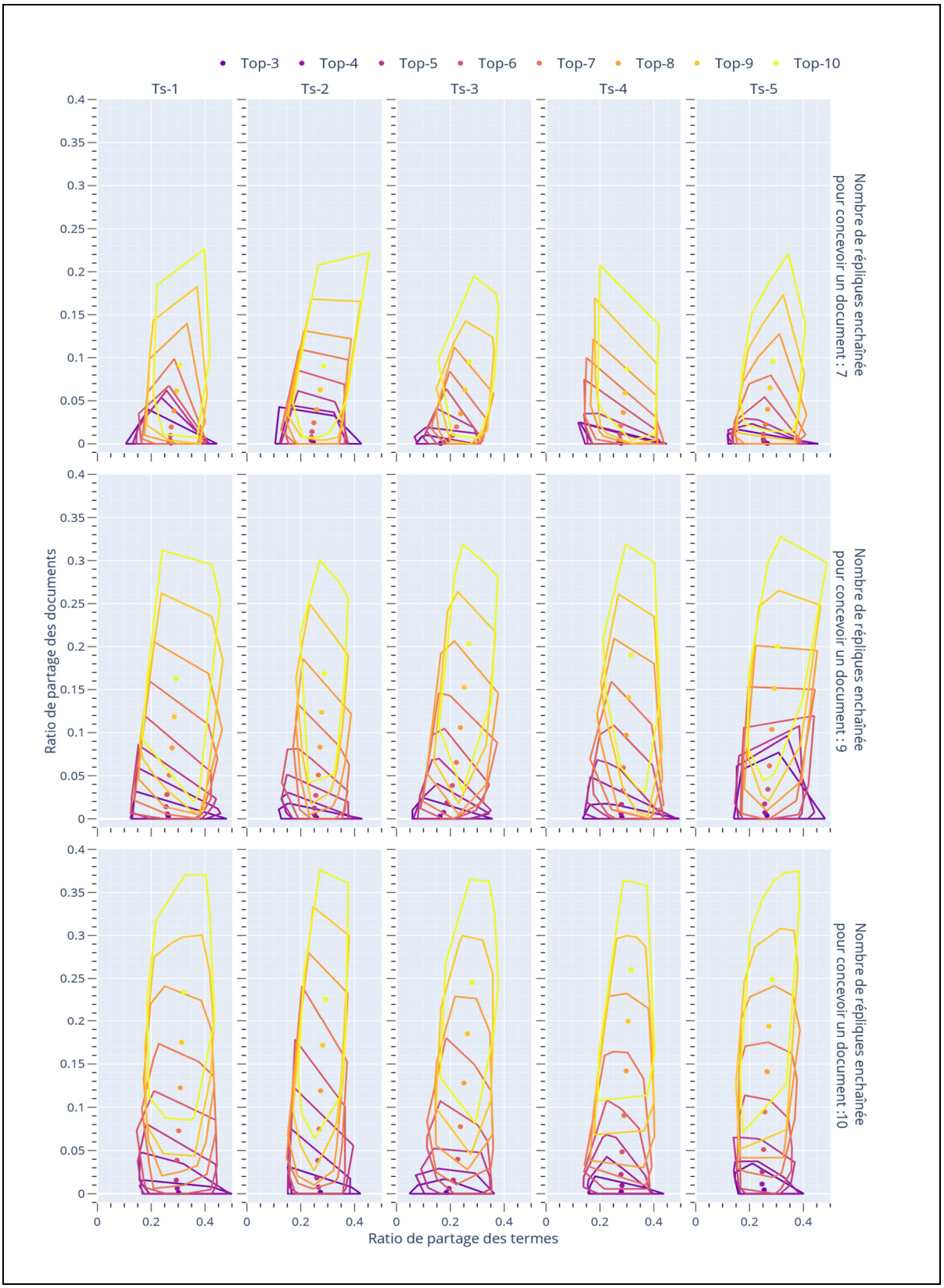

Top-3
Top-4
Top-5
Top-6
Top-7
Top-8
Top-9
Top-10
Ts-1
Ts-2
Ts-3
Ts-4
Ts-5
Nombre de répliques enchaînée pour concevoir un document : 7
Nombre de répliques enchaînée pour concevoir un document : 9
Nombre de répliques enchaînée pour concevoir un document :10
Ratio de partage des documents
Ratio de partage des termes
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4

Figure 5.7 Ratios de partage pour toutes les modélisations

Si on s'attarde à l'évolution des différents centroïdes à la suite de l'augmentation du top n, nous pouvons remarquer que plus le nombre de répliques enchaînées pour concevoir un document augmente plus la valeur du ratio de partage des documents augmente. Ce comportement est attendu, car plus nous augmentons le nombre de répliques pour concevoir un document, plus on diminue le nombre de documents total définissant les sac-de-mots (voir Tableau 5.1). Cela a donc comme conséquence d'augmenter les chances qu'un document soit partagé. En ce qui concerne le ratio de partage des termes lors de l'augmentation du nombre de répliques enchaînées, cette augmentation semble avoir un impact minime sur ce ratio, mais a tout de même un effet. Nous pouvons remarquer que l'alignement des centroïdes se décale généralement vers la droite lorsque le nombre de répliques augmente, c'est-à-dire que le ratio de partage augmente.

Si on compare l'évolution des centroïdes par rapport aux différentes Ts, c'est-à-dire les différentes colonnes, nous pouvons remarquer que les centroïdes de Ts-3 ont des ratios de partage des termes moins élevés pour les valeurs de top 3 à top 8 comparativement aux autres Ts.

Si on s'attarde aux surfaces entourant chaque centroïde, nous pouvons remarquer que c'est sous la colonne Ts-3 que ces surfaces sont les plus petites. Tel que présenté au tableau suivant, nous avons calculé la moyenne des surfaces engendrées des différents nombres de répliques enchaînées (7-9-10) par top n.

Tableau 5.8 Surface entourant les centroïdes par Ts en fonction des différents de top n

	Ts-1	**Ts-2**	**Ts-3**	**Ts-4**	**Ts-5**
Top 3	0,704	0,623	**0,540**	0,651	0,648
Top 4	0,669	0,556	**0,555**	0,616	0,596
Top 5	0,640	0,569	**0,525**	0,588	0,576
Top 6	0,659	0,592	**0,539**	0,605	0,598
Top 7	0,681	0,638	**0,605**	0,675	0,660
Top 8	0,736	0,678	**0,640**	0,705	0,718
Top 9	0,812	0,732	**0,684**	0,739	0,770
Top 10	0,850	0,776	**0,737**	0,776	0,823

De la même façon, nous présentons au tableau suivant les valeurs de l'aire des surfaces engendrées par les différents ratios en utilisant les centroïdes comme point de référence.

Tableau 5.9 Valeur des surfaces engendrées par les centroïdes pour les différents Ts et top n

	Ts-1	**Ts-2**	**Ts-3**	**Ts-4**	**Ts-5**
Top 3	0,0005	0,0005	**0,0003**	0,0005	0,0008
Top 4	0,0013	0,0014	**0,0009**	0,0016	0,0017
Top 5	0,0031	0,0032	**0,0026**	0,0041	0,0040
Top 6	0,0071	0,0068	**0,0063**	0,0087	0,0083
Top 7	0,0136	0,0132	**0,0125**	0,0164	0,0156
Top 8	0,0238	**0,0218**	**0,0218**	0,0272	0,0258
Top 9	0,0356	**0,0332**	0,0344	0,0404	0,0384
Top 10	0,0501	**0,0468**	0,0497	0,0559	0,0530

Par cette analyse et les résultats présentés au Tableau 5.8 et au Tableau 5.9, nous croyons que Ts-3 est un choix raisonnable par rapport aux autres techniques de substitution, car c'est cette technique de substitution qui accorde les résultats les plus bas.

Afin de sélectionner le nombre de répliques à enchaîner pour concevoir un document, nous présentons au tableau suivant les résultats des surfaces engendrées par les centroïdes à Ts-3 pour les différents nombres de répliques.

Tableau 5.10 Valeur des surfaces engendrées par les centroïdes à Ts-3 en fonction du nombre de répliques

	top n							
Nombre de répliques enchaînées pour concevoir un document	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
7	**0,000**	**0,000**	**0,001**	**0,002**	**0,004**	**0,008**	**0,016**	**0,026**
9	0,000	0,001	0,003	0,008	0,015	0,025	0,038	0,054
10	0,000	0,001	0,003	0,009	0,018	0,032	0,049	0,069

Par ces résultats nous sélectionnons le nombre de répliques de 7, car c'est le modèle qui a des surfaces plus basses pour tous les tops n. De plus, nous sélectionnons le modèle représenté par la pastille jaune comparativement à ceux représentés par les pastilles de couleur noire et rouge, car elle a un nombre de sujets (K) supérieur à 35 (voir Figure 5.5).

Pour conclure, le modèle de langue qui servira de base pour les différents calculs de génération de recommandations sera celui avec la technique de substitution 3 (suppression du nom du personnage) et un nombre de répliques à enchaîner pour construire un document de 7. De plus,

nous pouvons également remarquer que c'est ce modèle qui a obtenu le plus haut niveau score en stabilité (voir Figure 5.4).

5.3 Conclusion

L'objectif principal de ce chapitre était de sélectionner une modélisation que nous jugerons adéquate pour concevoir la base de nos calculs de recommandations vidéo. Pour ce faire, nous avons appliqué la méthode d'optimisation développée au chapitre 4 sur les 15 différents sac-de-mots. Ces sac-de-mots sont issus des sous-titres de la première saison de *Peppa Pig*, mais diffèrent par les différentes caractéristiques définissant la longueur des documents (3) et par les différentes techniques de substitution des noms de personnages (5).

Afin de sélectionner notre modèle final, nous avons posé les 3 critères de sélection suivants : maximiser le nombre de sujets (K), minimiser le partage d'éléments communs entre les différents sujets au sein des distributions finales et maximiser le score en stabilité.

La métrique du partage d'éléments communs est une métrique que nous avons développée qui met en relation les termes et les documents les plus probables de décrie les distributions sujet-termes et document-sujets. Son apport vient combler le fait que l'analyse en stabilité se concentre à évaluer si la paramétrisation permet une reproductibilité des ensembles de mots les plus probables décrivant chaque sujet, mais qui ne considère pas la seconde distribution que LDA génère, c'est-à-dire la distribution document-sujets. L'idée directrice de minimiser cette métrique est d'identifier une modélisation qui minimise le partage de termes les plus probables entre les différents regroupements de mots et qui minimise le nombre de documents partageant les mêmes sujets principaux.

Finalement, le modèle de langue qui servira de base pour les différents calculs de génération de recommandations est celui issu de la technique de substitution 3 (suppression du nom du personnage) et un nombre de répliques à enchaîner pour construire un document de 7.

CHAPITRE 6

LE SYSTÈME DE RECOMMANDATIONS

Ce chapitre détaille la démarche qui permet d'atteindre les sous-objectifs 4 et 5 de ce projet de recherche. Globalement, ces sous-objectifs concernent l'élaboration des différents algorithmes nécessaires pour le calcul des recommandations et du développement de l'application Web qui a permis l'évaluation du système de recommandations. Le sous-objectif 4 se traduit comme étant la troisième composante de notre schéma des principales composantes de notre système de recommandations présenté à la Figure 2.2. À la suite de la complétion de cette composante, nous pourrons répondre au sous-objectif 5 et ainsi, évaluer le système de recommandations et la cohérence du modèle de langue via des tests utilisateurs. L'analyse des résultats de ces tests se fera au chapitre suivant.

Pour ce faire, nous débuterons avec une courte explication sur l'implémentation de ce système (serveurs, technologies et principales librairies), brosserons un portrait général de ses différentes composantes et, finalement, détaillerons les composantes clés (estimation du centre d'intérêt, carte sémantique et détails pour le développement des tests d'intrusion).

6.1 Choix technologiques

En ce qui concerne le développement logiciel, notre système de recommandations est déployé sur un serveur Web constitué de deux entités distinctes. La première, la partie client, est la portion de l'application que l'utilisateur voit à l'écran et avec laquelle il interagit. La seconde, la partie serveur, est celle qui authentifie les différents utilisateurs, qui effectue les différents calculs de recommandations et qui sauvegarde les résultats.

Une des principales motivations qui nous a poussés à développer une application Web était de faciliter son accessibilité afin d'accueillir un maximum de participants. De plus, dans un futur

développement éventuel, cela permettrait de collecter des informations telles que le temps nécessaire pour répondre à une question, le nombre de clics effectués et même toutes autres données implicites pertinentes qui permettraient de mieux cerner les comportements de l'utilisateur.

D'un point de vue du développement Web, il existe une multitude de technologies disponibles pour concrétiser ce projet. Les principaux cadriciels utilisés pour le développement des parties de notre système sont énumérés au tableau suivant et, afin de rendre ce système accessible au public, les parties client et serveur ainsi que la base de données ont été déployées sur le service d'hébergement d'Hérokou.

Tableau 6.1 Cadriciels

	Cadriciel	Description	Url
Partie client	Angular	Angular est un cadriciel de développement d'application Web gratuit. Le code est ouvert (*open source)* et son langage de programmation est le TypeScript. Il est dirigé par l'équipe Angular de Google et par une communauté de particuliers et d'entreprises.	https://angular.io/
	D3.js	D3.js est une bibliothèque utilisant le langage de programmation JavaScript qui permet de produire des visualisations de données dynamiques et interactives dans les navigateurs Web.	https://d3js.org/
Partie serveur	Python Flask	Flask est un cadriciel de développement Web utilisant le langage de programmation Python.	https://github.com/pallets/flask/
Base de données	PostgreSQL	PostgreSQL est un système de gestion de bases de données relationnelles libre et gratuit qui met l'accent sur l'extensibilité et la conformité SQL.	https://www.postgresql.org/

6.2 Système global

Dans cette section, nous brosserons une vue générale des différentes composantes des parties client et serveur du système. La figure suivante permet de visualiser les différentes parties du système et de leurs interactions. De plus, cette figure sera réutilisée et bonifiée au fil des prochaines sections à titre de support visuel.

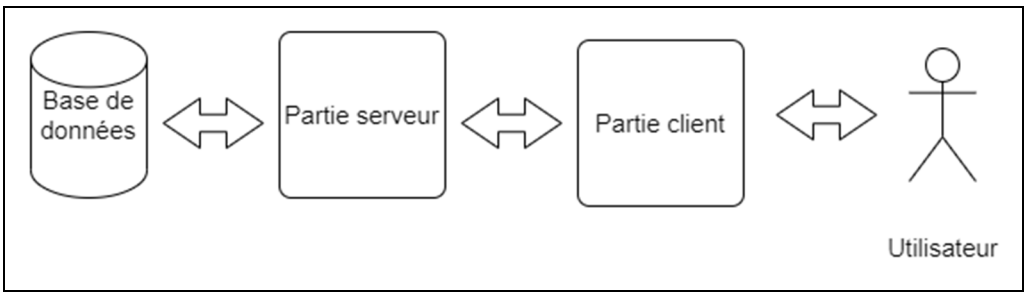


Figure 6.1 Système global

6.2.1 Partie client

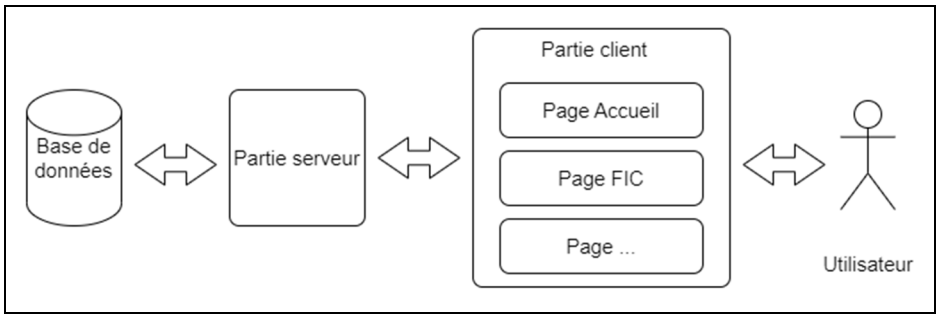


Figure 6.2 Partie client du système global

Comme préalablement mentionnée, la partie client a pour objectif de servir d'interface pour l'utilisateur afin qu'il puisse interagir avec la partie serveur (figure 6.2). Dans un but de simplifier les explications, nous décrivons l'application comme étant une suite de pages Web distinctes où certaines conditions doivent être complétées afin de passer à la suivante. La figure

suivante permet d'illustrer sous la forme d'un diagramme d'activités les conditions à respecter (les boîtes ovales et flèches) pour passer aux différentes pages (les boîtes rectangulaires). Les prochains paragraphes décrivent ces différentes pages.

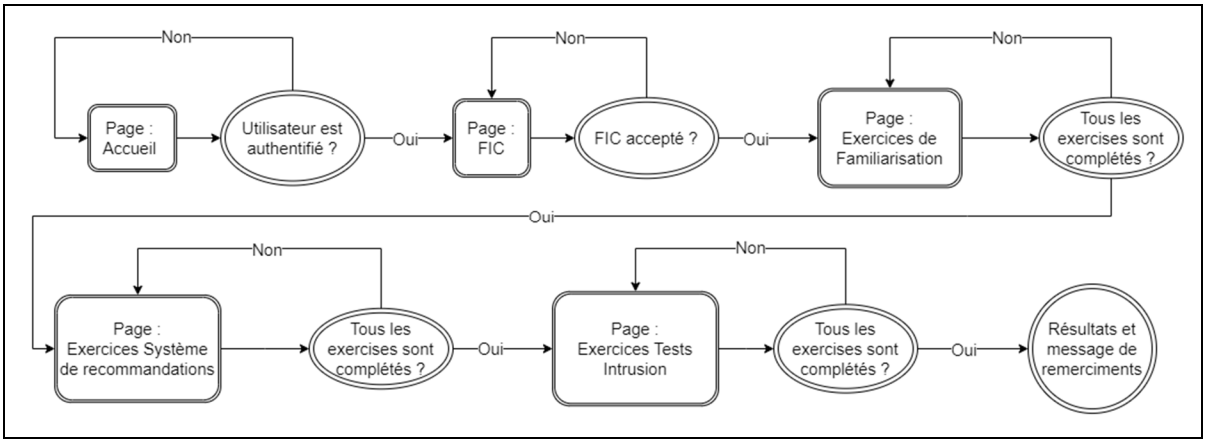


Figure 6.3 Application Web - partie client

Page : Accueil

Très simple dans sa présentation, cette page contient un message d'accueil et un bouton d'authentification qui démarrera, une fois l'authentification complétée, une session pour l'utilisateur. Le mécanisme d'authentification utilise un environnement fédéré, c'est-à-dire que l'authentification des utilisateurs est séparée de l'accès à l'application. Cela se fait via une entité externe fournissant une authentification indépendante des informations d'identification des utilisateurs. Cette approche permet de simplifier le développement, de rendre l'authentification plus sécuritaire, de réduire au minimum la nécessité d'administrer les utilisateurs et d'améliorer l'expérience utilisateur de l'application. Pour ce faire, nous avons utilisé les services de Google Identity[20].

Une fois l'authentification effectuée, l'utilisateur est dirigé à la page du FIC.

[20] https://developers.google.com/identity/sign-in/web/server-side-flow?fbclid=IwAR3OTth4Knx-DryfGvl3E85EYT_aJp_Kw9vtCTbxAIJPqlFXoe6bHEr3jK8

Page : FIC

Cette page décrit le Formulaire d'Information et de Consentement (FIC) que l'utilisateur doit accepter afin de débuter l'expérimentation.

Une fois le FIC accepté, l'utilisateur est dirigé à la page d'exercices de familiarisation.

Page : Exercices de Familiarisation

Nous avons jugé nécessaire que l'utilisateur effectue une première expérimentation avec les composantes clés du système avant qu'il débute les tâches liées à l'évaluation de ce dernier. Cela permettra à l'utilisateur de se familiariser avec les personnages de la télésérie, le niveau de langage utilisé et les fonctionnalités de la visionneuse, d'interagir avec la carte sémantique et d'avoir un aperçu du type de questions que nous allons lui poser.

Pour ce faire, nous avons considéré les étapes clés suivantes :

1. Niveau de familiarisation avec la télésérie

   Puisque notre population cible provient du milieu universitaire et que ces vidéos ont été réalisées pour un public de jeune âge, nous avons jugé nécessaire de demander à l'utilisateur s'il est familier ou non avec cette télésérie (figure 6.4). Nous supposons que les utilisateurs expérimentés seront mieux placés pour interpréter les ensembles de mots identifiés par LDA et de ce fait, ils devraient être plus performants lors des tests d'intrusion.

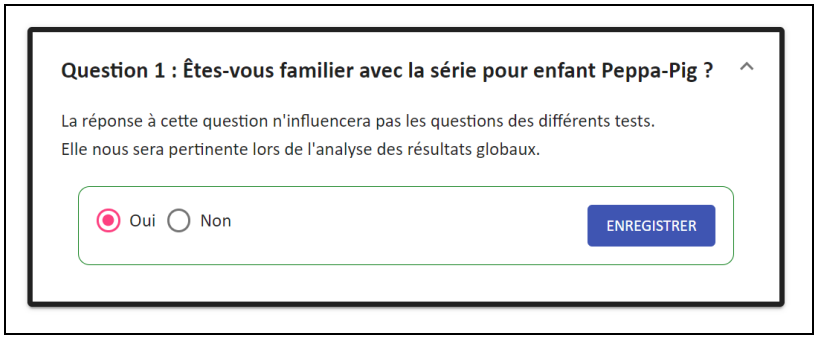


Figure 6.4 Familiarisation avec la télésérie *Peppa Pig*

2. Visionnement de courtes séquences vidéo et association de chaque vidéo à un ensemble de mots le décrivant le mieux.

   Cette étape consiste à visionner cinq courtes séquences vidéo d'environ 30 secondes chacune. Cela permet à l'utilisateur de se familiariser avec le niveau linguistique de la télésérie, les principaux personnages et la visionneuse. Après l'écoute de chaque séquence, un choix de réponse lui est proposé afin qu'il associe la séquence à un ensemble de mots la décrivant le mieux (figure 6.5).

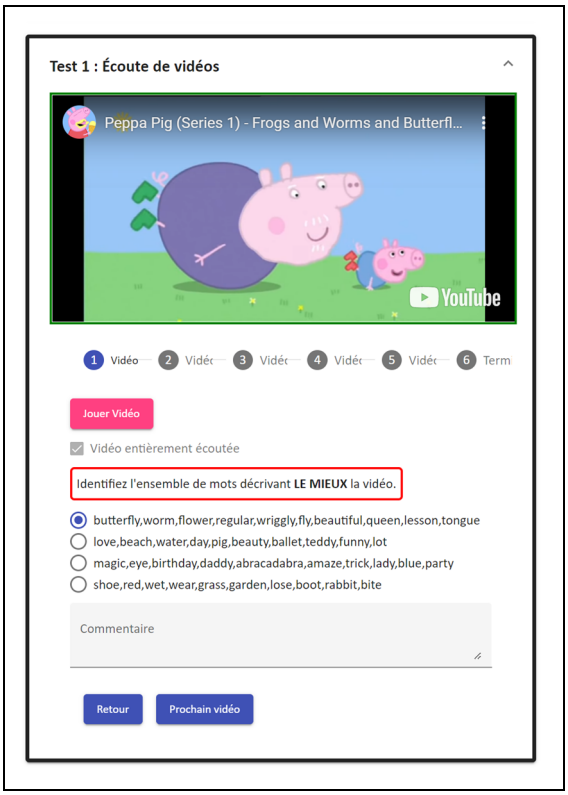


Figure 6.5 Visionneuse et question

3. Exploration des différentes fonctionnalités de la carte sémantique.

   La carte sémantique est construite sous la forme d'un graphe non orienté (représentations abstraites de réseaux composés de sommets reliés par des arêtes). Dans cette visualisation (voir image ci-dessous), le graphe est constitué de sommets (boules rouges et vertes) reliés par des arêtes (traits roses).

Les boules vertes représentent des ensembles de mots cohérents issus du modèle LDA sélectionné associés à un identifiant de sujet (boules rouges). Les liens n'ont aucun poids et si un lien existe entre deux nœuds, celui-ci est formé par le partage d'un terme entre ces regroupements.

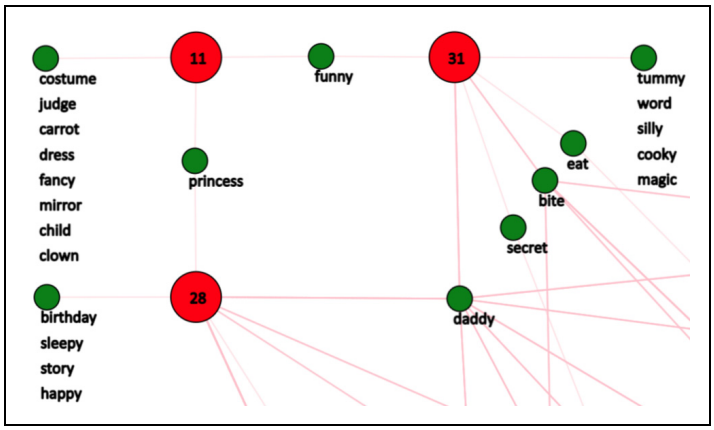


Figure 6.6 Carte sémantique

En prenant pour exemple l'image ci-dessus, la thématique 11 est directement liée à trois sommets verts décrits par les termes suivants :

- funny
- princess
- costume, judge, carrot, dress, fancy, mirror, child, clown

Un nœud vert contenant qu'un seul terme signifie que ce terme est partagé entre plusieurs thématiques. Comme il est possible de le remarquer, le terme « *funny* » est partagé entre les thématiques 11 et 31 et le terme « *princess* » est partagé entre les thématiques 11 et 28.

Nous définissons ce graphe comme étant une carte sémantique, car il expose les liens entre différents ensembles de termes sémantiquement similaires. L'utilisation du terme « carte » au lieu du terme « graphe » est employée, car combinée à un système de recommandations,

la carte permettra de positionner les centres d'intérêt de l'utilisateur déduits par le système de recommandations.

Pour ce qui a trait aux exercices, ceux-ci consistent à explorer les différentes fonctionnalités comme la mise en évidence d'ensemble de mots, à déterminer le terme commun à deux ensembles et à tester des fonctionnalités usuelles d'une carte telles qu'agrandir, réinitialiser, etc. Plus de détails concernant sa création seront présentés à la section 6.3.2.

Une fois tous les exercices de familiarisation complétés, l'utilisateur peut passer à l'étape suivante.

Page : Exercices Système de recommandations

Cette page permet d'évaluer si les distributions issues de la modélisation des sous-titres des vidéos faites par LDA sont adaptées à l'élaboration d'un système de recommandations vidéo ainsi que l'apport visuel qu'offre la carte sémantique.

Les exercices dans cette page se déroulent dans l'ordre suivant :

1. Sélectionner un regroupement de mots décrivant un centre d'intérêt de départ.

Figure 6.7 Sélection d'une thématique de départ

2. Spécifier les termes de prédilection contenus dans le regroupement de mots préalablement sélectionné.

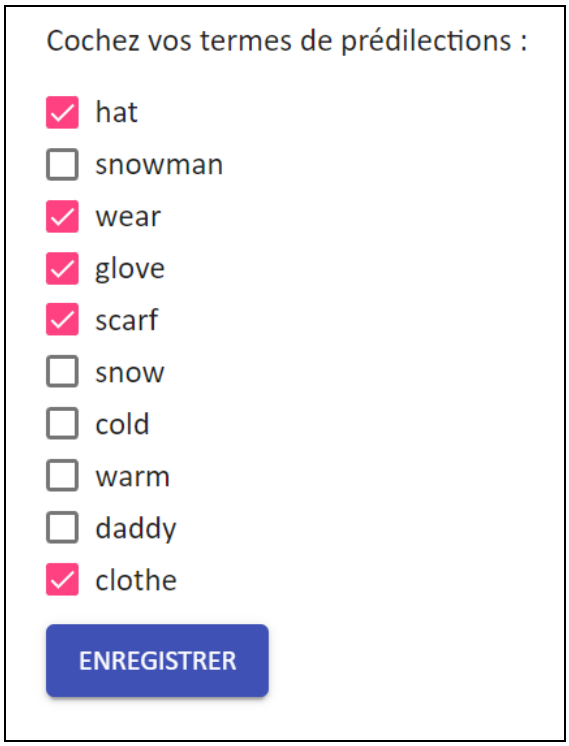


Figure 6.8 Termes de prédilections

3. Le centre d'intérêt initial est ajouté sur la carte (voir boule jaune figure 6.9).

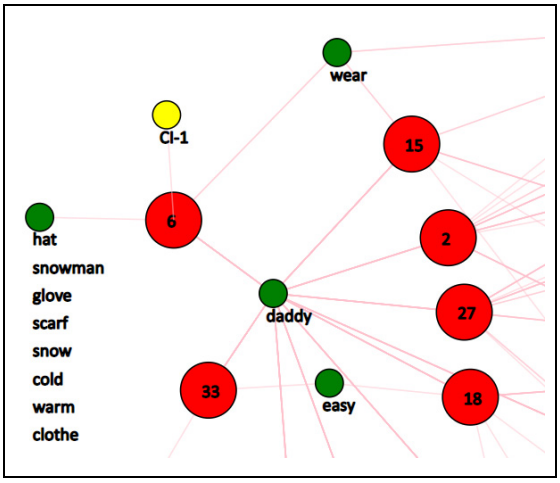


Figure 6.9 Carte sémantique et centre d'intérêt initial

4. Écouter une première série de vidéos (cinq) et mentionner si les termes de prédilections sont pertinents avec la séquence vidéo.

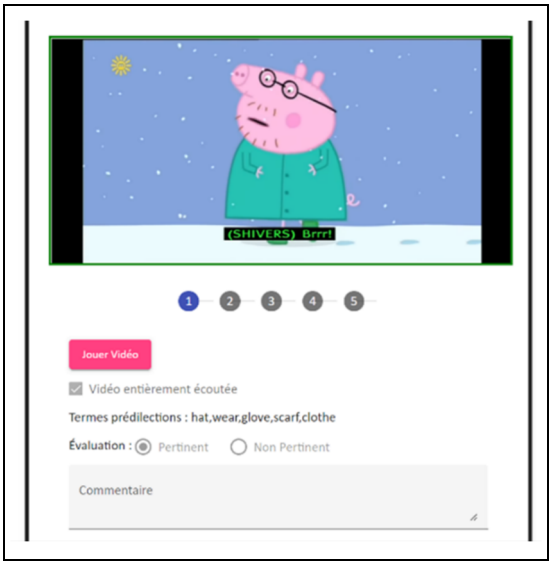


Figure 6.10 Écoute de vidéos

5. Le système calcule une nouvelle série de recommandations et ajoute un point illustrant l'estimation du nouveau centre d'intérêt de l'utilisateur sur la carte sémantique (voir boule c-2 figure 6.11).

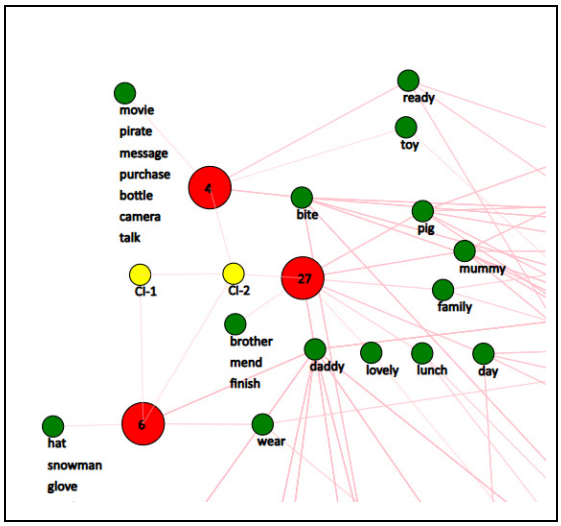


Figure 6.11 Carte sémantique et ajout
d'un nouveau centre d'intérêt

6. Le système répète les étapes 4 et 5, pour un nombre maximum d'itérations fixé à cinq, ou peut arrêter lorsque le système n'est plus en mesure d'émettre de nouvelles recommandations.

7. Répondre à quelques questions concernant la qualité des recommandations ainsi que sur l'interprétation de la carte sémantique à l'aide d'une échelle de Likert qui a les gradations suivantes : « oui, neutre et non ».

   - Question 1 : Est-ce que les recommandations concordent avec votre thématique de départ ?
   - Question 2 : Parmi les ensembles de mots que vous venez de mettre en évidence (boules vertes), est-ce que ces mots, sans nécessairement tous les sélectionner, vous permettraient de décrire le contenu des vidéos que vous venez de regarder ?
   - Question 3 : Est-ce que les ensembles de mots (boules vertes) liés à votre dernier centre d'intérêt (boule jaune ayant l'indice numérique le plus élevé) vous permettraient de décrire le contenu des vidéos que vous avez qualifiés de pertinent ?
   - Question 4 : Au-delà des choix de recommandation faits par l'algorithme, trouvez-vous pertinent de voir l'estimation de votre centre d'intérêt fait par le système de recommandations au travers de cette carte ?
   - Question 5 : Selon vous, serait-il pertinent de visualiser les vidéos en les sélectionnant à partir de la carte ?
   - Question 6 : Si cette interface était intégrée sur Netflix, est-ce que l'utiliseriez ?
   - Question 7 : Qu'est-ce que vous amélioriez sur cette carte ?

Une fois complétée, l'utilisateur est dirigé à la prochaine section.

Page : Exercices Tests d'intrusion

Basés sur les tests d'intrusion développés par Chang *et al*. (2009), ces derniers exercices permettront d'évaluer la qualité de notre modèle sélectionné à l'aide d'utilisateurs externes. Cette évaluation nous permettra donc de valider le potentiel de la méthodologie développée au chapitre 4 pour la recherche d'une paramétrisation quasi optimale de LDA. Cet exercice est constitué de deux tests, soit le mot intrus (*word intrusion*) et le sujet intrus (*topic intrusion*), qui sont sous la forme de choix de réponse.

Le test du mot intrus (*word intrusion*) consiste à identifier le terme intrus parmi une liste de quatre termes pour un total de 38 questions (figure 6.12).

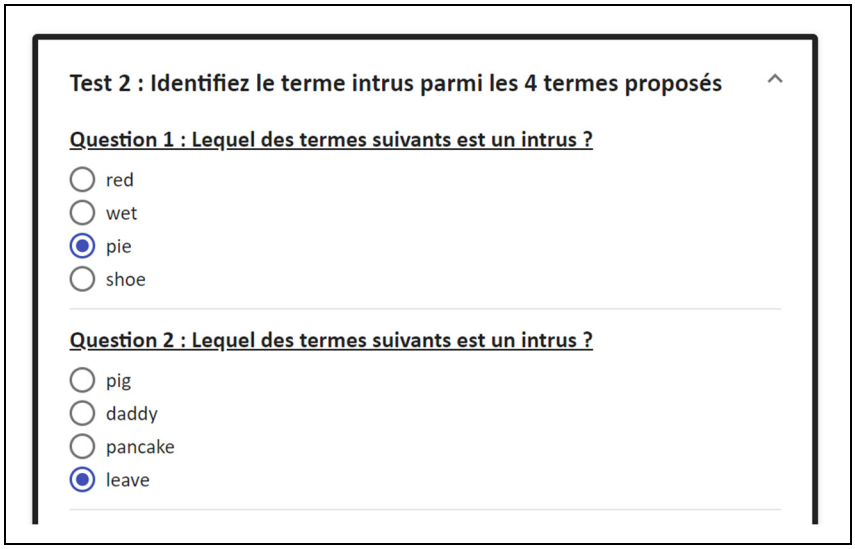


Figure 6.12 Exemple du test du mot intrus (word intrusion)

Le test du sujet intrus (*topic intrusion*) consiste à écouter 20 séquences vidéo et d'identifier pour chacune d'elle le regroupement de mots la décrivant le mieux (figure 6.13).

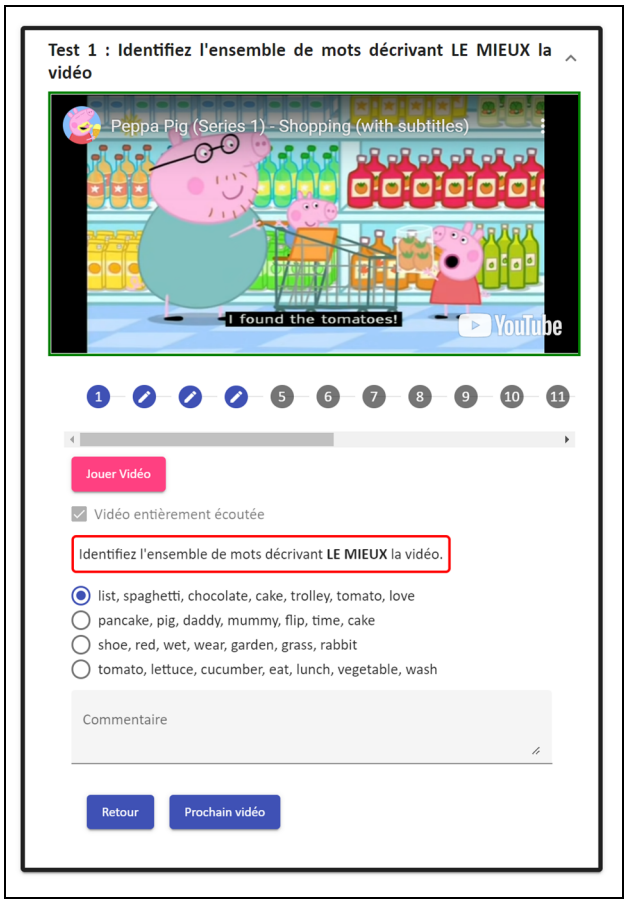


Figure 6.13 Exemple du test du regroupement de mots intrus (topic intrusion)

Plus de détails seront présentés à la section 6.3.3.

6.2.2 Partie serveur

La partie serveur a pour principal objectif de fournir les informations nécessaires à la partie client suite aux choix de l'utilisateur, d'exécuter les différents calculs intermédiaires afin d'estimer le centre d'intérêt de l'utilisateur, d'émettre de nouvelles recommandations ainsi que de sauvegarder ces informations dans la base de données. Le schéma présenté à la figure 6.14 permet d'illustrer la séparation des concepts généraux de la conception et de la communication entre la partie client, jusqu'à la base de données en passant par la partie serveur.

L'idée principale est que la partie serveur est découpée en plusieurs API où chacune de ces routes est spécialisée pour les différentes pages de l'interface utilisateur. Selon la provenance de la requête (la route empruntée), différents calculs sont effectués avant de sauvegarder les résultats dans la base de données sous leur table respective.

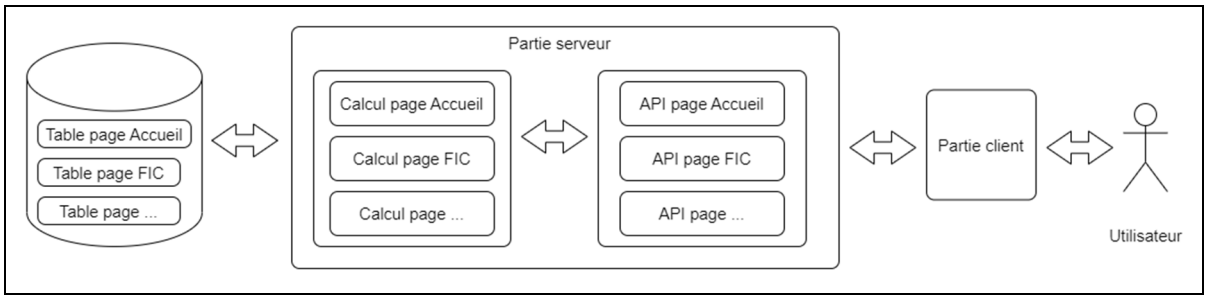


Figure 6.14 Partie serveur du système global

6.3 Composantes clés

Ayant brossé un portrait général du fonctionnement de notre système de recommandations, cette section s'attardera aux détails des composantes clés dans l'ordre suivant :

1. Calcul des recommandations et estimation du centre d'intérêt de l'utilisateur
2. Création de la carte sémantique
3. Détails pour le développement des tests d'intrusion

6.3.1 Calcul des recommandations et estimation du centre d'intérêt de l'utilisateur

Le cœur du système de recommandations concerne le problème d'optimisation à résoudre afin d'émettre une recommandation. L'approche que nous proposons repose sur la technique de filtrage sur le contenu. Elle combine une approche de filtrage sur les résultats issus du modèle de langue et sur les informations contenues dans un modèle représentant l'historique de visualisation de l'utilisateur. La solution est inspirée de celle proposée par Zhu *et al.* (2013) qui est principalement basée sur l'utilisation de la distribution document-sujets issue de

l'entrainement de LDA, mais pour laquelle nous avons ajouté la considération de la pertinence des vidéos (considérer l'évaluation des vidéos faite par l'utilisateur).

En définissant les termes suivants pour un ensemble de M séquences vidéo indépendantes que nous dénotons par $D = (d_1, d_2, \ldots, d_M)$, LDA décrit un document (les sous-titres d'une séquence vidéo) par une distribution des sujets latents que nous dénotons par $Z = (z_1, z_2, \ldots, z_K)$ où K est le nombre total de sujets et z_k est caractérisé par une distribution du vocabulaire par sujet.

La distribution des sujets d'une vidéo, notée comme la probabilité de l'ensemble des sujets Z sachant une séquence vidéo d_i, se traduit comme suit : $P(Z|d_i)$. En considérant un historique H de visualisation de G vidéos pour un usager, $H = (h_1, h_2, \ldots, h_G)$, il est possible d'estimer un centre d'intérêt en calculant la distribution moyenne des sujets de l'historique des vidéos.

$$P(Z|H) = \frac{1}{G}\sum_{i=1}^{i=G} P(Z|d_i) \tag{6.1}$$

En posant l'hypothèse qu'un utilisateur souhaite regarder des vidéos dont le contenu correspond à ses préférences, le problème de la recommandation de vidéos peut être formalisé en un problème d'optimisation pour lequel on cherche à trouver des vidéos ayant des distributions de sujets similaires à la distribution des préférences de l'utilisateur (centre d'intérêt). En dénotant d_r comme une vidéo à recommander (parmi la collection de M vidéos) et en calculant sa distance au centre d'intérêt, il est possible de sélectionner la vidéo la plus près du centre d'intérêt. Dans le contexte de cette étude, nous avons sélectionné la mesure de Manhattan pour calculer cette distance, car c'est cette mesure qui a été priorisée par Zhu *et al.* (2013). En vertu de cette formulation, nous pouvons traduire comme suit :

$$\arg\min_{d_r \in M} \| P(Z|d_r) - P(Z|H) \| \tag{6.2}$$

Cependant, l'équation qui permet d'estimer le centre d'intérêt de l'utilisateur assume que les vidéos de l'ensemble H (historique des recommandations) sont toutes classifiées comme étant des vidéos pertinentes, ce qui n'est usuellement pas le cas. De plus, nous croyons que les vidéos qui seraient classifiées comme non pertinentes sont également porteuses d'information. C'est-à-dire qu'elles identifient les sujets z_k auxquels nous devons diminuer les probabilités pour les prochaines recommandations.

Pour ce faire, nous considérons les $P(Z|d_i)$ comme des coordonnées où chaque Z_k correspond à une dimension vectorielle. Sachant que l'historique de visualisation est constitué de P vidéos classifiées comme pertinentes et NP vidéos classifiées comme non pertinentes, nous calculons le vecteur moyen de chacun de ces sous-ensembles.

$$\overrightarrow{Pertinent} = \frac{1}{P}\sum_{i=1}^{i=P} P(Z|d_i) \tag{6.3}$$

$$\overrightarrow{Non\ Pertinent} = \frac{1}{NP}\sum_{i=1}^{i=NP} P(Z|d_i) \tag{6.4}$$

Puisque nous voulons nous éloigner des sujets z_k non pertinents pour la mise à jour du centre d'intérêt, nous calculons la différence entre le vecteur pertinent et non pertinent que nous dénotons par Δ.

$$\Delta = \overrightarrow{Pertinent} - \overrightarrow{Non\ Pertinent} \tag{6.5}$$

Ensuite, le nouveau centre d'intérêt peut être calculé en additionnant Δ au vecteur pertinent. Nous pouvons résumer le calcul de la mise à jour du centre d'intérêt de l'utilisateur par l'équation suivante :

$$\overrightarrow{Centre\ d'intérêt} = \overrightarrow{Pertinent} + \Delta \tag{6.6}$$

Les prochaines lignes décrivent les détails de ce nouveau calcul d'estimation du centre d'intérêt de l'utilisateur à l'aide d'un exemple.

Supposons une distribution document-sujets pour laquelle le nombre de sujets (K) est de trois. La figure 6.15 illustre les différents vidéos (point gris) dans l'espace vectoriel où les axes représentent les $P(z_k|d_i)$ bornées entre 0 et 1 inclusivement.

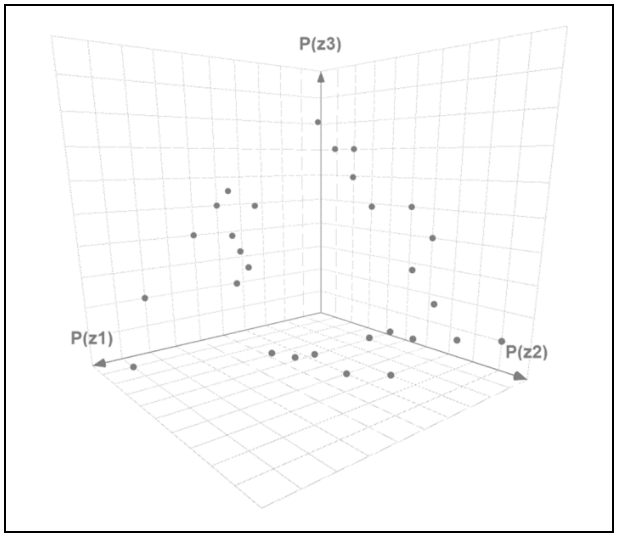


Figure 6.15 Représentation des documents

Supposons que la première série de recommandations demandée par un utilisateur concerne des vidéos où le sujet z_1 est prédominant. Le système recommandera une sélection de vidéos (cinq) ayant la $P(z_1)$ la plus élevée dans l'ensemble des vidéos. Dans l'exemple de la figure

6.16, nous utilisons une sélection de sept vidéos; ceux-ci sont représentés à l'aide des points bleus.

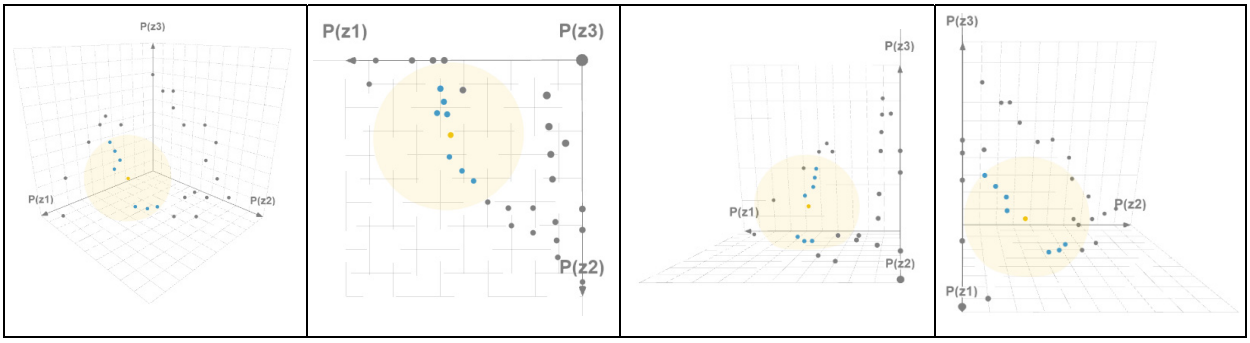


Figure 6.16 Représentation des documents au centre d'intérêt initial

Cette première série de recommandations constitue le premier appart à l'historique de visualisation de l'utilisateur. Après le visionnement de ces vidéos, supposons que l'utilisateur classifie quatre vidéos comme étant pertinentes (Tableau 6.2) et trois vidéos comme étant non pertinentes (Tableau 6.3). La première étape du calcul d'estimation du nouveau centre d'intérêt consiste à déterminer les vecteurs moyens de chacun de ces ensembles.

Tableau 6.2 Vidéos classifiées comme étant pertinentes

Pertinent					
	$P(z_1)$	**$P(z_2)$**	**$P(z_3)$**	**Somme**	**Domaine des valeurs possibles**
Vidéo 1	0,50	0,10	0,40	1,00	[0:1]
Vidéo 2	0,50	0,20	0,30	1,00	[0:1]
Vidéo 3	0,55	0,20	0,25	1,00	[0:1]
Vidéo 4	0,50	0,15	0,35	1,00	[0:1]
Vecteur pertinent	0,51	0,16	0,33	1,00	[0:1]

Tableau 6.3 Vidéos classifiées comme étant non pertinentes

Non pertinent					
	$P(z_1)$	**$P(z_2)$**	**$P(z_3)$**	**Somme**	**Domaine des valeurs possibles**
Vidéo 5	0,50	0,46	0,04	1,00	[0:1]
Vidéo 6	0,45	0,50	0,05	1,00	[0:1]
Vidéo 7	0,55	0,40	0,05	1,00	[0:1]
Vecteur non pertinent	0,50	0,45	0,05	1,00	[0:1]

Nous représentons aux images de la figure 6.17 les vidéos considérées comme pertinentes sous la sphère verte et les non pertinentes sous la sphère rouge.

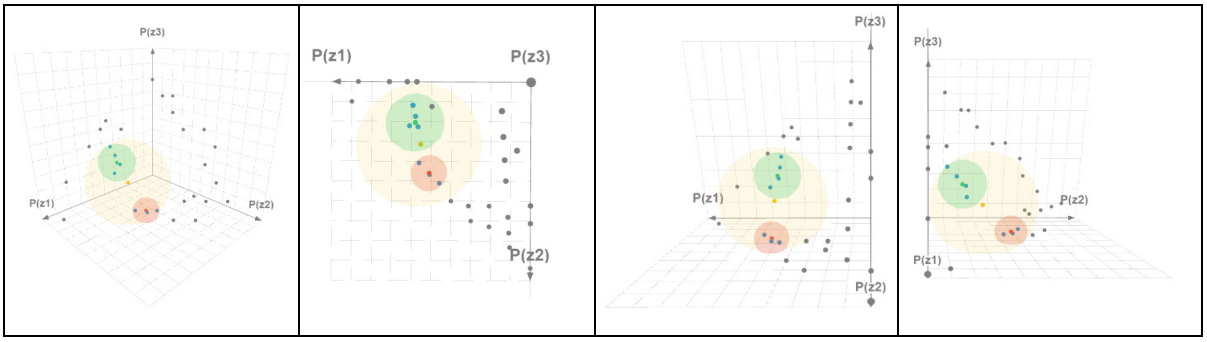


Figure 6.17 Représentation des documents pertinents et non pertinents

La seconde étape consiste à calculer le Δ, soit la différence entre les deux vecteurs préalablement calculés (Tableau 6.4). Il est également important de mentionner que cette soustraction impacte le domaine des valeurs possibles de chacune des composantes. C'est-à-dire que nous passons d'un domaine borné de [0:1] à [-1:1].

Tableau 6.4 Différence entre les vecteurs pertinents et non pertinents (Δ)

Différence					
	$P(z_1)$	**$P(z_2)$**	**$P(z_3)$**	**Somme**	**Domaine des valeurs possibles**
Vecteur pertinent	0,51	0,16	0,33	1,00	[0:1]
Vecteur non pertinent	0,50	0,45	0,05	1,00	[0:1]
Δ	0,01	-0,29	0,28	---	[-1:1]

Ensuite, on additionne le vecteur Δ au vecteur pertinent.

Tableau 6.5 Coordonnées du nouveau centre d'intérêt

Nouveau centre d'intérêt					
	$P(z_1)$	**$P(z_2)$**	**$P(z_3)$**	**Somme**	**Domaine des valeurs possibles**
Vecteur pertinent	0,51	0,16	0,33	1,00	[0:1]
Δ	0,01	-0,29	0,28	0,00	[-1:1]
Nouveau centre d'intérêt	0,53	**-0,13**	0,60	1,00	[-1:1]

Nous devons ensuite ramener le domaine des valeurs possibles entre 0 et 1 du nouveau centre d'intérêt calculé. Dans l'exemple ci-dessus, nous pouvons remarquer que la valeur de $P(z_1)$ du nouveau centre d'intérêt est inférieure à 0. L'idée est de répartir cette valeur négative sur les autres coordonnées positives ($P(z_1)$ et $P(z_3)$). Pour ce faire, nous calculons un vecteur d'ajustement comme suit :

1. Toutes les composantes positives ($P(z_1)$ et $P(z_3)$) se voient ajouter le résultat de

$$\frac{P(z_2)}{\text{Nombre de sujet} - 1} \tag{6.7}$$

2. L'ajustement de $P(z_2)$ se voit attribuer la valeur absolue de $P(z_2)$.

En additionnant les composantes du nouveau centre d'intérêt aux composantes du vecteur d'ajustement, nous obtenons le nouveau centre d'intérêt ajusté.

Tableau 6.6 Calcul du vecteur d'ajustement

	$P(z_1)$	**$P(z_2)$**	**$P(z_3)$**	**Somme**	**Domaine des valeurs possibles**
Nouveau centre d'intérêt	0,53	**-0,13**	0,60	1,00	[-1:1]
Vecteur d'ajustement	-0,06 (-0,13/2)	**0,13**	-0,06 (-0,13/2)	0,00	[-1:1]
Nouveau centre d'intérêt ajusté	0,46	**0**	0,54	1,00	[0:1]

Si l'ensemble des composantes du nouveau centre d'intérêt ajusté ne respecte pas le domaine des valeurs possibles, nous réappliquons le calcul du vecteur d'ajustement.

Les images de la figure 6.18 permettent de voir le déplacement du centre d'intérêt initial (de couleur jaune) vers le nouveau centre d'intérêt qui est à l'extérieur (de couleur orange) et le déplacement de ce dernier vers le nouveau centre d'intérêt ajusté (de couleur mauve).

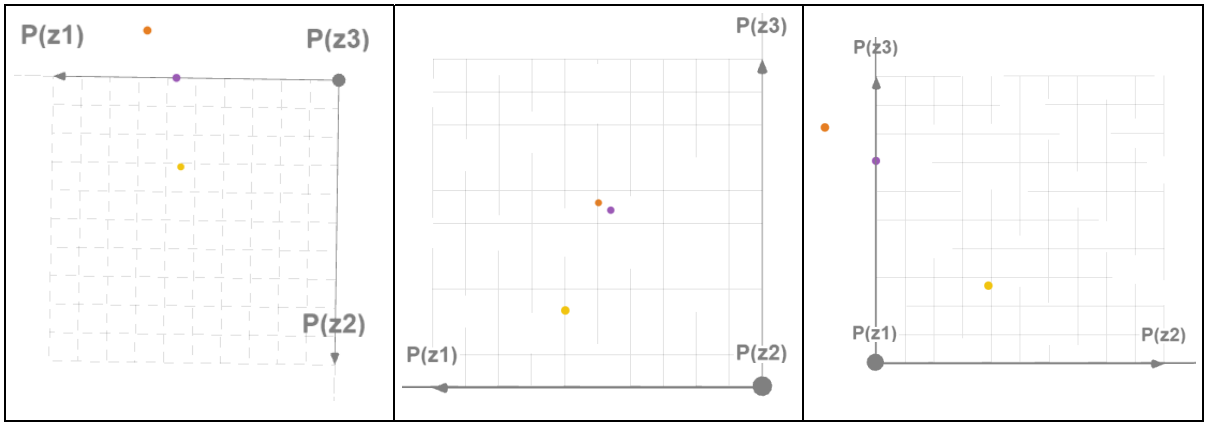


Figure 6.18 Déplacement du centre d'intérêt

Ensuite, une seconde recommandation est émise (les vidéos dans la sphère mauve illustré à la figure 6.19).

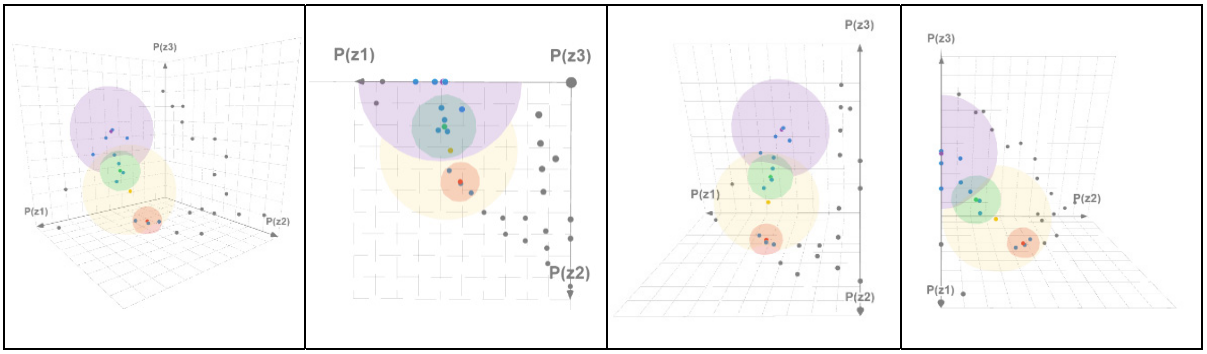


Figure 6.19 Nouvelles recommandations

Ceci illustre le mécanisme d'estimation du nouveau centre d'intérêt de l'utilisateur afin d'émettre une nouvelle liste de recommandations. Le calcul subséquent des centres d'intérêts et des recommandations futures applique ce même mécanisme.

6.3.2 Création de la carte sémantique

Le développement de la carte sémantique exploite la distribution sujet-termes du modèle LDA sélectionné. Puisque cette distribution est sous la forme de probabilités d'appartenance d'un terme à un sujet, il est alors possible de décrire chaque sujet z_k en sélectionnant, par exemple, les 10 termes ayant les probabilités les plus élevées. C'est à partir de ces listes de termes que nous avons bâti une représentation sous la forme d'un graphe non orienté. L'avantage d'exploiter ce type de représentation est qu'elle délaisse les représentations cartésiennes qui insinuent une notion de distance entre les éléments disposés en fonction des différents axes. De plus, cette représentation nous permettra d'illustrer les termes décrivant chaque sujet ainsi que le partage des termes communs entre les différents sujets.

Le développement de ce type de visualisation a été motivé par l'ajout d'un support permettant aux utilisateurs de visualiser les différentes thématiques qui caractérisent les différentes séquences vidéo contenues dans la base de données. De plus, par cette représentation, il est possible de positionner le centre d'intérêt de l'utilisateur dans cet ensemble de thématiques. Puisque le centre d'intérêt est recalculé après le visionnement et la classification de la liste de recommandations, ce nouveau centre d'intérêt est ajouté à la carte. L'enchaînement des centres d'intérêt permet alors de visualiser l'évolution des thématiques pour lesquels l'utilisateur a été intéressé.

Les sous-sections suivantes explicitent l'élaboration de la carte et l'ajout des centres d'intérêt.

6.3.2.1 Élaboration de la carte sémantique

Prenons par exemple les dix termes décrivant les sujets 9, 28 et 34 de la matrice sujet-termes du modèle LDA sélectionné que nous représentons au tableau suivant où les termes dans les cases bleutées sont des termes partagés entre différents sujets.

Tableau 6.7 Matrice sujet-termes

Terme / z_k	1	2	3	4	5	6	7	8	9	10
9	doctor	brown	bear	bed	visitor	stay	goodbye	cookie	pretend	brave
28	tooth	fairy	asleep	fall	noise	awake	stay	uncle	leave	pillow
34	Secret	Box	Word	Silly	custard	doughnut	tommy	cookie	daddy	guess

Afin de transformer ces informations sous la forme d'un graphe, nous avons représenté les identifiants de sujet z_k par des nœuds rouges et les termes par des nœuds verts. Nous avons

créé les associations entre les termes (boules vertes) et leurs sujets d'appartenance (boules rouges) par des traits roses. La figure 6.20 illustre la représentation de cet exemple.

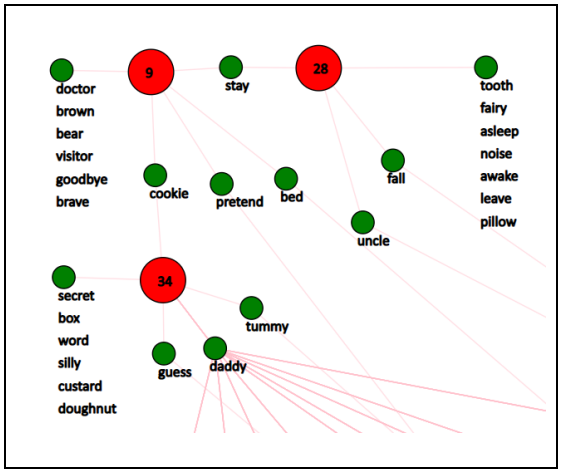


Figure 6.20 Carte sémantique

Une fois le graphe construit, il est intégré au système de recommandations à l'aide de la librairie D3.js.

6.3.2.2 Ajout des centres d'intérêts

L'ajout des centres d'intérêt sur la carte (boules jaunes) est le reflet du calcul de la mise à jour du centre d'intérêt de l'utilisateur qui a été présenté à la section 6.3.1. Comme mentionné précédemment, l'estimation d'un centre d'intérêt est calculée à partir de la distribution document-sujets de l'historique des vidéos regardées par l'utilisateur, et donc, par cette représentation, nous pouvons maintenant décrire le centre d'intérêt calculé par des termes.

La figure 6.21 illustre le centre d'intérêt initial d'un utilisateur qui est représenté par la boule jaune. Dans le contexte d'un système de recommandations, le centre d'intérêt initial est un cas

particulier où il n'y a pas d'historique de recommandations. Afin de générer les premières recommandations, cet utilisateur aurait spécifié des vidéos abordant le sujet 6.

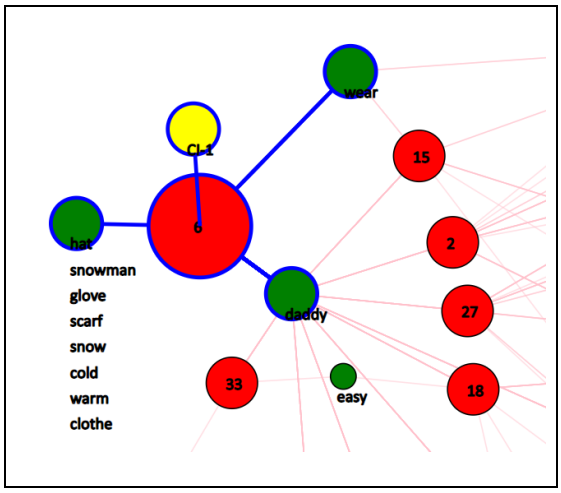


Figure 6.21 Visualisation du centre d'intérêt initial sur la carte

À la suite de l'écoute et de la classification des vidéos recommandées, un nouveau centre d'intérêt est recalculé. Comparativement au centre d'intérêt initial, ce nouveau centre d'intérêt atteindra plusieurs sujets (boules rouges) tels qu'illustrés à la figure 6.22. Ce centre d'intérêt atteint maintenant les boules 6, 4 et 27.

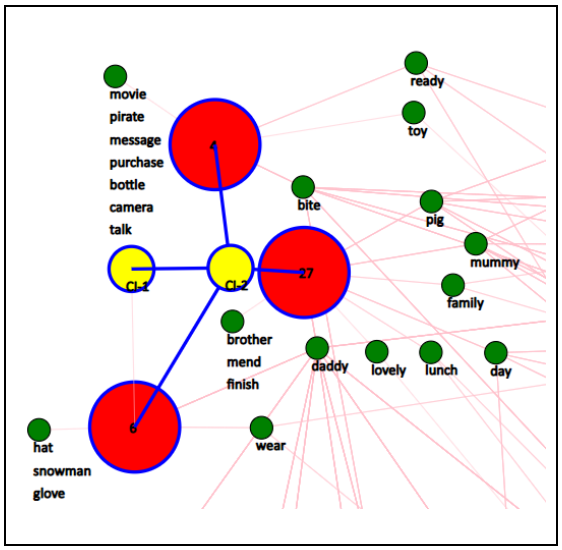


Figure 6.22 Visualisation du second centre d'intérêt sur la carte

Puisque le centre d'intérêt est une distribution par rapport aux sujets des vidéos de l'historique, certains sujets auront des probabilités élevées et d'autres faibles. Afin de représenter ce nouveau centre d'intérêt et les liens aux sujets dominants qui le décrivent, nous avons fait le choix de sélectionner les trois probabilités les plus élevées. De plus, nous ajoutons un lien entre ce nouveau centre d'intérêt et son prédécesseur.

6.3.3 Détails pour le développement des tests d'intrusion

Les tests d'intrusion ont été développés sous l'approche proposée par Chang *et al.* (2009) et préalablement discutés dans la section 1.3.1 de la revue de littérature. Ces auteurs ont proposé deux tests différents, le test du mot intrus (*word intrusion)* et le test du sujet intrus *(topic intrusion)* que nous avons adaptés à notre contexte. Chang *et al.* (2009, p. 3) décrivent le test du mot intrus (*word intrusion)* comme suit :

> Présenter à l'utilisateur six mots dans un ordre aléatoire. L'utilisateur a alors pour tâche de trouver le mot intrus. Lorsque l'ensemble des mots sans l'intrus ont un sens ensemble, alors l'utilisateur devrait facilement identifier l'intrus. Par exemple, la plupart des gens identifient facilement le mot « pomme » comme le

> mot intrus dans l'ensemble suivant : « chien, chat, cheval, pomme, cochon, vache » parce que les termes « chien, chat, cheval, cochon, vache » ont un sens ensemble, ce sont tous des animaux. Pour l'ensemble « voiture, professeur, ornithorynque, agile, bleu, Zaïre » qui manque de cohérence, l'identification de l'intrus sera plus difficile. Les gens choisiront généralement un intrus au hasard, ce qui implique que ce regroupement de mots manque de cohérence.

Dans notre contexte, nous avons choisi d'évaluer les regroupements de mots de chaque sujet (38) issus de la modélisation sélectionnée en appliquant le test du mot intrus (*word intrusion)* tel que proposé par Chang *et al.* (2009). Pour chaque sujet, nous avons sélectionné les trois mots les plus probables auxquels nous ajoutons un intrus afin d'évaluer la cohérence de ce regroupement de mots. Le terme intrus est sélectionné au hasard parmi les termes ayant une faible probabilité d'appartenir au sujet actuel, mais qui a une probabilité élevée d'appartenir à un autre sujet. De cette façon, cela permet de réduire la possibilité que l'intrus provienne du même regroupement de mots cohérents qui décrit le sujet en question et permet de s'assurer que l'intrus n'est pas rejeté uniquement en raison qu'il n'appartienne à aucun regroupement de mots décrivant les autres sujets. Finalement, l'ordonnancement de ces 4 mots est mélangé avant d'être présenté à l'utilisateur.

Chang *et al.* (2009, p. 4) décrivent le test du sujet intrus (*topic intrusion)* comme suit :

> Présenter à l'utilisateur un extrait d'un document et quatre sujets où chacun de ces sujets est représenté par ses huit mots les plus probables. Trois de ces sujets sont les sujets à plus forte probabilité assignés à ce document. Le sujet intrus restant est choisi au hasard parmi les autres sujets à faible probabilité d'appartenir au document. L'utilisateur a pour instruction de choisir le sujet qui n'appartient pas au document. Comme dans le test du mot intrus (word intrusion), si l'attribution des sujets aux documents était pertinente et intuitive, nous nous attendrions à ce que les utilisateurs choisissent le sujet que nous avons ajouté de manière aléatoire comme étant le sujet qui n'appartient pas au document.

Dans notre contexte, nous avons choisi de présenter la séquence vidéo associée au document au lieu de présenter la transcription qui a été utilisée pour entrainer le modèle. Nous justifions

ce choix par le fait que nous cherchons à évaluer le potentiel d'utiliser une technique de modélisation par sujet sur les sous-titres de vidéos afin de concevoir un système de recommandations vidéo. Nous pouvons également renchérir en mentionnant que sans la vidéo, il serait difficile pour l'utilisateur d'extrapoler le contexte du document dû à sa longueur (peu de mots) ainsi que par la nature du corpus (dialogues entre personnages pouvant contenir des erreurs de transcription).

De plus, nous avons également modifié la tâche d'identification de l'intrus par une tâche d'identification du sujet qui décrit **le mieux** la séquence vidéo dues aux distributions document-sujets obtenues. Cette adaptation est motivée par le fait que nous avons optimisé la tâche de modélisation par sujet en ayant pour objectif que chaque séquence vidéo appartienne, dans la grande majorité, à un sujet dominant. Si nous avions appliqué la formulation du test d'intrusion tel que mentionné par Chang *et al.* (2009) (qui stipule de décrire un document par les regroupements de mots appartenant aux trois sujets qui ont les probabilités les plus élevées, en plus d'ajouter une nouvelle liste de mot identifiée comme le vrai intrus) l'utilisateur serait probablement indécis quant à la sélection du vrai intrus, car selon lui, il y aurait deux, voire trois intrus parmi les quatre choix qui lui sont présentés.

Afin d'appuyer cette adaptation, nous avons dénombré pour chaque document les trois probabilités les plus élevées décrivant l'appartenance aux sujets. Nous présentons aux figures suivantes, sous la forme d'histogramme, le décompte des valeurs de la probabilité la plus élevée qu'un document appartienne à un sujet, le décompte des valeurs de la seconde probabilité la plus élevée qu'un document appartienne à un sujet et le décompte des valeurs de la troisième probabilité la plus élevée qu'un document appartienne à un sujet.

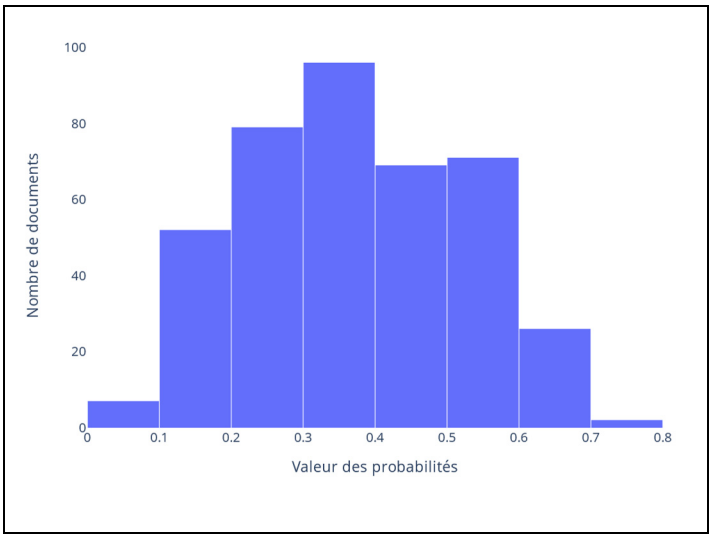


Figure 6.23 Répartition de la première probabilité la plus élevée

Sachant que nous avons 38 sujets, si les probabilités d'un document d'appartenir aux différents sujets étaient équiprobables, ces probabilités seraient de 1/38=2,63%. Par cette première représentation (figure 6.23), nous pouvons remarquer que la majeure partie des valeurs de la probabilité la plus élevée qu'un document appartienne à un sujet est grandement plus élevée que 2,63%.

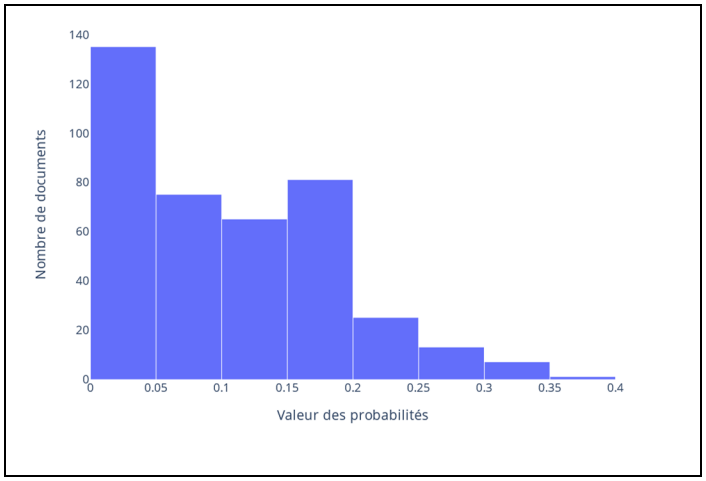


Figure 6.24 Répartition de la seconde probabilité la plus élevée

Par cette deuxième représentation (figure 6.24), nous pouvons remarquer que la seconde probabilité a grandement diminué par rapport à la représentation précédente.

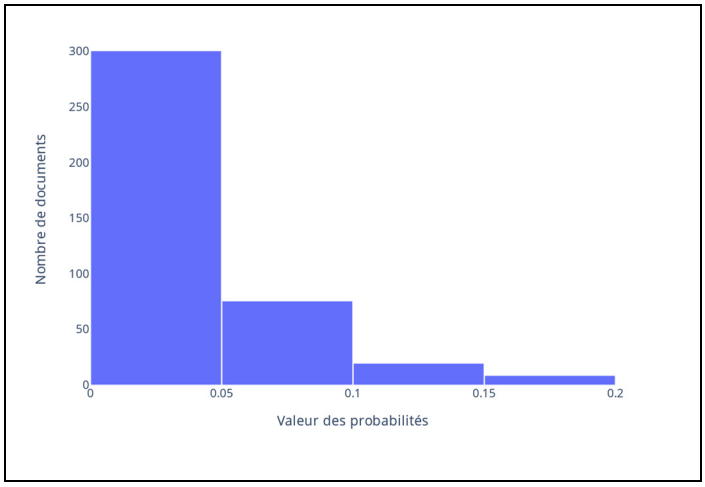


Figure 6.25 Répartition de la troisième probabilité la plus élevée

Par cette troisième représentation (figure 6.25), nous pouvons remarquer que la troisième probabilité commence à être très faible pour la majorité des documents (300) sur un total de 402 documents.

Par ces différents graphiques, nous pouvons observer qu'un document appartient majoritairement à un seul sujet et que la deuxième et la troisième probabilités les plus élevées qu'un document appartienne à un sujet diminuent drastiquement. Cependant, ces représentations ne reflètent pas la distribution des probabilités associée à un document. Elles dénombrent les trois probabilités les plus élevées.

Afin de quantifier le nombre de probabilités uniques qui décrit un document par rapport aux différents sujets, nous avons appliqué une seconde analyse. Pour ce faire, nous avons dénombré pour chaque document, le nombre de probabilités uniques qui décrit sa distribution sur les différents sujets. Prenons l'exemple des distributions document-sujets présentées au tableau

ci-dessous. En comparant les probabilités associées à chacun des sujets pour un document, nous pouvons dénombrer les différentes valeurs de probabilité. Notez que pour cet exemple, nous présentons les probabilités avec une précision à deux décimales (tableau 6.8), mais qu'une précision à 16 décimales a été utilisée pour produire la Figure 6.26.

Tableau 6.8 Exemple du nombre de probabilités uniques

	Sujet-1	**Sujet-2**	**Sujet-3**	**Sujet-4**	**Sujet-5**	**Nombre de probabilités uniques**
Document-1	0,20	0,20	0,20	0,20	0,20	1
Document-2	0,60	0,10	0,10	0,10	0,10	2
Document-3	0,30	0,25	0,15	0,15	0,15	3
Document-4	0,25	0,18	0,17	0,15	0,25	4
Document-5	0,30	0,10	0,15	0,20	0,25	5

Ce que le nombre de probabilités uniques nous permet de constater est que si cette valeur est d'un, cela signifie que le document n'appartient pas à un sujet en particulier et que du point de vue du test d'intrusion tel que proposé par Chang *et al.* (2009), l'identification du regroupement de mots intrus sera une sélection faite au hasard. Dans le cas où le nombre de probabilités uniques est de deux, l'identification du regroupement de mots intrus serait également une sélection faite au hasard. En revanche, si nous demandions à l'utilisateur d'identifier le sujet qui décrit le mieux ce document, l'utilisateur se verrait moins hésitant quant à son choix (moins d'ambiguïté). De plus, la valeur du nombre de probabilités uniques permet de déduire un niveau de difficulté dans la tâche d'identification du sujet décrivant le mieux un document. Par exemple, on pourrait s'attendre qu'il soit plus difficile d'identifier le sujet qui décrit le mieux un document si le nombre de probabilités différentes est de cinq comparativement à s'il est de deux.

Après avoir appliqué cette technique du décompte du nombre de probabilités uniques sur l'ensemble des différentes distributions document-sujets, nous obtenons les résultats présentés à l'image ci-dessous.

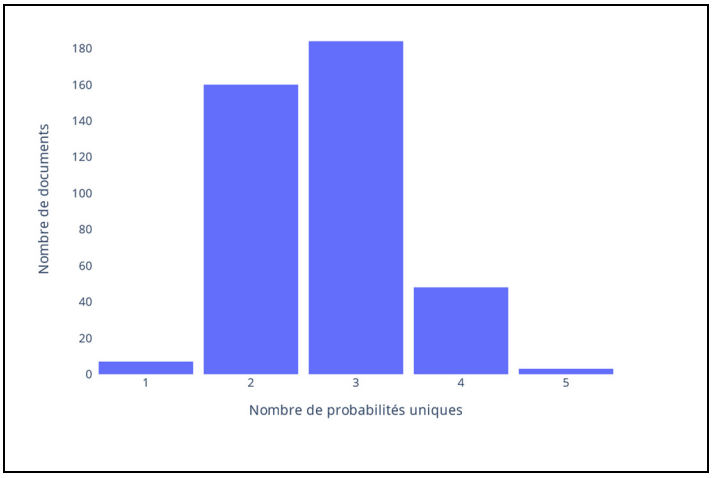


Figure 6.26 Dénombrement des différentes probabilités associées à document

Cette image nous permet de constater que nous avons cinq classes différentes où chacune correspond au nombre de probabilités uniques associées à un document.

Cette constatation nous a amené à construire trois sous-ensembles de niveau de difficulté (facile, moyen, difficile) pour la tâche d'identification. Nous considérons que si un document est constitué de deux probabilités uniques pour décrire les 38 sujets, cela insinue qu'il y a une probabilité élevée pour un sujet et que l'ensemble des autres sujets ont la même probabilité, donc il sera facile pour un utilisateur d'identifier le sujet dominant. Avec cette même logique, nous catégorisons les documents appartenant à la classe 3 comme étant moyennement difficiles et regroupons sous le groupe difficile les classes 1, 4 et 5.

Finalement, en fonction de ces trois sous-ensembles de niveau de difficulté, nous avons sélectionné de manière aléatoire cinq documents de niveau facile, dix documents de niveau

moyen et cinq documents de niveau difficile afin de construire 20 questions pour la tâche d'identification.

6.4 Conclusion

Ce chapitre a permis de décrire l'élaboration de notre application Web qui réunit les différentes composantes présentées à la Figure 2.2 afin de concevoir le système de recommandions. Nous avons présenté les différentes technologies utilisées qui ont permis d'implémenter les différentes parties (client et serveur) de cette application. La description de la partie client a permis de décrire les différents exercices auxquels les participants seront invités à compléter une fois authentifiés. La description de la partie serveur a permis de détailler les principaux algorithmes qui permettent l'estimation du centre d'intérêt suite à l'écoute et à la classification des vidéos, l'élaboration de la carte sémantique et le développement des différents tests d'intrusion.

Finalement, le développement de cette application nous permettra de mettre à l'épreuve les différentes hypothèses qui ont guidé le développement de ce système de recommandations. Plus particulièrement lors de l'évaluation des résultats, nous allons évaluer le pouvoir d'estimation du centre d'intérêt de l'utilisateur et l'interprétabilité de la carte sémantique. De plus, nous allons évaluer à l'aide des participants le potentiel de la démarche liée à la modélisation des sous-titres et la sélection du modèle optimal identifié quantitativement de manière qualitative. Cette évaluation sera possible en évaluant si les ensembles de mots présentent une cohérence sémantique identifiable par le test du mot intrus (*word intrusion)* ainsi que la qualité des affectations des sujets aux documents par le test du sujet intrus (*topic intrusion*).

CHAPITRE 7

RÉSULTATS DE LA PHASE D'ÉVALUATION

Ce chapitre a pour objectif de présenter les résultats obtenus lors cette première phase d'évaluation et donc, de compléter le sous-objectif 5. L'application développée nous a permis d'évaluer deux aspects de ce projet de recherche. Le premier aspect concerne l'utilisation de la modélisation des thématiques inférées par LDA sur un corpus constitué des sous-titres de vidéos pour enfants dans un contexte de recommandations vidéo ainsi que l'utilisation de la carte sémantique en tant que support visuel. Le second aspect concerne la cohérence des sujets latents identifiés par LDA à la suite de l'application de notre méthode d'optimisation de ses paramètres. Pour ce faire, nous avons procédé à une étude pilote avec un groupe de participants neurotypiques recrutés au sein de notre communauté universitaire.

Nous débuterons la présentation des résultats par une analyse du taux de participation et de rétention aux différents exercices. Suivra l'analyse des résultats obtenus aux questions qualitatives lors de l'utilisation du système de recommandations et de la carte sémantique. Cette analyse permettra de recueillir l'appréciation quant à la qualité des recommandations et quant au potentiel de la carte sémantique à exposer les différents thèmes abordés dans les sous-titres des vidéos analysées ainsi qu'à exposer les liens entre les différentes séquences vidéo identifiées comme pertinentes. Finalement, nous analyserons les résultats obtenus aux tests d'intrusion.

7.1 Évolution du taux de participation

Notre application a été mise en ligne le 25 novembre 2022 pour une période d'un mois. Cette application a interpellé la curiosité de plusieurs membres de la communauté étudiante, mais ce n'est qu'une fine partie qui a complété l'ensemble des tests.

Nous présentons à la figure suivante l'évolution du taux de rétention des participants au fil des différents tests où les bandes bleues représentent le nombre de participants ayant complété les différents tests et les bandes rouges représentent le nombre de participants qui ont abandonné.

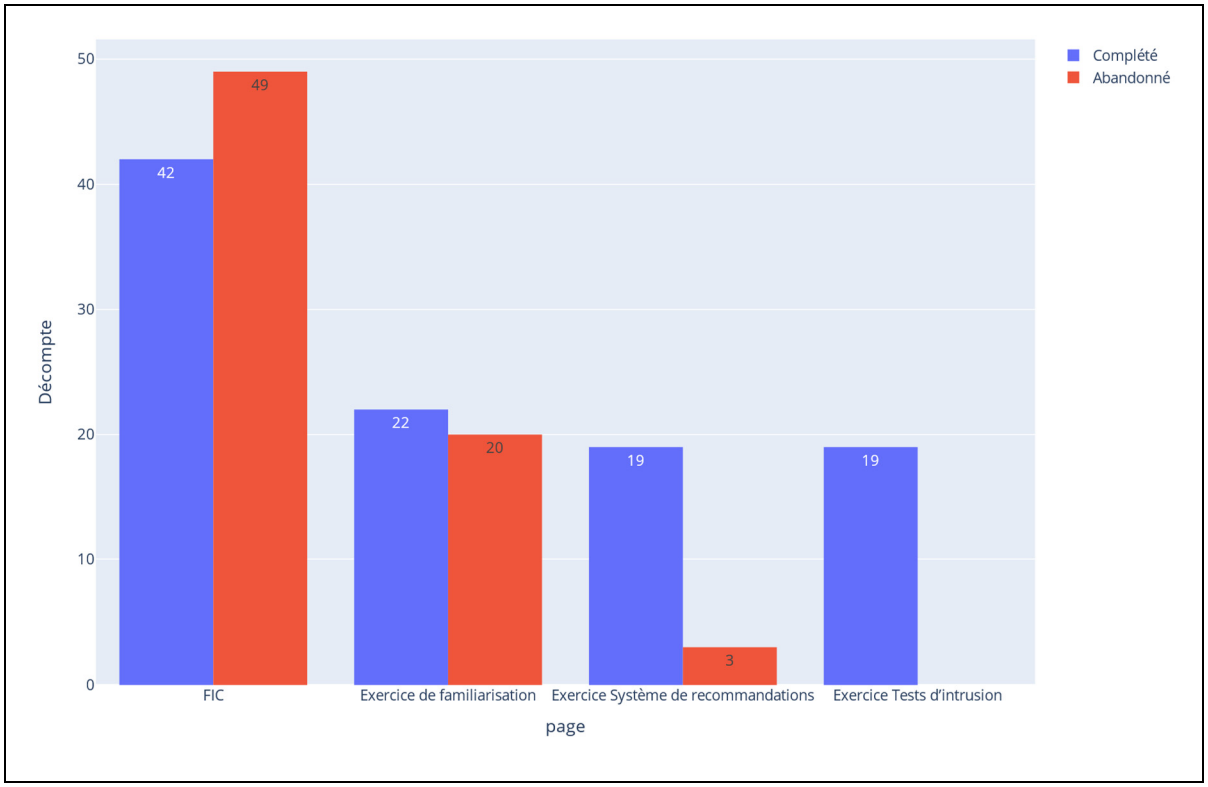


Figure 7.1 Dénombrement des participants aux différents tests

Comme mentionné au chapitre 6 et plus particulièrement à la Figure 6.3, pour qu'un participant puisse effectuer les tests, il doit accepter le Formulaire d'Information et de Consentement (FIC). Nous pouvons alors déduire que 91 participants se sont connectés à l'application et que 42 d'entre eux ont accepté de poursuivre (42/91=46,15%). Le test suivant consistait à l'exercice de familiarisation. Pour les 42 participants ayant accepté le FIC, 22 d'entre eux ont complété ce test (22/42=53,38%). A suivi le test concernant l'exercice sur le système de recommandations, lequel a été complété par 19 des 22 participants (19/22=86,36%). Pour le dernier test, soit les exercices sur les tests d'intrusion, tous les participants l'ont complété (19/19=100,00%). Malgré les efforts mis en place afin de permettre aux participants de

sauvegarder l'état de leur avancement au cours de différents exercices ainsi que la période d'un mois pour les compléter, nous pouvons calculer que seulement 45,23% (19/42) des participants ayant accepté le FIC ont complété l'ensemble des tests.

7.2 Évaluation du système de recommandations

L'évaluation d'un système de recommandations est particulièrement difficile en raison de ses composantes interactives, de son caractère automatique et des différentes interventions faites par le participant. Pour cette première phase d'évaluation, nous avons opté pour l'utilisation d'un questionnaire, qui s'affiche une fois l'exercice complété. Nous avons posé six questions où le participant devait répondre à l'aide d'une échelle de *Likert* composée de trois valeurs nominales (non, neutre, oui) et une question ouverte afin de recueillir son avis quant à des améliorations potentielles sur la carte. De plus, pour chacune des questions, nous avons laissé la possibilité au participant de nous faire part de ses commentaires.

Avant de débuter l'analyse des résultats pour les différentes questions, voici une remise en contexte de l'exercice proposé aux participants :

> Afin d'initier la première série de recommandations vidéo, le participant doit sélectionner une des thématiques issues de la modélisation pour laquelle il souhaite regarder des vidéos. Le système émet ensuite une première série de recommandations contenant cinq vidéos qu'il doit regarder. Après l'écoute de chacune des vidéos, le participant doit juger si elle est pertinente ou non à la thématique de départ qu'il a préalablement sélectionnée. Après l'écoute et la classification de cette première série de vidéos, le système recalcule une nouvelle liste de recommandations vidéo. Chaque nouvelle liste de recommandations contient un minimum de trois et un maximum de cinq vidéos. Finalement, nous avons limité le nombre de vidéos à regarder à 20.

Une fois la limite de vidéos recommandées atteinte, le participant est amené à remplir le formulaire d'évaluation. Ce formulaire contient sept questions. La première question concerne la qualité des recommandations et les questions suivantes concernent la pertinence de la carte en tant que support visuel lors de l'utilisation d'un système de recommandations.

7.2.1 Présentation des résultats pour chaque question

Question 1 : Est-ce que les recommandations concordent avec votre thématique de départ ?

L'objectif de cette question est d'évaluer si le calcul proposé (section 6.3.1) pour estimer le centre d'intérêt du participant afin de calculer une nouvelle liste de recommandations a permis d'émettre des recommandations qui satisfaisaient l'intérêt du participant.

Le graphique suivant dénombre pour chacune des valeurs nominales possibles la réponse des participants. Comme nous pouvons le constater, aucune personne n'a répondu *Non*, trois personnes ont répondu *Neutre*, 16 personnes ont répondu *Oui*.

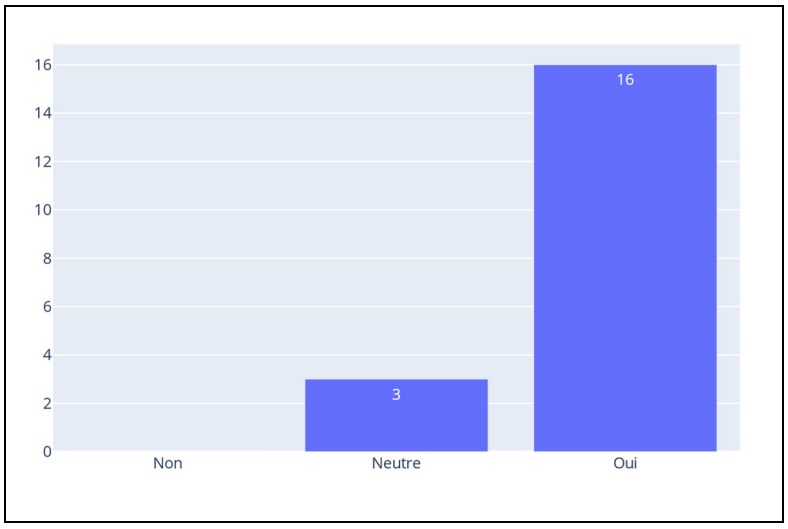


Figure 7.2 Répartition des réponses des participants pour la question 1

Par cette représentation, nous pouvons conclure que la majorité des participants s'accordent à dire que les recommandations correspondaient bien à leur thématique de départ. Cependant, certains participants, dont ceux qui ont attribué la mention *Neutre,* ont mentionné que certaines vidéos, parmi le maximum de 20, n'avaient aucun lien avec la thématique de départ.

Plusieurs raisons peuvent expliquer l'impression d'incohérence lors de l'émission des recommandations. Nous pouvons mentionner le nombre de sujets total élevé (38) décrivant peu de vidéos (402) ainsi qu'un nombre minimum de trois vidéos à émettre par recommandation sur une base de données limitée (une saison).

Les prochaines questions concernent la carte sémantique. La carte a été conçue afin d'offrir un support visuel qui a pour but d'exposer les différents thèmes abordés dans toutes les séquences vidéo contenues dans notre base de données. Nous croyons que cela permet aux participants d'effectuer une recherche active afin de mieux comprendre les choix imposés par l'algorithme (en donnant une vue globale) et d'exposer au participant les liens entre les différentes séquences vidéo identifiées comme pertinentes.

Question 2 : Parmi les ensembles de mots que vous venez de mettre en évidence (boules vertes), est-ce que ces mots, sans nécessairement tous les sélectionner, vous permettraient de décrire le contenu des vidéos que vous venez de regarder ?

L'objectif de cette question est d'évaluer si les regroupements de mots identifiés par LDA permettent de décrire le contenue des vidéos regardées.

Le graphique suivant dénombre pour chacune des valeurs nominales possibles la réponse des participants. Comme nous pouvons le constater, une personne a répondu *Non*, trois personnes ont répondu *Neutre*, 15 personnes ont répondu *Oui*.

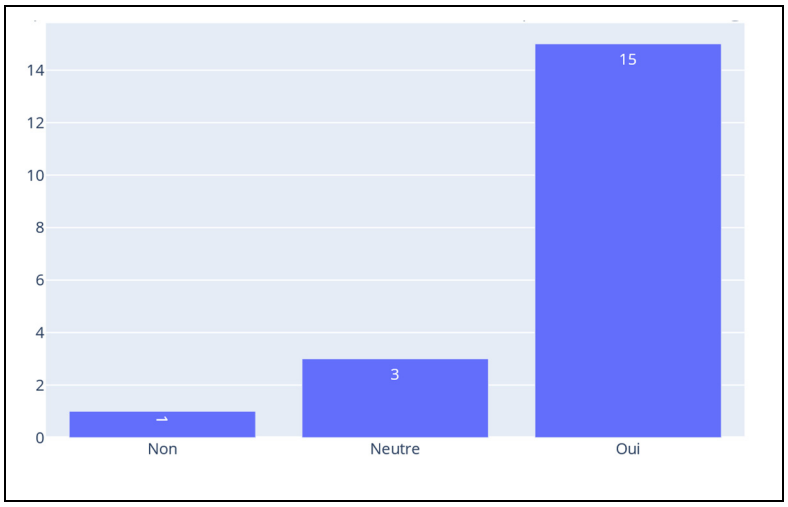


Figure 7.3 Répartition des réponses des participants pour la question 2

Par cette représentation, nous pouvons conclure que la majorité des participants s'accordent à dire qu'il y a un potentiel descriptif à l'utilisation de ces différents regroupements de mots afin de décrire le contenu des vidéos qu'ils ont regardés. Cependant, il est mentionné par plusieurs participants que ce n'est pas tous les termes qui permettent de décrire le contenu des vidéos et qu'il serait bien d'épurer l'ensemble des termes provenant des différents regroupements de mots afin de réduire les ambigüités.

Question 3 : Est-ce que les ensembles de mots (boules vertes) liés à votre dernier centre d'intérêt (boule jaune ayant l'indice numérique le plus élevé) vous permettraient de décrire le contenu des vidéos que vous avez qualifié de pertinent ?

Au fil des recommandations et de l'évaluation de la pertinence des vidéos accordée par le participant, le calcul d'estimation du centre d'intérêt tend à identifier les différents sujets de prédilection. C'est donc dire que l'évaluation de la pertinence accordée aux vidéos élimine ou ajoute des sujets ce qui permet de raffiner le centre d'intérêt.

L'objectif de cette question est d'évaluer si le calcul d'estimation du centre d'intérêt a le comportement attendu.

Le graphique suivant dénombre pour chacune des valeurs nominales possibles la réponse des participants. Comme nous pouvons le constater, quatre personnes ont répondu *Non*, une personne a répondu *Neutre*, 14 personnes ont répondu *Oui*.

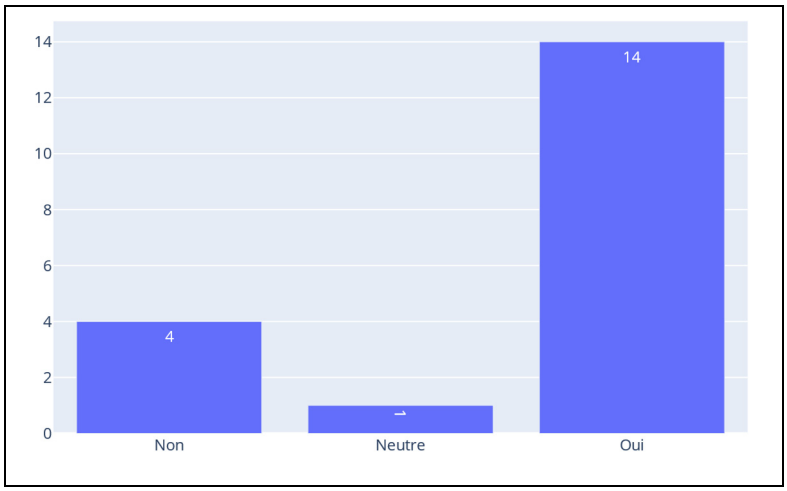


Figure 7.4 Répartition des réponses des participants pour la question 3

Par cette représentation, nous pouvons conclure que la majorité des participants semblent dire que *Oui*. Cependant, un participant a mentionné que le centre d'intérêt final est toujours lié à la thématique de départ. Un autre participant mentionne que le centre d'intérêt final concorde bien avec les vidéos classifiées comme pertinentes, mais que seulement deux regroupements de mots sur trois semblent pertinentes pour décrire les vidéos. Ce même participant a cependant mentionné que bien que le troisième regroupement de mots ne décrive pas les vidéos classifiées de pertinentes, il a un lien sémantique avec ces dernières. Dans ce cas, le troisième sujet regroupait des mots reliés à la nourriture (par exemple « *custard* » et « *doughnut* »), mais c'était d'autres types d'aliments qui étaient dans la vidéo (par exemple « *pancake* »).

Basé sur ces commentaires, une piste d'amélioration serait de donner un indicatif de la force des liens entre le centre d'intérêt et les différents sujets auxquels il est lié.

Question 4 : Au-delà des choix de recommandation faits par l'algorithme, trouvez-vous pertinent de voir l'estimation de votre centre d'intérêt fait par le système de recommandations au travers cette carte ?

L'objectif de cette question est d'évaluer si l'utilisateur trouve pertinent de voir son centre d'intérêt estimé par le système de recommandations au sein d'une représentation des thématiques utilisées par le système pour calculer les recommandations.

Le graphique suivant dénombre pour chacune des valeurs nominales possibles la réponse des participants. Comme nous pouvons le constater, quatre personnes ont répondu *Non*, cinq personnes ont répondu *Neutre*, dix personnes ont répondu *Oui*.

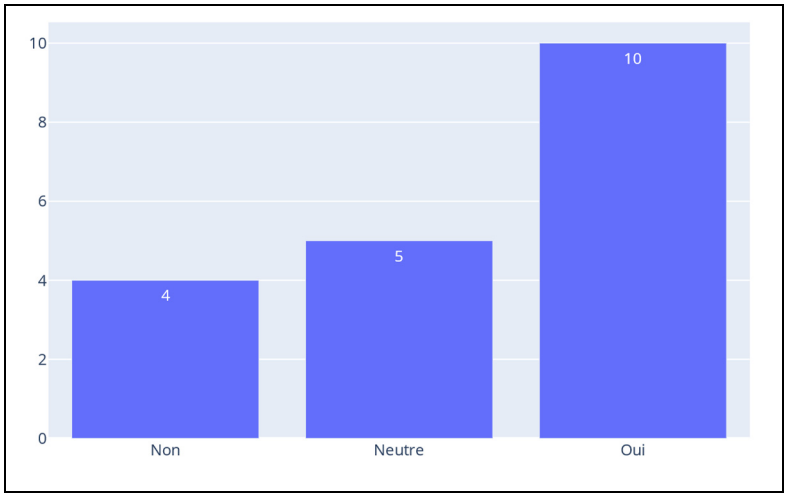


Figure 7.5 Répartition des réponses des participants pour la question 4

Par cette représentation nous pouvons conclure que la réponse des participants semble mitigée.

Des participants ont mentionné que cette représentation est intéressante et qu'elle permet de visualiser assez efficacement leurs centres d'intérêt calculés. De plus, l'évolution des centres d'intérêt permet de mettre en évidence les thématiques abordées dans les vidéos. Cependant

les participants qui adhéraient le moins à l'utilisation de cette carte ont mentionné que la difficulté réside dans la présentation surchargée des éléments de la carte ce qui rend sa lecture et la visualisation des liens difficiles même lorsque la fonctionnalité de mise en évidence est appliquée

Question 5 : Si cette interface était intégrée sur Netflix, est-ce que vous l'utiliseriez ?

L'objectif de cette question est de voir l'intérêt des participants quant à l'ajout de ce type de représentation dans un contexte d'utilisation d'un système de recommandations. Usuellement, les systèmes de recommandations vidéo, Netflix par exemple, sont utilisés comme des boîtes noires et n'offrent aucune représentation des intérêts de l'utilisateur par rapport aux caractéristiques utilisées pour calculer les recommandations.

Le graphique suivant dénombre pour chacune des valeurs nominales possibles la réponse des participants. Comme nous pouvons le constater, trois personnes ont répondu *Non*, sept personnes ont répondu *Neutre*, neuf personnes ont répondu *Oui*.

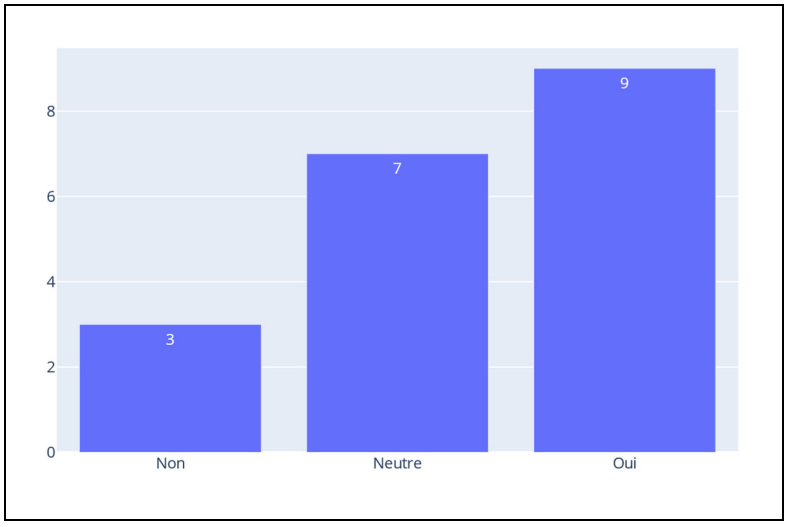


Figure 7.6 Répartition des réponses des participants pour la question 5

Les résultats et commentaires nous laissent croire que la description par thématique et la visualisation sont pertinentes, mais que la difficulté réside dans l'utilisation de la carte. Par exemple des participants ont mentionné que le côté visuel et interactif donne une plus grande autonomie de choix contrairement à l'algorithme imposé par Netflix et que la carte permet une autre façon de parcourir les vidéos autre que les catégories auxquelles nous sommes habitués tels que « recommandés pour vous », « récemment ajoutés » et « les plus populaires ». Bien que ces commentaires soient pertinents, ils semblent être davantage liés à la prochaine question.

Question 6 : Selon vous, serait-il pertinent de visualiser les vidéos en les sélectionnant à partir de la carte ?

L'objectif de cette question est de voir l'intérêt des participants quant à l'ajout de cette fonctionnalité.

Le graphique suivant dénombre pour chacune des valeurs nominales possibles la réponse des participants. Comme nous pouvons le constater, deux personnes ont répondu *Non*, trois personnes ont répondu *Neutre*, 14 personnes ont répondu *Oui*.

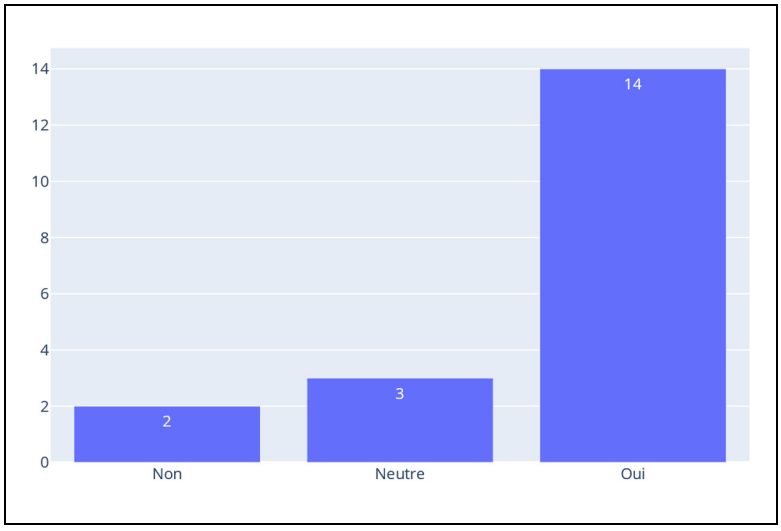


Figure 7.7 Répartition des réponses des participants pour la question 6

Par cette représentation nous pouvons conclure que la majorité des participants semblent dire que *Oui*. En comparant cette question avec les deux précédentes, nous déduisons que la carte sémantique semble beaucoup plus intéressante si elle a une utilité versus si elle ne sert qu'à comprendre la logique derrière les recommandations. Un des participants mentionne que la sélection des vidéos à partir de la carte serait intéressante si cela permettait de débloquer certains centres d'intérêt. Certains participants ont mentionné que la recherche par mots-clés à partir de la barre de recherche leur convient. Un participant évoque le potentiel de cette fonctionnalité à trouver des vidéos abordant plusieurs thématiques d'un seul coup et à explorer d'autres vidéos connexes au centre d'intérêt calculé.

Question 7 : Qu'est-ce que vous amélioriez sur cette carte ?

L'objectif de cette question ouverte est d'obtenir les commentaires des participants afin d'obtenir des pistes d'amélioration liées à cette carte et ainsi de comprendre les difficultés/désagréments que les participants ont rencontrés lors de son utilisation.

Globalement, les participants ont mentionné que la carte est visuellement surchargée, c'est-à-dire qu'elle contient trop de boules et de liens qui s'entrecroisent. Cela rendait l'exploration et la lecture de la carte difficile lors des tâches d'identification d'éléments afin de répondre à certaines des questions. Pour ce faire, les participants auraient aimé avoir des fonctionnalités qui auraient facilité la mise en évidence des éléments tels que :

- Ajouter un bouton qui permettrait d'afficher seulement les termes reliés aux différents centres d'intérêt.
- Ajouter un outil de recherche pour atteindre plus rapidement une thématique (boule rouge) afin d'éviter d'agrandir et de réduire la taille des éléments sur la carte.

- Ajouter des filtres qui permettraient d'isoler l'information pertinente que le participant cherche à regarder.
- Ajouter une fonctionnalité qui permet de cacher des éléments.
- Extraire l'information pertinente et la présenter sous la forme d'un tableau.

Certains participants ont également mentionné l'ajout des nouvelles fonctionnalités suivantes :

- Sélectionner des termes pour créer le centre d'intérêt de l'utilisateur afin que le système produise une liste de recommandations selon ces contraintes.
- Créer un centre d'intérêt à partir de la carte pour initier les recommandations.

7.2.2 Discussion

Pour cette première série de tests, plus de la moitié des participants ont répondu dans l'affirmative à toutes les questions. Cela nous permet de conclure que le système de recommandations semble correctement générer des vidéos en lien avec la thématique de départ sélectionnée (question 1) et que la carte sémantique semble être utile dans les contextes de recherche et de sélection de vidéos ainsi que pour décrire les contenus des vidéos regardées (questions 2 à 7).

7.3 Tests d'intrusion

Les tests d'intrusion permettent une analyse qualitative des distributions produites par LDA par l'adéquation du jugement des personnes externes au projet de recherche. Ces tests permettent de vérifier si les distributions identifiées par LDA sont cohérentes avec le jugement humain.

Pour chacun de ces tests (deux), nous avons évalué la distribution des scores finaux des participants en dissociant ceux qui sont familiers avec la série *Peppa Pig* et ceux qui ne le sont pas. Le score final d'un participant pour un test est calculé en appliquant la somme des réponses attendues divisée par le nombre total de questions.

7.3.1 Le sujet intrus (topic intrusion)

La tâche d'identification du sujet intrus permet d'évaluer si la décomposition d'un document en fonction des sujets est cohérent pour le jugement humain. Pour cette série d'exercices, les participants doivent identifier le regroupement de mots décrivant le mieux la séquence vidéo regardée.

Ce test est constitué de 20 questions catégorisées par trois niveaux de difficulté : facile, moyen et difficile. Pour chacune de ces questions, le participant doit choisir la bonne réponse parmi quatre choix. Nous avons sélectionné cinq questions de niveau facile, dix de niveau moyen et cinq de niveau difficile. Ces 20 questions utilisent toutes des séquences vidéo différentes et ont été sélectionnées de façon aléatoire. Cet échantillon représente environ 5% (20/402) de l'ensemble des vidéos contenues dans la base de données.

La figure suivante montre la distribution des résultats obtenus pour l'ensemble des utilisateurs, où chaque question a une pondération d'un point. En regardant le trait horizontal pointillé, on peut y voir une moyenne globale de 72% avec un écart-type de 6%. Tel qu'attendu, la moyenne des participants connaissant la série (77%) est plus élevée que la moyenne globale ainsi que de celle où les participants sont non familiers avec la série (68%). Les points à gauche des représentations de boîtes à moustache (*box plot*) sont les résultats individuels des participants.

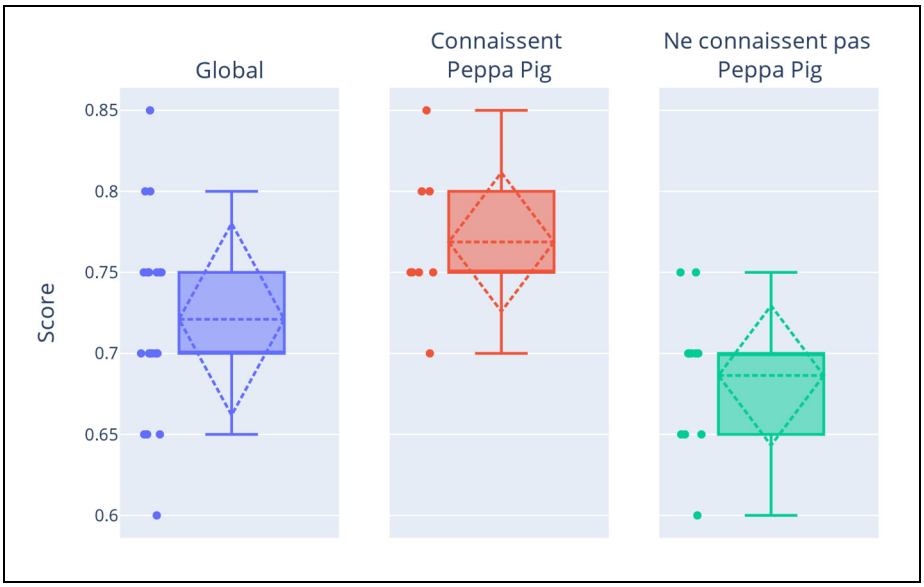


Figure 7.8 Distribution des résultats obtenus par les participants au test du sujet intrus (topic intrusion)

En assumant qu'un utilisateur aurait répondu aléatoirement à chacune de ces questions, cet utilisateur aurait probablement obtenu un score final de 25%. En comparant ce score aléatoire à la moyenne générale obtenue par les utilisateurs (72%), nous pouvons conclure que, pour cet échantillon de vidéos sélectionnées, les regroupements de mots les plus probables de décrire une séquence vidéo identifiée par LDA sont cohérents avec cette dernière.

Les sous-sections suivantes présentent les questions et les résultats par niveau de difficulté. Dans ces sections, nous allons également commenter les questions dont le score global est inférieur à 50% et approfondirons ces questions en analysant les transcriptions et les différentes probabilités associées aux différents choix.

7.3.1.1 Questions du niveau de difficulté facile

Le graphique suivant montre pour chacune des questions la répartition des réponses des utilisateurs. La couleur rouge représente une mauvaise réponse et le vert la réponse attendue.

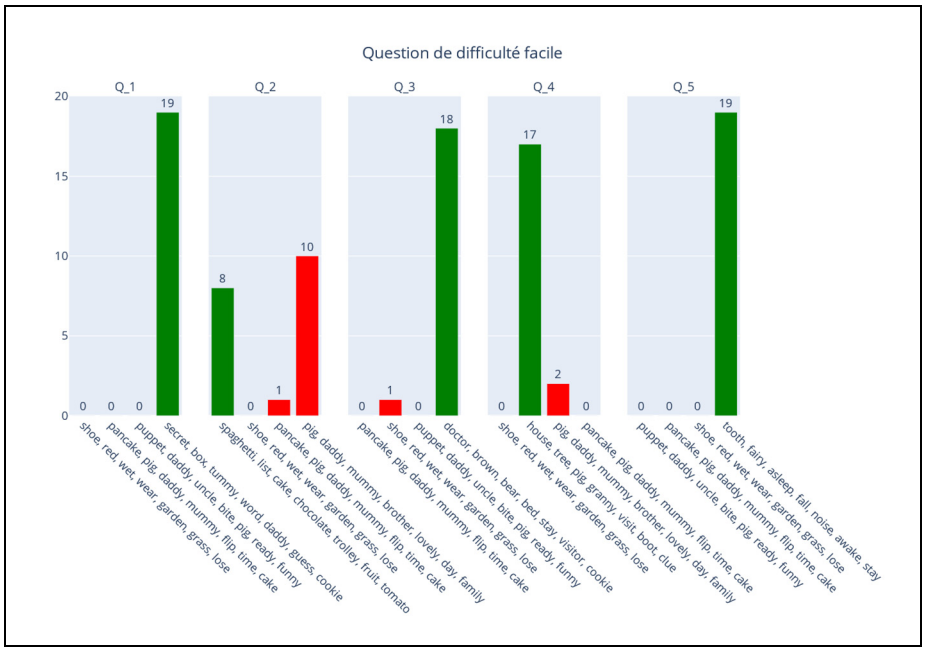


Figure 7.9 Répartition des réponses données par les participants aux questions du niveau de difficulté facile

Le tableau ci-dessous présente les scores aux cinq questions de niveau facile.

Tableau 7.1 Score des participants aux questions du niveau de difficulté facile

	Questions				
	1	2	3	4	5
Global	100,00%	42,11%	94,74%	89,47%	100,00%
Familier avec *Pappa Pig*	100,00%	50,00%	100,00%	87,50%	100,00%
Non familier avec *Pappa Pig*	100,00%	36,36%	90,91%	90,91%	100,00%

Nous remarquons que le score global de la question 2 est inférieur à 50% en raison des utilisateurs qui ne sont pas familiers avec cette télésérie. Ci-dessous, nous fournissons les détails de la vidéo associée à cette question qui seront nécessaires pour l'analyse.

Question 2

Transcription sans prétraitement[21]				
I'm Peppa Pig this is my little brother this is Bobby pig peppa and George are going shopping pepper and George like shopping George loves sitting in the trolley so does pepper daddy can I sit in the trolley - oh you're too big for the trolley pepper oh but you				
Transcription avec prétraitement				
pig little brother mummy pig shopping like shop love sit trolley daddy sit trolley big trolley				
Résumé des résultats				
	Sujet 19	Sujet 1	Sujet 2	Sujet 27
Probabilité	0,277	0,012	0,012	0,277
Décompte	8	0	1	10
Regroupement de mots	spaghetti, list, cake, chocolate, trolley, fruit, tomato	shoe, red, wet, wear, garden, grass, lose	pancake, pig, daddy, mummy, flip, time, cake	pig, daddy, mummy, brother, lovely, day, family

Après la lecture des transcriptions et des regroupements de mots, nous sommes d'accord que ces deux regroupements de mots sont cohérents avec la séquence vidéo. Le sujet 27 concerne les principaux personnages de la télésérie et le sujet 19 consiste en l'activité qu'ils accomplissent. De plus, selon les probabilités attribuées par LDA, ces deux sujets ont la même

[21] Cette séquence vidéo débute à 00:02 et se termine à 00:46 (https://www.youtube.com/watch?v=EL6rvG9qVAc&feature=youtu.be&ab_channel=UCdRyB_BAJDfo0clpilGkFzQ)

valeur. Pour cette question, nous concluons que les participants ayant sélectionné le sujet 27 ont également la bonne réponse.

7.3.1.2 Questions du niveau de difficulté moyen

Les tableaux et graphiques suivants montrent pour chacune des questions la répartition des réponses et les scores aux dix questions de niveau de difficulté moyen des participants. La couleur rouge représente une mauvaise réponse et le vert la réponse attendue.

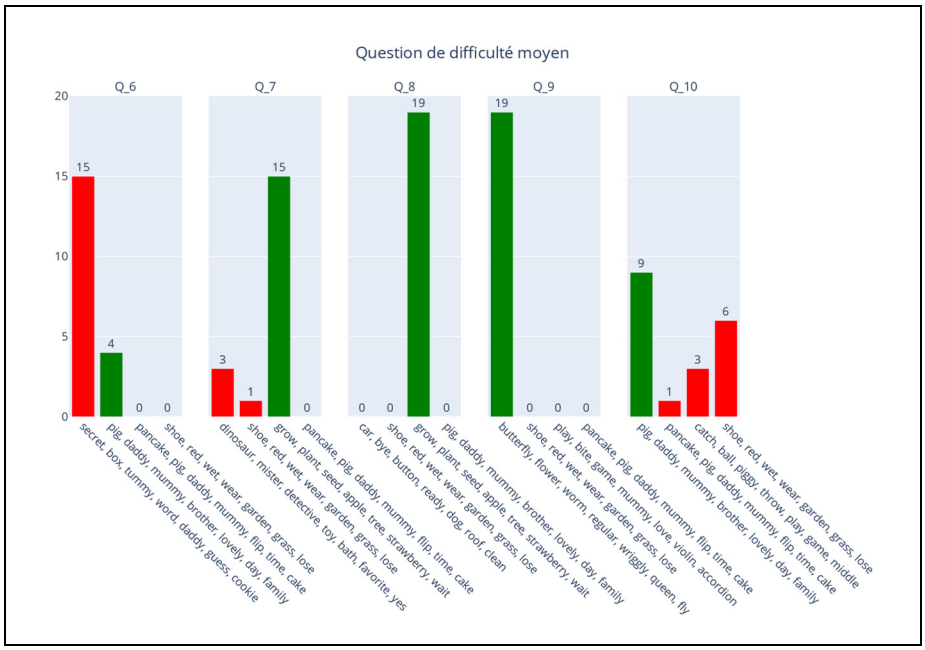


Figure 7.10 Répartition des réponses données par les participants aux questions 6 à 10 du niveau de difficulté moyen

Tableau 7.2 Score des participants aux questions 6 à 10 du niveau de difficulté moyen

	Questions				
	6	7	8	9	10
Global	21,05%	78,95%	100,00%	100,00%	47,37%
Familier avec *Pappa-Pig*	12,50%	100,00%	100,00%	100,00%	50,00%
Non familier avec *Pappa-Pig*	27,27%	63,64%	100,00%	100,00%	45,45%

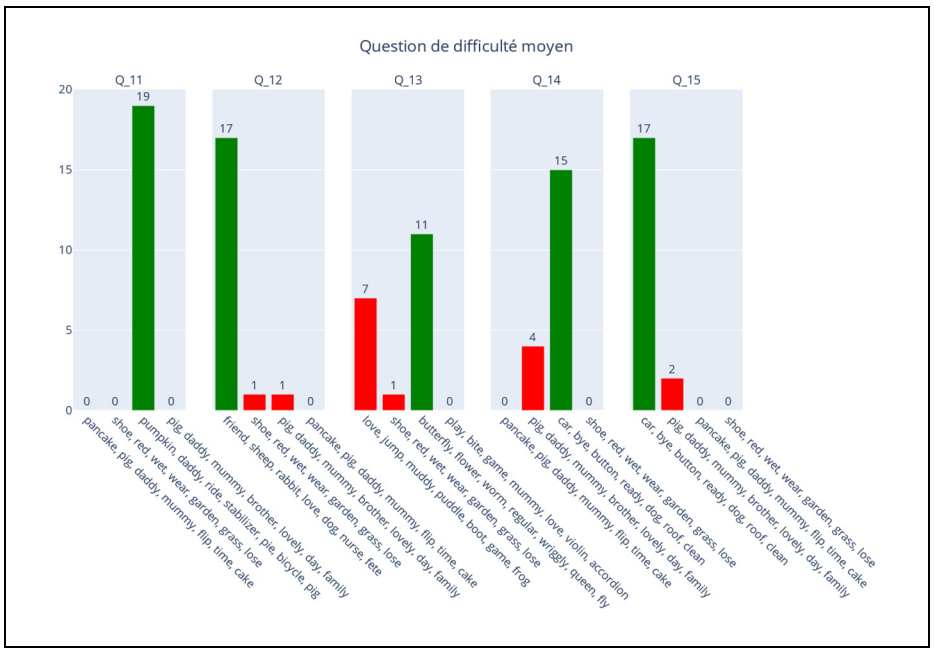


Figure 7.11 Répartition des réponses données par les participants aux questions 11 à 15 du niveau de difficulté moyen

Tableau 7.3 Score des participants aux questions 11 à 15 du niveau de difficulté moyen

	Questions				
	11	12	13	14	15
Global	100,00%	89,47%	57,89%	78,95%	89,47%
Familier avec *Pappa-Pig*	100,00%	100,00%	75,00%	75,00%	100,00%
Non familier avec *Pappa-Pig*	100,00%	81,82%	45,45%	81,82%	81,82%

Globalement, nous pouvons remarquer que la moyenne des scores pour les exercices de ce niveau de difficulté (76%) est inférieure à celle du niveau de difficulté facile (85%).

Notons également que les scores globaux aux questions 6 et 10 sont inférieurs à 50% où la questions 6 est la plus manquée. Pour la question 6, nous remarquons que les participants ayant mentionnés être familiers avec la télésérie ont des scores laissant croire à une sélection fait au

hasard (plus petit ou égal à 25%). Pour la question 10, nous remarquons que le score global est inférieur à 50% en raison des utilisateurs qui ne sont pas familiers avec cette télésérie.
Ci-dessous, nous fournissons les détails des vidéos associées à ces questions qui seront nécessaires pour l'analyse.

Question 6

Transcription sans prétraitement[22]				
a big this is my little brother George this is musical instrument mummy pig and Daddy pig have been tidying the house Oh mummy pig and I found this old box in the Attic Oh can anyone guess what's inside hmm nope it's full of musical instruments				
Transcription avec prétraitement				
big little brother musical instrument mummy pig daddy pig tidy house mummy pig old box attic guess inside nope musical instrument				
Résumé des résultats				
	Sujet 34	Sujet 27	Sujet 2	Sujet 1
Probabilité	0,145	0,409	0,012	0,012
Décompte	15	4	0	0
Regroupement de mots	secret, box, tummy, word, daddy, guess, cookie	pig, daddy, mummy, brother, lovely, day, family	pancake, pig, daddy, mummy, flip, time, cake	shoe, red, wet, wear, garden, grass, lose

Après la lecture des transcriptions et des regroupements de mots, nous remarquons que la majorité des utilisateurs ont sélectionné le sujet 34, même si LDA a accordé une probabilité plus élevée au sujet 27 pour décrire cette scène.

[22] Cette séquence vidéo débute à 00:02 et se termine à 00:41 (https://www.youtube.com/watch?v=r13Cv2h23s&feature=youtu.be&ab_channel=UCdRyB_BAJDfo0clpilGkFzQ)

Tout comme à la question 2 de la section précédente, un des sujets (27) concerne les principaux personnages de la télésérie et le second sujet (34) consiste en l'activité qu'ils accomplissent. Basé sur le décompte des mots, le sujet 27 semble être plus présent, cependant, nous jugeons que les participants ayant sélectionné le sujet 34 n'ont pas tort.

Question 10

Transcription sans prétraitement[23]				
don't worry daddy pig is using a bucket to catch the drips well done daddy pig easy as pie hey what quick find something else to catch the water well done Peppa Bobby the bags are very loud it's okay children don't be frightened				
Transcription avec prétraitement				
worry daddy pig use bucket catch drip daddy pig easy pie quick catch water mummy bag loud okay child frighten				
Résumé des résultats				
	Sujet 27	Sujet 2	Sujet 12	Sujet 1
Probabilité	0,263	0,015	0,180	0,015
Décompte	9	1	3	6
Regroupement de mots	pig, daddy, mummy, brother, lovely, day, family	pancake, pig, daddy, mummy, flip, time, cake	catch, ball, piggy, throw, play, game, middle	shoe, red, wet, wear, garden, grass, lose

Après la lecture des transcriptions, des regroupements de mots et des réponses des participants, nous pouvons remarquer une ambiguïté quant à l'identification de la réponse attendue.

[23] Cette séquence vidéo débute à 03:08 et se termine à 03:41 (https://www.youtube.com/watch?v=tm8nM_E3Ikc&feature=youtu.be&ab_channel=UCdRyB_BAJDfo0clpilGkFzQ)

Tout comme aux questions précédentes, le sujet 27 concerne les principaux personnages de la télésérie et le second sujet (12) consiste en l'activité qu'ils accomplissent, c'est-à-dire attraper des gouttes d'eau. Cependant, un plus grand nombre de participants ont identifié le sujet 1 comparativement au sujet 12 comme celui décrivant le mieux la séquence vidéo. Ce choix a probablement été influencé par le terme « *wet* » qui a un lien les gouttes d'eau.

Bien que les regroupements de mots identifiés par LDA ainsi que leur probabilité respective nous semblent cohérents avec la transcription, l'identification du regroupement de mots étant le plus cohérent avec cette séquence vidéo a été plus difficile.

7.3.1.3 Questions du niveau de difficulté difficile

Le graphique suivant montre pour chacune des questions la répartition des réponses des utilisateurs. La couleur rouge représente une mauvaise réponse et le vert la réponse attendue.

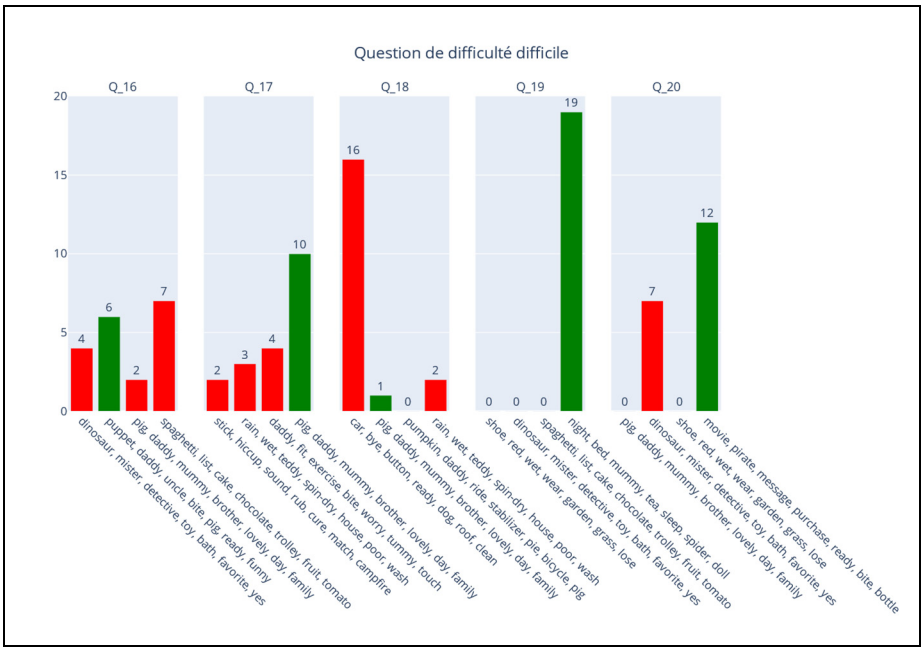


Figure 7.12 Répartition des réponses données par les participants aux questions du niveau de difficulté difficile

Le tableau ci-dessous présente les scores aux cinq questions de niveau difficile.

Tableau 7.4 Score des participants aux questions du niveau de difficulté difficile

	Questions				
	16	17	18	19	20
Global	31,58%	52,63%	5,26%	100,00%	63,16%
Familier avec *Pappa-Pig*	50,00%	62,50%	12,50%	100,00%	62,50%
Non familier avec *Pappa-Pig*	18,18%	45,45%	0,00%	100,00%	63,64%

Globalement, nous pouvons remarquer que la moyenne des scores pour les exercices de ce niveau de difficulté (51%) est inférieure à celle des deux autres niveaux de difficulté. Nous pouvons également remarquer que les scores globaux aux questions 16 et 18 sont inférieurs à 50%. Ci-dessous, l'analyse des réponses des participants à ces questions.

Question 16

Transcription sans prétraitement[24]				
George a dinosaur puppet it needs pointy taste then scary dinosaur coming after lunch we'll do a puppet show auntie peg has made spaghetti for lunch this spaghetti is best spaghetti ever oh you can tell you to a brothers alike in every way what do				
Transcription avec prétraitement				
dinosaur puppet need pointy taste scary dinosaur come lunch puppet pig make spaghetti lunch spaghetti good spaghetti tell brother alike way				
Résumé des résultats				
	Sujet 23	Sujet 3	Sujet 27	Sujet 19
Probabilité	0,084	0,226	0,084	0,155
Décompte	4	6	2	7
Regroupement de mots	dinosaur, mister, detective, toy, bath, favorite, yes	puppet, daddy, uncle, bite, pig, ready, funny	pig, daddy, mummy, brother, lovely, day, family	spaghetti, list, cake, chocolate, trolley, fruit, tomato

[24] Cette séquence vidéo débute à 02:17 et se termine à 02:53 (https://www.youtube.com/watch?v=7ZnBf8HH5k&feature=youtu.be&ab_channel=UCdRyB_BAJDfo0clpilGkFzQ)

Après la lecture des transcriptions, des regroupements de mots et des réponses des participants, nous pouvons remarquer que l'ensemble des sujets sont cohérent avec la séquence vidéo. Dans le même ordre d'idée que les analyses précédentes, c'est majoritairement le sujet 3 qui permet de décrire la scène. Cependant, les sujets 19 et 23 viennent bonifier la description de la scène par les mots « *spaghetti* » et « *dinosaur* ». Le sujet 27 quant à lui, fait référence à l'énumération des principaux personnages de cette télésérie.

En comparant les différentes probabilités des regroupements de mots à celles des tests des niveaux de difficulté facile et moyen, nous observons que les probabilités de cette question sont beaucoup plus proches, ce qui rend l'identification à un sujet dominant plus difficile. Toutefois, après l'analyse de cette séquence vidéo, les regroupements de mots identifiés par LDA ainsi que leur probabilité respective nous semblent cohérents avec la transcription.

Question 18

Transcription sans prétraitement[25]				
the car yes if you want to daddy pig has some warm soapy water to wash the car daddy pig is watching aloof mummy pig is washing the bonnet pepper is washing the doors George wants to wash the windows but he is too little poor George let me help you oh dear George has				
Transcription avec prétraitement				
car yes want daddy pig warm soapy water wash car daddy pig watch aloof mummy pig wash bonnet wash door want wash window little poor let help dear				
Résumé des résultats				
	Sujet 8	Sujet 27	Sujet 10	Sujet 36
Probabilité	0,121	0,286	0,121	0,066
Décompte	16	1	0	2
Regroupement de mots	car, bye, button, ready, dog, roof, clean	pig, daddy, mummy, brother, lovely, day, family	pumpkin, daddy, ride, stabilizer, pie, bicycle, pig	rain, wet, teddy, spin-dry, house, poor, wash

Après la lecture des transcriptions, des regroupements de mots et des réponses des participants, nous pouvons remarquer que ce sont les sujets 8, 27 et 36 qui sont les plus cohérents avec la séquence vidéo.

Dans le même ordre d'idée que les analyses précédentes, le sujet 27 concerne les principaux personnages de la télésérie et les sujets 8 et 36 consistent en l'activité qu'ils accomplissent,

[25] Cette séquence vidéo débute à 01:16 et se termine à 01:55 (https://www.youtube.com/watch?v=2gUp6ygqIQ&feature=youtu.be&ab_channel=UCdRyB_BAJDfo0clpilGkFzQ)

c'est-à-dire de nettoyer la voiture. Bien qu'aucun participant n'ait sélectionné le sujet 10, LDA attribue à ce sujet la même probabilité qu'au sujet 8 (12%). En regardant les termes du sujet 10, nous remarquons que les termes « *daddy* » et « *pig* » apparaissent quelques fois dans la transcription et bien que le terme « *ride* » n'apparaisse pas, ce terme a un sens avec le terme « car » qui est mentionné à quelques reprises.

Finalement, l'identification du sujet étant le plus cohérent avec cette séquence vidéo nous semble plus difficile. Cependant, après l'analyse de cette séquence vidéo, les sujets identifiés par LDA ainsi que leur probabilité respective nous semblent cohérents avec la transcription.

7.3.1.4 Discussion

Globalement, ce test nous permet de conclure que, pour cet échantillon de vidéos sélectionnées, les regroupements de mots les plus probables de décrire une séquence vidéo identifiée par LDA sont cohérents avec cette dernière.

L'analyse des résultats nous permet de conclure que l'utilisation de LDA a un potentiel dans un contexte de recommandations vidéo, puisque le sujet dominant associé une vidéo est représentatif de cette dernière.

Cependant l'analyse des résultats nous a permis d'identifier le regroupement de mots 27 comme étant le sujet qui porte le plus à confusion, car ce regroupement de mots fait référence aux personnages principaux de la télésérie en plus d'être présent dans la séquence vidéo.

7.3.2 Le mot intrus (word intrusion)

La tâche d'identification du mot intrus permet d'évaluer si un regroupement de mots présente une cohérence sémantique identifiable pour des participants humains. Ce test est constitué de 38 questions où chacune des questions est liée à un regroupement de mots décrivant un sujet

découvert par LDA. Pour chacun de ces regroupements de mots, nous avons présenté les trois mots les plus probables de décrire ce sujet et avons ajouté un terme intrus. Les participants doivent alors identifier le mot intrus parmi quatre choix de réponses.

Comme au test précédent, nous avons calculé un score moyen pour l'ensemble des participants, où chaque question a une pondération d'un point. La figure suivante montre la distribution des résultats obtenus pour l'ensemble des participants. En regardant le trait horizontal pointillé, on peut y voir une moyenne globale de 54% avec un écart-type de 6%. Tel qu'attendu, la moyenne et le quartile 1 des participants connaissant la série (56% et 54% respectivement) sont plus élevés que ceux décrivant les résultats globaux (54% et 50% respectivement) ainsi que de ceux où les participants sont non familiers avec la série (53% et 47% respectivement). Les points à gauche des représentations de boîtes à moustache (*box plot*) sont les résultats individuels des participants.

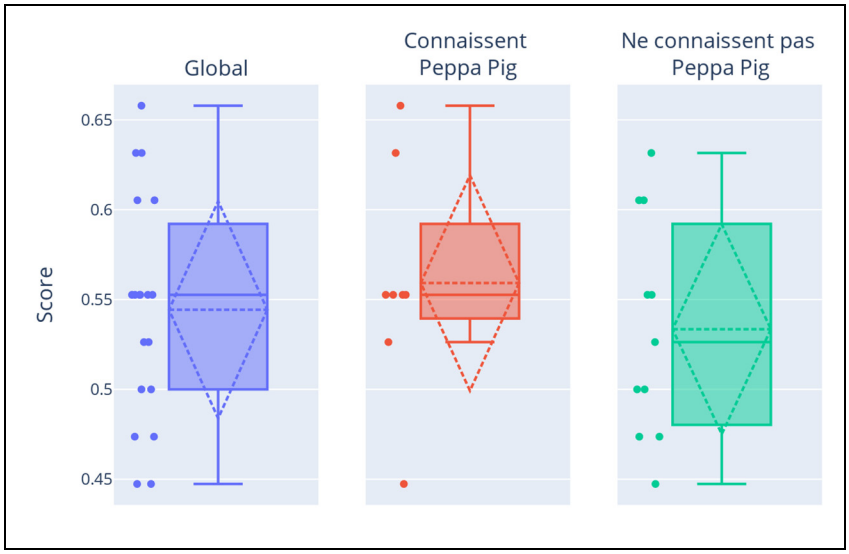


Figure 7.13 Distribution des résultats obtenus par les participants au test du mot intrus (word intrusion)

En assumant qu'un utilisateur aurait répondu aléatoirement à chacune de ces questions, cet utilisateur aurait probablement obtenu un score final de 25%. En comparant ce score aléatoire à la moyenne générale obtenue par les utilisateurs (54%), nous observons que la majorité des

différents regroupements de mots semblent induire une cohérence sémantique chez les participants.

La liste ci-dessous énumère des raisons pouvant expliquer les difficultés rencontrées pour identifier le mot intrus. Afin d'analyser les résultats, nous avons regardé les questions moins bien réussies (taux de réussite inférieur à 50%) et nous avons assigné chacune de ces questions à la raison qui semble expliquer le mieux la difficulté rencontrée.

- Les termes proviennent d'un corpus dédié à des enfants de bas âge où les adultes actuels n'ont pas connu ces vidéos dans leur enfance. Par conséquent, l'identification de l'intrus risque d'être difficile. En effet, un adulte connaissant peu ou pas la série utilisera ces connaissances générales plutôt que le contexte de la série. Par exemple, à la question 2, les quatre termes présentés sont « *brother, daddy, pig, pancake* ». Dans ce cas, le mot intrus est « *brother* », cependant presque tous les participants ont sélectionné le mot « pancake ». Dans la première saison de cette télésérie, un épisode est dédié à « *Daddy Pig »* qui cuisine des crêpes. Nous supposons que les participants ont sélectionné le mot « *pancake* », puisqu'ils ne connaissent pas cet épisode. Cette raison s'applique également aux épisodes 10, 11, 13, 14, 21, 22, 25, 29, 34 et 35.
- Similairement à la raison précédente, bien que nous ayons appliqué un traitement particulier aux noms des personnages de la série, certaines associations liées à la nomenclature se retrouvent dans les regroupements de mots. Par exemple, à la question 9, les quatre termes présentés sont « *garden, bear, brown, doctor* ». Dans ce cas, le mot intrus est « *garden* », puisqu'un des personnages est l'ours brun docteur. Pour cette question, seulement six participants sur 19 ont identifié le terme intrus attendu.
- Les images véhiculées dans les séquences vidéo ne sont pas prises en compte dans notre méthodologie, dû à l'usage unique de LDA. De ce fait, un participant considère les images et non seulement les transcriptions dans sa réponse. Par exemple, à la question 8, les quatre termes présentés sont « *red, button, bye, car*». Dans ce cas, le

mot intrus est « *red* », cependant la majorité des participants ont sélectionné le mot « *bye*». Nous supposons que les participants n'ont pas identifié le mot « *red* » puisque l'automobile est rouge. De plus, dans les épisodes où l'automobile est présente, le mot « *red* » est mentionné moins souvent que les autres termes. Cette raison s'applique également à l'épisode 30.

- Les participants se basent sur leur a priori, voire les souvenirs que ces mots peuvent leur rappeler afin d'identifier un groupe de mots cohérents (trois mots parmi les quatre) pour déduire l'intrus. Par exemple, à la question 16, les quatre termes présentés sont « *bottle, game, bite, play* ». Dans ce cas, le mot intrus est « *bottle* », cependant la majorité des participants ont sélectionné le mot « *bite* ». Nous pensons que ce regroupement de mots a fait remonter des souvenirs de jeunesse en lien au jeu populaire de la bouteille aux participants. Si tel est le cas, le terme « *bite* » n'est pas approprié (dans le contexte du jeu de la bouteille) et pourrait expliquer son identification.
- Dans certains cas, la probabilité du mot intrus n'était pas loin du mot avec la probabilité la moins élevée du regroupement de mots. Par exemple, à la question 33, les quatre termes sont « *silly, picture, wall, daddy* ». Dans ce cas, le mot intrus est « *silly* » avec une probabilité de 1,4% et le mot « *picture* » (qui est le mot avec la probabilité la moins élevée du regroupement de mots) a une probabilité de 3,9%. Cette petite différence rend cette question plus difficile. Cette raison s'applique également aux épisodes 17 et 23.

De manière plus générale par rapport à ce test, nous croyons que l'engagement des participants à vouloir identifier le mot intrus peut avoir été affecté par le grand nombre de sujets à valider et par le fait que c'était le dernier test et que certains regroupements de mots ne peuvent leur évoquer aucune cohérence.

Une dernière raison pouvant expliquer les difficultés rencontrées afin de cerner la cohérence des sujets peut provenir de notre objectif de paramétrisation du modèle. En effet, nous avons

utilisé une fonction objective (*R-square*) ayant la particularité de faire émerger un plus grand nombre de sujets latents. Une autre fonction objective aurait pu identifier un nombre moindre de sujets (*Silhouette, Coherence*) et donc, créer des sous-ensembles de regroupements plus faciles à identifier en termes de cohérence. À l'article présenté au chapitre 4, nous avons démontré que les sujets identifiés par la fonction objective utilisée (R-Square) peuvent être agglomérés en sujets plus généraux. Avoir plusieurs sujets spécialisés mais connexes (si sous la même ramification) génère une certaine difficulté. La représentation ci-dessous illustre la représentation hiérarchique du modèle utilisé :

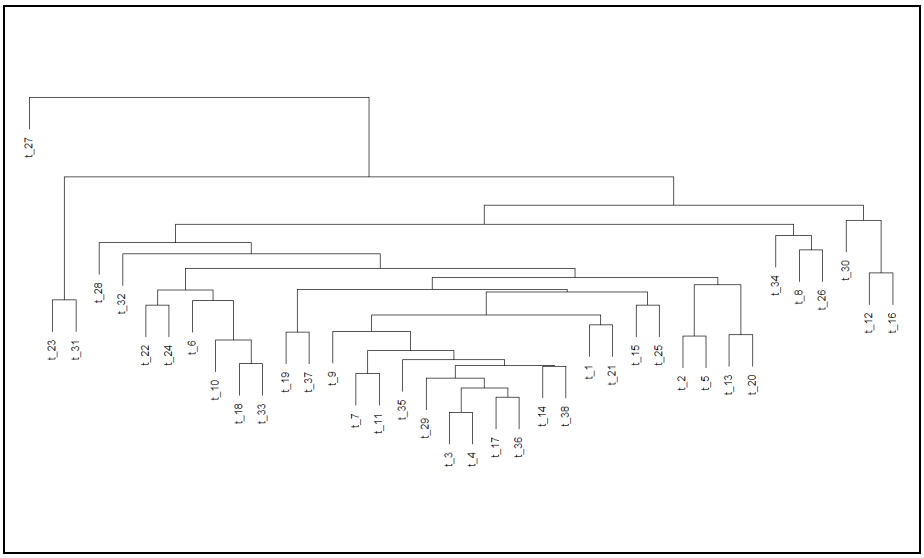


Figure 7.14 Représentation hiérarchique du modèle utilisé

Ci-dessous se trouvent les graphiques montrant la répartition des réponses des utilisateurs ainsi que les tableaux présentant les scores. Pour les graphiques, la couleur rouge représente une mauvaise réponse et le vert la réponse attendue.

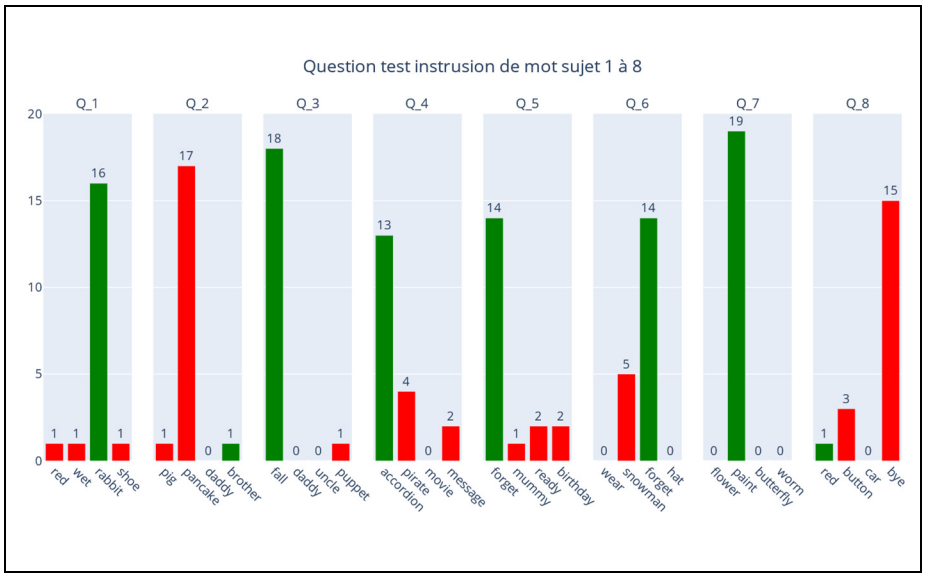


Figure 7.15 Répartition des réponses données par les participants aux questions 1 à 8

Tableau 7.5 Score des participants aux questions 1 à 8

	Questions							
	1	2	3	4	5	6	7	8
Global	84,21%	5,26%	94,74%	68,42%	73,68%	73,68%	100,00%	5,26%
Familier avec *Pappa Pig*	75,00%	0,00%	100,00%	75,00%	62,50%	87,50%	100,00%	12,50%
Non familier avec *Pappa Pig*	90,91%	9,09%	90,91%	63,64%	81,82%	63,64%	100,00%	0,00%

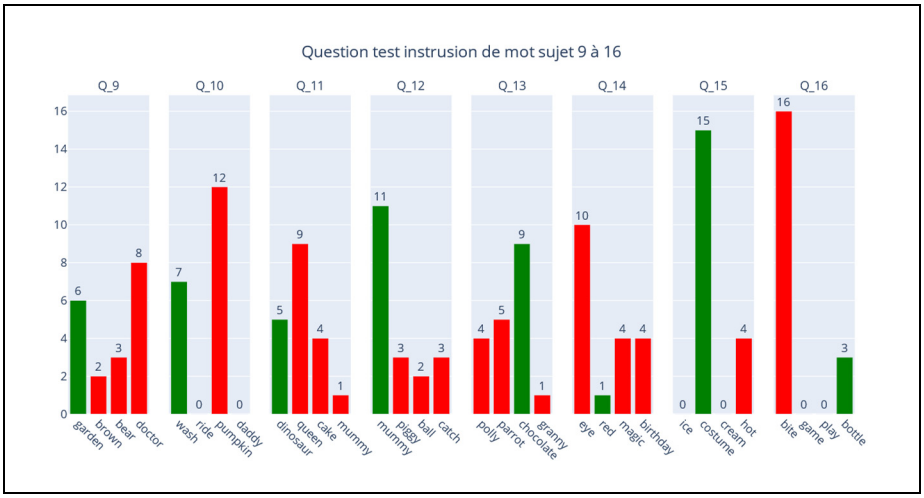


Figure 7.16 Répartition des réponses données par les participants aux questions 9 à 16

Tableau 7.6 Score des participants aux questions 9 à 16

	Questions							
	9	10	11	12	13	14	15	16
Global	31,58%	36,84%	26,32%	57,89%	47,37%	5,26%	78,95%	15,79%
Familier avec *Pappa Pig*	37,50%	37,50%	25,00%	62,50%	37,50%	12,50%	87,50%	25,00%
Non familier avec *Pappa Pig*	27,27%	36,36%	27,27%	54,55%	54,55%	0,00%	72,73%	9,09%

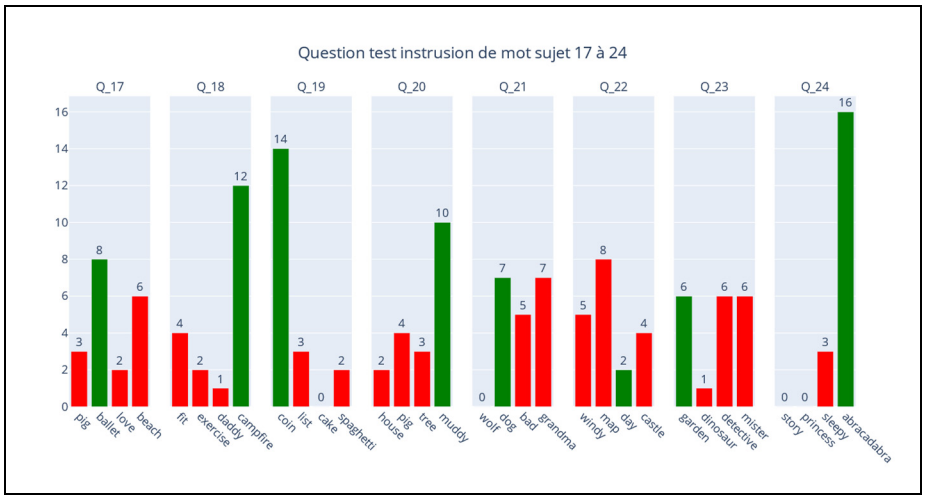


Figure 7.17 Répartition des réponses données par les participants aux questions 17 à 24

Tableau 7.7 Score des participants aux questions 17 à 24

	Questions							
	17	18	19	20	21	22	23	24
Global	42,11%	63,16%	73,68%	52,63%	36,84%	10,53%	31,58%	84,21%
Familier avec *Pappa Pig*	25,00%	62,50%	75,00%	75,00%	50,00%	0,00%	25,00%	100,00%
Non familier avec *Pappa Pig*	54,55%	63,64%	72,73%	36,36%	27,27%	18,18%	36,36%	72,73%

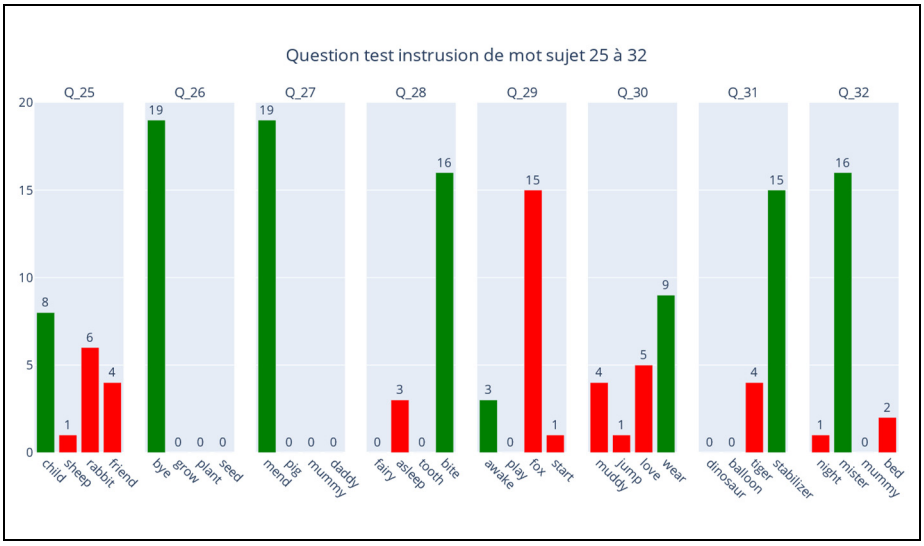


Figure 7.18 Répartition des réponses données par les participants aux questions 25 à 32

Tableau 7.8 Score des participants aux questions 25 à 32

	Questions							
	25	26	27	28	29	30	31	32
Global	42,11%	100,00%	100,00%	84,21%	15,79%	47,37%	78,95%	84,21%
Familier avec *Pappa Pig*	62,50%	100,00%	100,00%	75,00%	25,00%	50,00%	87,50%	75,00%
Non familier avec *Pappa Pig*	27,27%	100,00%	100,00%	90,91%	9,09%	45,45%	72,73%	90,91%

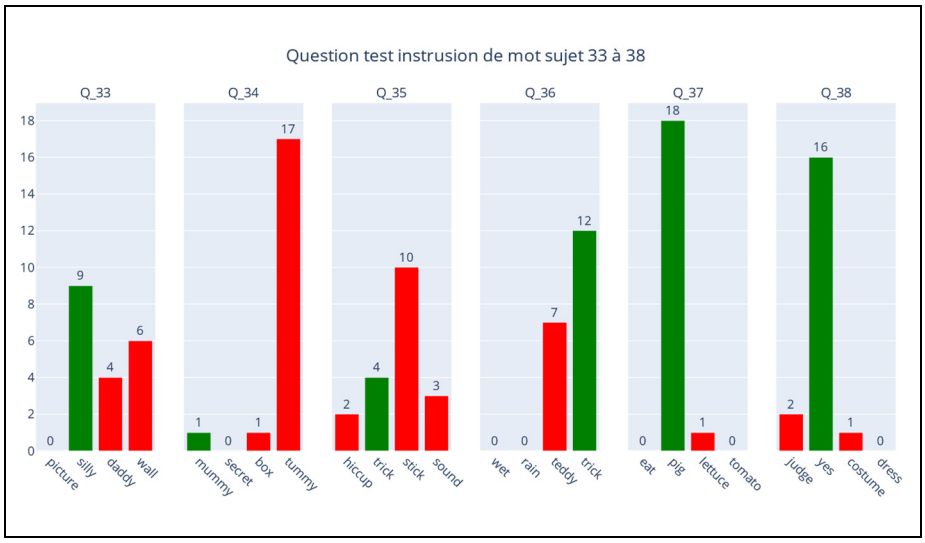


Figure 7.19 Répartition des réponses données par les participants aux questions 33 à 38

Tableau 7.9 Score des participants aux questions 33 à 38

	Questions					
	33	34	35	36	37	38
Global	47,37%	5,26%	21,05%	63,16%	94,74%	84,21%
Familier avec *Pappa Pig*	37,50%	0,00%	12,50%	62,50%	100,00%	87,50%
Non familier avec *Pappa Pig*	54,55%	9,09%	27,27%	63,64%	90,91%	81,82%

7.3.2.1 Discussion

Comme au test précédent, si un utilisateur avait répondu de manière aléatoire à chacune des questions, il aurait probablement obtenu un score final de 25%. En comparant ce score aléatoire à la moyenne générale obtenue par les utilisateurs (54%), nous pouvons conclure que, pour la majorité des ensembles de mots découverts par l'optimisation de notre modèle, ceux-ci présentent une cohérence sémantique identifiable par l'humain.

Nous avons noté plusieurs raisons pouvant expliquer les difficultés rencontrées afin d'identifier le mot intrus dans un regroupement de mots. Ces dernières sont divisées en deux grandes catégories : le profil des participants (niveau de familiarisation avec la télésérie et a priori (expérience antérieure)) et les caractères techniques du test (nombre de questions, choix de la fonction objective, sélection du mot intrus et nature du corpus (sous-titres)).

Après l'analyse des raisons expliquant les difficultés rencontrées, nous nous sommes questionnés sur la forme utilisée de ce test. Une alternative afin d'évaluer la cohérence des sujets serait de demander aux participants de passer en revue ces différents regroupements afin de déterminer s'ils sont cohérents ou non et ensuite, de résumer l'ensemble de ces termes en un terme ou une combinaison de termes plus généraux (étiquette). Le tableau ci-dessous présente un exemple de ce test alternatif où nous proposons une étiquette à chacune de ces listes de termes.

Termes / **Étiquette**	1	2	3	4	5	6	7	8	9	10
snowman	*hat*	*snowman*	*wear*	*glove*	*scarf*	*snow*	*cold*	*warm*	*daddy*	*clothe*
Butterfly / worm	*butterfly*	*worm*	*flower*	*regular*	*wriggly*	*fly*	*beautiful*	*queen*	*lesson*	*tongue*
Grocery	*list*	*spaghetti*	*chocolate*	*cake*	*trolley*	*fruit*	*tomato*	*love*	*onion*	*shop*

7.4 Conclusion

Ce chapitre nous a permis de présenter et d'analyser les résultats collectés par notre application et donc de compléter le sous-objectif 5. L'application a permis d'évaluer deux aspects liés à ce projet de recherche. Le premier aspect concerne l'utilisation de la modélisation des thématiques inférées par LDA sur un corpus constitué des sous-titres de vidéos pour enfants dans un contexte de recommandations vidéo ainsi que l'utilisation de la carte sémantique en

tant que support visuel. Le second aspect concerne la cohérence des sujets latents identifiés par LDA à la suite de l'application de notre méthode d'optimisation de ses paramètres.

Concernant le premier aspect, l'évaluation du système de recommandations, plus de la moitié des participants ont répondu dans l'affirmative à toutes les questions du formulaire. Cela nous permet de conclure que le système de recommandations semble correctement générer des vidéos en lien avec la thématique de départ sélectionnée et que la carte sémantique semble être utile dans les contextes de recherche et de sélection de vidéos ainsi que pour décrire les contenus des vidéos regardées.

Concernant le second aspect, l'évaluation de la cohérence des sujets latents identifiés par LDA, nous avons basé notre évaluation sur les tests d'intrusion proposés par Chan *et al.* (2009) (*topic intrusion* et *word intrusion*).

Pour le test du sujet intrus (*topic intrusion*), les participants ont répondu à 20 questions (dont cinq de niveau facile, dix de niveau moyen et cinq de niveau difficile). Le tableau ci-dessous présente les résultats moyens par niveau de difficulté :

Tableau 7.10 Résultats moyens par niveau de difficulté pour le test de regroupement de mots (topic intrusion)

	Facile (questions 1 à 5)	**Moyen (questions 6 à 15)**	**Difficile (questions 16 à 20)**
Global	85,26%	76,32%	50,53%
Familier avec *Pappa-Pig*	87,50%	81,25%	57,50%
Non familier avec *Pappa-Pig*	83,64%	72,73%	45,45%

Toutes les moyennes globales sont bien au-dessus d'une moyenne aléatoire (25%). De manière générale, cela nous permet de confirmer la pertinence des distributions produites par LDA. De plus, nous avons analysé les questions pour lesquelles plus de la moitié des participants ont

incorrectement choisi le regroupement de mots qui décrit le mieux la séquence vidéo afin de s'assurer que les regroupements de mots identifiés par LDA ainsi que leur probabilité respective soient cohérents. Ces résultats nous permettent également de valider la méthode de création des niveaux de difficulté des tests, puisque les moyennes diminuent lorsque le niveau de difficulté augmente. Finalement, ce tableau nous permet de constater que plus le niveau de difficulté augmente, plus les écarts entre les participants familiers et ceux qui ne le sont pas augmentent.

Pour le test du mot intrus (*word intrusion*), celui-ci a été moins bien réussi que le test précédent. En comparant le score aléatoire (25%) à la moyenne générale obtenue par les utilisateurs (54%), nous pouvons conclure que, pour la majorité des ensembles de mots découverts par l'optimisation de notre modèle, ceux-ci présentent une cohérence sémantique identifiable par l'humain.

Nous avons noté deux types de raison pouvant expliquer les difficultés rencontrées : le profil des participants (niveau de familiarisation avec la télésérie et a priori (expérience antérieure)) et les caractères techniques du test (nombre de questions, choix de la fonction objective, sélection du mot intrus et nature du corpus (sous-titres)).

Après l'analyse des raisons expliquant les difficultés rencontrées, nous avons proposé une alternative afin d'évaluer la cohérence des regroupements de mots qui serait de demander aux participants de passer en revue ces différents regroupements afin de déterminer s'ils sont cohérents ou non et ensuite, de résumer l'ensemble de ces termes en un terme ou une combinaison de termes plus généraux.

Malgré les efforts mis en place afin de permettre aux participants de sauvegarder l'état de leur avancement au cours de différents exercices ainsi que la période d'un mois pour les compléter, nous avons calculé que seulement 45,23% (19/42) des participants ayant accepté le FIC ont complété l'ensemble des tests.

CHAPITRE 8

DISCUSSION GÉNÉRALE

L'objectif général de ce projet de recherche consistait à développer un système de recommandations vidéo basé sur l'analyse de sous-titres en utilisant l'approche du filtrage sur le contenu. Cette première étape s'insère dans un objectif plus vaste (le programme de recherche) où d'autres caractéristiques contenues dans les vidéos comme les sons, les images et les objets détectés pourront y être ajoutées afin d'affiner le système. Dans cette première itération, nous nous sommes consacrés à la modélisation des sous-titres, car il s'agit d'une source d'information ayant un potentiel descriptif prometteur quant à l'interprétation du contenu de la vidéo. Quant au programme de recherche dans lequel s'introduit ce projet de recherche, celui-ci a pour but d'apporter une solution innovante et responsable afin de soutenir le développement linguistique des enfants autistes d'âge préscolaire. L'avantage de développer ce type de technologie est qu'il sera facile d'y accéder via des appareils électroniques connus et largement répandus (tablettes, cellulaires). Même si l'utilisation de ces appareils est une préoccupation croissante dans le domaine de la pédopsychiatrie, une utilisation supervisée pourrait être bénéfique pour le développement de ces enfants. De plus, nous croyons que l'élaboration de cette solution répondra à deux problématiques inhérentes à l'autisme comme la difficulté d'interaction due aux limitations de la communication verbale et la limitation de leurs intérêts.

L'atteinte de l'objectif général de ce projet recherche a été scindé en ces 5 sous-objectifs :

- SO-1 : Extraction des sous-titres des vidéos en ligne
- SO-2 : Optimisation du modèle de langue Latent Dirichlet Allocation (LDA)
- SO-3 : Optimisation d'un modèle de langue sur un corpus constitué de transcriptions automatiques de vidéo pour enfant
- SO-4 : Génération et visualisation des recommandations vidéo et des centres d'intérêt de l'utilisateur

- SO-5 : Évaluation du système par des utilisateurs

Le SO-1 nous a permis de développer une méthodologie afin de concevoir le corpus à l'étude et de l'analyser. Pour ce faire, nous avons élaboré un mécanisme d'extraction automatique des sous-titres des vidéos mis en ligne sur la plateforme YouTube (section 3.1). Ce mécanisme nous a permis de comparer différents outils d'extraction (*YouTube-dl, Pytube, YouTube-transcript-api*) quant à leur capacité à extraire les sous-titres des vidéos en provenance de différentes chaînes YouTube (*PeppaPigSurprise, PeppaPigSubtitled, RexyKids, PeppaPigAsia*). Cette comparaison nous a permis d'identifier la combinaison de l'outil *YouTube-dl* et de la chaîne *PeppaPigSubtitled* comme celle offrant le plus haut niveau de similarité entre les sous-titres extraits et notre valeur témoin. De plus, nous nous sommes questionnés sur la notion du droit d'auteur quant à l'utilisation de ce type d'outil. Nous en avons déduit que nous sommes autorisés à utiliser ce genre d'outil pourvu que le contexte d'utilisation de ces sous-titres demeure dans un contexte d'usage loyal par rapport à l'auteur et que l'utilisation de ces sous-titres n'engendre aucun produit commercialisable.

Une fois l'extraction des sous-titres complétée, nous avons développé un mécanisme de découpage d'un épisode en courtes séquences vidéo. Ce mécanisme est basé sur l'enchaînement du nombre de répliques pour concevoir une séquence vidéo. À la suite de l'analyse de la durée et du nombre de termes décrivant les séquences produites, nous avons sélectionné trois valeurs candidates (section 3.2). L'idée de fragmenter un épisode en plusieurs séquences est qu'un épisode peut contenir plusieurs thématiques différentes et que nous voulons être en mesure d'émettre à l'utilisateur de courtes séquences vidéo qui abordent des thématiques précises.

L'auteur a nommé les personnages principaux de la télésérie en utilisant des répétitions de termes liés aux espèces animales. Combiné au choix du modèle de langue que nous avons sélectionné pour bâtir le système de recommandations, ce type d'accentuation risque de perturber sa stabilité. Cette perturbation est causée par les fréquences anormalement élevées

comparativement aux autres termes qui décrivent le vocabulaire du corpus. Puisque le nom des personnages est une source d'information pertinente pour décrire une scène, nous avons évalué l'impact de cinq techniques de substitution des noms des personnages sur la stabilité de la modélisation finale (section 3.3). Finalement, ce sous-objectif a été complété et nous a permis d'identifier 3 valeurs candidates de fragmentation des épisodes auxquelles nous avons appliqué les cinq différentes techniques de substitution des noms des personnages afin de concevoir 15 sac-de-mots différents tous basés sur le corpus à l'étude. Ceux-ci ont ensuite été modélisés et évalués au SO-3 afin d'identifier la modélisation optimale à notre contexte d'utilisation. Nous croyons que cette méthodologie qui est détaillée au chapitre 3 a permis d'atteindre ce premier sous-objectif et qu'elle pourrait servir de cadre pour des projets similaires.

En ce qui concerne la modélisation des sous-titres, nous avons priorisé l'utilisation de l'algorithme LDA comme modèle de langue, car c'est un modèle non supervisé et qu'il permet de faire émerger des sujets contenus dans un corpus de texte. Le fait qu'il soit non supervisé lui donne l'avantage d'être utilisable sur n'importe quel ensemble de texte sans avoir à passer par une étape d'annotation du corpus à l'étude. De plus, sa capacité à faire émerger automatiquement des sujets latents sous une forme probabiliste permet de concevoir les bases pour le calcul des recommandations. Cependant, son utilisation et, plus particulièrement le choix de ses paramètres, reste une tâche complexe, subjective et couteuse en temps.

Puisqu'il n'existe aucune paramétrisation générale adaptée pour tous les corpus, nous avons dû nous attarder à cette tâche. L'élaboration d'un cadriciel permettant d'identifier une paramétrisation quasi optimale de ce modèle de langue réfère au SO-2. L'approfondissement de ce sous-objectif nous a permis de développer une approche automatique de recherche des paramètres à l'aide de l'algorithme de recherche communément appelé l'Algorithme Génétique. Afin de guider l'Algorithme Génétique à identifier des possibilités de paramètres, nous avons utilisé différentes fonctions objectives telles que celles usuellement utilisées comme *Silhouette* et *Coherence* et des moins connues comme *R-Square et Raw score* et avons

même proposé une nouvelle méthode que nous avons nommée *2PJ*. Dans l'élaboration de ce cadriciel, nous avons démontré que l'utilisation des fonctions objectives moins connues ont un potentiel à découvrir plus de sujets latents comparativement à celles communément employées. Nous avons également abordé la question de la stabilité du modèle afin de développer une approche basée sur les travaux Mantyla *et al.* (2018) et Rierger *et al.* (2020) pour quantifier la stabilité de la modélisation décrite par le jeu de paramètres.

Globalement, nous croyons que le développement de ce cadriciel est plus complet puisqu'il aborde la recherche de tous les paramètres de LDA comparativement à d'autres solutions proposées. Par exemple, certaines solutions se concentrent uniquement à identifier un nombre de sujets (paramètre K) et imposent le choix des paramètres α et β et, en plus, n'abordent pas la question de la stabilité de la modélisation finale. Les détails de ce cadriciel a été soumis à la revue *Journal of Machine Learning Research* et est présenté au chapitre 4. En termes de travaux futurs, le cadriciel proposé offre plusieurs pistes d'amélioration (telles que discutées en sein de l'article) grâce à ses deux étapes distinctes identifiées comme étant la recherche des paramètres quasi optimaux et l'analyse de stabilité. De plus, la solution actuelle a été développée sous le langage R, mais pourrait être traduite en Python puisque ce langage de programmation semble être plus populaire. Finalement, nous pouvons conclure que ce sous-objectif a été atteint.

Le SO-3 consistait principalement à modéliser les différents sac-de-mots produits au SO-1 à l'aide de la méthode préalablement développée au SO-2 et à identifier une modélisation optimale pour concevoir la base des calculs des recommandations et d'estimation du centre d'intérêt de l'utilisateur. L'ensemble de cette démarche est présenté au chapitre 5. Afin d'identifier la modélisation optimale, nous avons posé les trois critères suivants : maximiser le nombre de sujets (K), maximiser le score en stabilité et minimiser les ratios de partage. Le ratio de partage est une métrique que nous avons élaborée afin de comparer des modélisations issues de différents sac-de-mots entre elles, car l'analyse en stabilité ne permet pas cette comparaison.

Bien que l'analyse en stabilité permette de comparer différentes modélisations issues d'un même sac-de-mots à l'aide du score en stabilité, elle ne permet pas de comparer différentes modélisations issues de sac-de-mots différents. De plus, l'analyse en stabilité se concentre à évaluer si la paramétrisation permet une reproductibilité des ensembles de mots les plus probables décrivant chaque sujet, mais ne considère pas la seconde distribution que LDA génère, c'est-à-dire la distribution document-sujets. Le ratio de partage consiste donc à calculer le niveau de partage des éléments communs pour les éléments les plus probables décrivant la matrice sujet-termes et la matrice document-sujets. À l'aide des trois critères de sélection, nous avons été en mesure de sélectionner la modélisation finale et avons remarqué que cette modélisation concorde avec celle qui a obtenu le score en stabilité le plus élevé. Par cette correspondance, nous croyons que la métrique du score en stabilité est pertinente pour la sélection du modèle final. Finalement, nous pouvons conclure que ce sous-objectif a été atteint.

Une fois le modèle sélectionné, nous avons développé une application Web afin de procéder à une étude pilote avec un groupe de participants neurotypiques recrutés au sein de notre communauté universitaire. L'élaboration de cette application et l'analyse des données recueillie s'insèrent respectivement sous les sous-objectifs 4 et 5. Dans ce mémoire, nous avons principalement détaillé le calcul des recommandations et d'estimation du centre d'intérêt de l'utilisateur, l'élaboration de la carte sémantique ainsi que les tests d'intrusion.

Le calcul des recommandations et d'estimation du centre d'intérêt de l'utilisateur est le cœur du système de recommandations. Il peut être vu comme étant un problème d'optimisation à résoudre qui considère l'historique des visualisations de l'utilisateur et la distribution document-sujets décrivant les différentes séquences vidéo. Nous nous sommes inspirés de la solution proposée par Zhu *et al.* (2013) pour aborder ce problème. Cependant, la formulation qu'ils ont proposée afin d'estimer le centre d'intérêt de l'utilisateur assume que les vidéos de l'historique des recommandations seraient toutes classifiées comme étant des vidéos pertinentes, ce que nous croyons qui n'est usuellement pas le cas. De plus, nous croyons que

les vidéos qui seraient classifiées comme non pertinentes sont également porteuses d'information, puisqu'elles identifient les sujets auxquels nous devons diminuer les probabilités pour les prochaines recommandations. Pour ce faire, nous avons proposé une nouvelle formulation qui considère les vidéos classifiées par l'utilisateur comme pertinentes et non pertinentes afin de mieux cerner le centre d'intérêt. Telle que présentée à la section 6.3.1, l'estimation du centre d'intérêt est décrite comme suit :

$$\overrightarrow{Centre\ d'intérêt} = \overrightarrow{Pertinent} + \Delta \tag{8.1}$$

$$\Delta = \overrightarrow{Pertinent} - \overrightarrow{Non\ Pertinent} \tag{8.2}$$

Ce qui est équivalent à :

$$\overrightarrow{Centre\ d'intérêt} = \overrightarrow{Pertinent} + \overrightarrow{Pertinent} - \overrightarrow{Non\ Pertinent} \tag{8.3}$$

$$\overrightarrow{Centre\ d'intérêt} = 2 * \overrightarrow{Pertinent} - 1 * \overrightarrow{Non\ Pertinent} \tag{8.4}$$

$$\overrightarrow{Centre\ d'intérêt} = poids\ P * \overrightarrow{Pertinent} + poids\ NP * \overrightarrow{Non\ Pertinent} \tag{8.5}$$

Les poids utilisés dans notre mécanisme d'estimation d'un nouveau centre d'intérêt pourraient être le sujet d'une recherche future. En effet, les poids actuellement appliqués aux vecteurs pertinent et non pertinent pourraient être différents et être sélectionnés en fonction de la volonté de pénaliser les vidéos non pertinentes. Dans le cadre de cette recherche, nous avons utilisé des poids de 2 et -1 respectivement pour les vecteurs pertinent et non pertinent afin de calculer le nouveau centre d'intérêt.

En ce qui concerne l'élaboration de la carte sémantique, nous avons développé une nouvelle représentation de la matrice sujet-termes issue de LDA sous la forme d'un graphe qui nous a permis d'illustrer l'évolution de l'estimation des centres d'intérêt de l'utilisateur lors de l'évaluation des vidéos qui lui était recommandées. Le développement de ce type de visualisation a été motivé par l'ajout d'un support permettant aux utilisateurs de visualiser les différentes thématiques qui caractérisent les différentes séquences vidéo contenues dans la base

de données. De plus, par cette représentation, il a été possible de positionner le centre d'intérêt de l'utilisateur dans cet ensemble de thématiques et de suivre son évolution après la classification des différentes recommandations.

En ce qui concerne les tests d'intrusion, ces tests ont été proposés par Chan *et al.* (2009) afin de vérifier si les distributions identifiées par LDA sont cohérentes avec le jugement humain. Ces auteurs ont proposé deux tests différents, le test du mot intrus (*word intrusion)* et le test du sujet intrus *(topic intrusion)*. En raison du contexte dans lequel nous avons utilisé LDA, c'est-à-dire la recommandation vidéo, nous avons modifié la formulation originale du test du sujet intrus afin de présenter la séquence vidéo associée au document au lieu de présenter la transcription qui a été utilisée pour entrainer le modèle. Nous avons justifié ce choix par le fait que nous cherchons à évaluer le potentiel d'utiliser une technique de modélisation par sujet sur les sous-titres de vidéos afin de concevoir un système de recommandations vidéo et qu'il aurait été difficile pour un participant d'extrapoler le contexte du document dû à sa longueur (peu de mots) et à la nature du corpus (dialogues entre personnages pouvant contenir des erreurs de transcription). De plus, en raison des distributions document-sujets obtenues, nous avons également modifié la tâche d'identification de l'intrus par une tâche d'identification du regroupement de mots décrivant **le mieux** la séquence vidéo. Cette adaptation a été motivée par le fait que nous avons optimisé la tâche de modélisation par sujet en ayant pour objectif que chaque séquence vidéo appartienne, dans la grande majorité, à un sujet dominant. Si nous avions appliqué la formulation du test d'intrusion tel que mentionné par Chan *et al.* (2009) (qui stipule de décrire un document par les regroupements de mots appartenant aux trois sujets qui ont les probabilités les plus élevées, en plus d'ajouter une nouvelle liste de mot identifiée comme le vrai intrus), l'utilisateur aurait probablement été indécis quant à la sélection du vrai intrus, car selon lui, il y aurait deux, voire trois intrus parmi les quatre choix qui lui sont présentés.

En plus de répondre au SO-4, le développement de cette application et de ces différentes composantes nous a permis d'évaluer deux aspects de ce projet de recherche. Le premier aspect concerne l'utilisation de la modélisation des thématiques inférées par LDA sur un corpus constitué des sous-titres de vidéos pour enfants dans un contexte de recommandations vidéo ainsi que l'utilisation de la carte sémantique en tant que support visuel. Le second aspect concerne la cohérence des sujets latents identifiés par LDA à la suite de l'application de notre méthode d'optimisation de ses paramètres.

Le dernier sous-objectif, le SO-5, consistait à évaluer les résultats des participants. Pour répondre à ce sous-objectif, nous avons demandé aux utilisateurs de répondre à une série de questions qualitatives et une série de tests d'intrusion.

Tout d'abord, l'évaluation du système de recommandations et de la carte sémantique a été réalisée à l'aide d'un formulaire composé de sept questions. Celles-ci ont permis de recueillir l'appréciation quant à la qualité des recommandations et au potentiel de la carte sémantique à exposer les différents thèmes abordés dans les sous-titres des vidéos analysées ainsi qu'à exposer les liens entre les différentes séquences vidéo identifiées comme pertinentes. Globalement, plus de la moitié des participants ont répondu dans l'affirmative à toutes ces questions. En ce qui concerne la qualité des recommandations, la majorité des participants s'accordent à dire que les recommandations correspondaient bien à leur thématique de départ. En ce qui concerne la carte sémantique, celle-ci semble être utile pour décrire les contenus des vidéos regardées. Afin d'améliorer l'interaction avec la carte, les participants ont proposé certaines fonctionnalités (discutées à la section 7.2).

En ce qui concerne l'évaluation de la cohérence des sujets latents identifiés par notre modélisation finale, les tests d'intrusion nous ont permis d'effectuer des analyses qualitatives des distributions identifiées par LDA quant à savoir si elles sont cohérentes avec le jugement humain. L'analyse des résultats au test d'identification du regroupement décrivant le mieux une séquence vidéo nous a permis de conclure que l'utilisation de LDA a un potentiel dans un

contexte de recommandations vidéo. L'analyse des résultats au test d'identification du mot intrus nous a permis de conclure que, pour la majorité des ensembles de mots découverts par l'optimisation de notre modèle, ceux-ci présentent une cohérence sémantique identifiable par l'humain. Cependant, après l'analyse des raisons expliquant les difficultés rencontrées, nous nous sommes questionnés sur la pertinence de ce test en raison du niveau de difficulté engendré par le modèle sélectionné (lequel contient beaucoup de sujets distincts qui pourraient être regroupés) et part la provenance du corpus à l'étude (majoritairement inconnu de la communauté étudiante). À la suite de cette réflexion, nous avons proposé une alternative afin d'évaluer la cohérence des regroupements de mots qui consisterait à demander aux participants de passer en revue ces différents regroupements afin de déterminer s'ils sont cohérents ou non et ensuite, de résumer l'ensemble de ces termes en un terme ou une combinaison de termes plus généraux.

CONCLUSION

Ce dernier chapitre permet de conclure sur l'ensemble de ce projet de recherche qui consistait à développer un système de recommandations vidéo basé sur l'analyse des sous-titres de vidéos afin de soutenir le développement linguistique d'enfants autistes. Nous subdivisions cette conclusion en trois sous-sections. La première sous-section précise les différentes contributions et innovations que ce projet nous a permis d'accomplir. La seconde sous-section discute des perspectives de ce projet dans le contexte de l'autisme et la dernière sous-section priorise les travaux futurs.

8.1 Contributions

Le développement de ce projet de recherche nous a amené à innover dans la discipline du traitement des langues naturelles et des systèmes de recommandations. Comme contribution dans le domaine du traitement des langues naturelles, nous pouvons identifier la méthodologie développée au chapitre 4. L'élaboration de cette méthodologie a été présentée au 88^{e} congrès de ACFAS en mai 2021 et soumise en tant qu'article scientifique au *Journal of Machine Learning Research*. Cette méthodologie a également été partagée avec un groupe de chercheurs dirigé par les professeurs Roxane de la Sablonnière, Ph. D. et Éric Lacourse, Ph. D. de l'Université de Montréal des départements de psychologie et de sociologie. Notre participation leur a permis d'extraire les thématiques dominantes sur une question ouverte lors d'une étude longitudinale au sein de la population canadienne concernant la responsabilité de la crise de la COVID-19. Notre collaboration à ce projet de recherche nous a permis de contribuer à une seconde publication qui a été soumise au *American Journal of Sociology* en novembre 2022. De plus, ces chercheurs ont comparé les résultats obtenus par une annotation manuelle à ceux identifiés par notre méthode d'optimisation du modèle LDA et ont remarqué une adéquation entre ces deux résultats.

À titre d'innovation en lien avec la représentation de la distribution sujet-termes produite par LDA, nous pouvons nommer la représentation de cette distribution sous la forme de graphe. Les représentations les plus connues et utilisées sont les nuages de mots et LDAVis, mais ne permettent pas de visualiser le partage des termes communs entre les différents sujets.

Concernant les systèmes de recommandations, nous pouvons mentionner le potentiel de la carte sémantique au sein de ce type de système. Lors de notre étude pilote, les participants ont mentionné l'utilité de cette représentation pour décrire le contenu des vidéos regardées, pour avoir un aperçu des différentes thématiques des vidéos contenues dans base de données et qu'éventuellement il serait pertinent de pouvoir contrôler le système de recommandation (sélectionner des vidéos) à partir de celle-ci. Nous pouvons également identifier une seconde innovation concernant le calcul de la mise à jour de l'estimation du centre d'intérêt de l'utilisateur, car nous avons considéré les vidéos identifiées comme non pertinentes en sus des vidéos pertinentes pour ce calcul.

Nous avons également identifié une considération non abordée dans l'approche du développement des tests d'intrusion et plus particulièrement dans le test du sujet intrus (*topic intrusion*). Dans l'approche originalement proposée, un document se voit attribuer plusieurs sujets spécifiques ayant des probabilités élevées de décrire ce document. Dans ce contexte, il est possible de demander à un utilisateur externe d'identifier un sujet intrus en lui montrant par exemple, les trois sujets les plus dominants et un sujet à faible probabilité (l'intrus). Dans notre contexte, c'est-à-dire qu'un document se voit attribuer une probabilité élevée pour un sujet et que le reste des probabilités sont basses, la formulation d'identification de l'intrus offrirait trois bonnes réponses. Nous avons alors reformulé la question en ce sens, et donc, l'utilisateur doit identifier le sujet décrivant le mieux le document affiché.

8.2 Perspective sur l'autisme

Le projet d'envergure (programme de recherche) dans lequel s'inscrit ce projet vise à apporter une solution innovante dans le milieu de l'éducation des enfants autistes. Actuellement, ce milieu est sollicité pour renouveler son programme auprès de ces enfants, car les méthodes utilisées sont encore majoritairement fixées sur des stratégies comportementales de type behavioriste qui n'utilisent pas les forces de ces enfants, ou ciblent une normalisation des comportements qui est à la fois hors d'atteinte dans l'état actuel des connaissances et non conforme aux aspirations de cette population. Les différents acteurs de ce milieu ont fréquemment noté qu'une proportion importante de ces enfants présentent un intérêt marqué pour les tablettes et les téléphones intelligents. Ces derniers s'orientent spontanément vers des applications ou des vidéos comportant des lettres, des chiffres et des formes, et ont une avance, souvent considérable, sur les enfants typiques à ce sujet. Cette exposition permet à ces enfants d'atteindre rapidement et de manière autodidacte une connaissance du code écrit et de se construire un vocabulaire qui facilite leur scolarisation plusieurs années plus tard.

La solution proposée dans ce travail s'est concentrée sur l'élaboration d'un système de recommandations basé sur l'analyse des sous-titres afin de caractériser le contenu des vidéos. De plus, la personnalisation des recommandations est rendue possible en demandant à l'utilisateur d'indiquer si la vidéo regardée est pertinente ou non à une thématique de départ. Afin d'améliorer cette première solution afin qu'elle soit en mesure de servir au milieu de l'éducation, les travaux futurs devront se concentrer sur l'approfondissement d'un mécanisme permettant d'estimer la pertinence de la vidéo lorsque l'enfant s'adonne à une tâche afin d'éliminer l'indication manuelle fait par l'utilisateur. C'est-à-dire que les enfants ne manifesteront pas leur appréciation d'une séquence vidéo en étiquetant la vidéo à l'aide d'un mécanisme de notation en ligne (cœur, étoiles, pouce). Pour cela, une des pistes de solution serait d'établir un mécanisme ayant la capacité de déduire l'émotion de l'enfant à l'aide de capteur de signaux physiologique ou non physiologique. De plus, l'analyse des sous-titres ne permet pas de décrire tous les éléments visuels d'une vidéo qui pourrait intéresser l'enfant.

Pour cela, une **analyse multimodale des vidéos** permettrait d'augmenter le niveau de détails qui améliorerait l'estimation des intérêts de l'enfant.

Concernant l'analyse multimodale des vidéos, dans cette première phase, nous nous sommes attardés à l'analyse des sous-titres des vidéos afin d'en extraire des ensembles de mots cohérents qui permettent de décrire les différentes scènes. Selon notre première expérience effectuée avec des participants neurotypiques provenant du milieu universitaire, ces ensembles de mots permettent de décrire/résumer le contenu d'une séquence vidéo d'environ 25 secondes. Cependant, il est attendu qu'un enfant s'attardera à d'autres éléments qui ne seront pas identifiables par ce type d'analyse. Pensons aux différentes couleurs, objets, personnages compris dans une scène qui, une fois détectés, amélioreront la capacité à détailler le contenu des séquences vidéo. Par exemple, il nous a été décrit par la mère d'un enfant que celui-ci portait plus d'attention au bandeau d'information en continu qu'aux images décrites par le journaliste lors de l'écoute du téléjournal.

Nous croyons que les pistes de solutions que nous venons de proposer amélioreront la personnalisation du système de recommandations dans un contexte non supervisé. Cependant, ces propositions nous font réfléchir au niveau de difficulté qui sera rencontrée afin d'élaborer une solution générale qui s'adaptera aux différentes sources d'informations favorisées par les différents enfants et qui favorise leur développement. Cela nous amène à poser les remarques suivantes :

- Combien existe-t-il de sources d'informations (vidéos) reconnues pourvues de valeurs didactiques et enrichissantes pour ces enfants ?
- À quel point l'analyse multimodale des vidéos sera généralisable aux différentes sources d'informations (vidéos) satisfaisant la curiosité des enfants ?
- Existe-t-il une base de données d'évaluation clinique spécifiant des vidéos d'intérêt pour chaque enfant ?

Finalement, la vision globale de ce programme de recherche va au-delà des problématiques d'interprétation de modalités (émotions et vidéos) et d'optimisation d'un système. En guise de retombées positives, ce système de recommandations aidera les parents et éducateurs à mieux comprendre les intérêts et le développement linguistique de leurs enfants. De plus, les différentes informations collectées seront une contribution avec un impact élevé dans la communauté scientifique, car il n'existe actuellement aucune base de données sur l'évolution du développement du langage chez les enfants autistes. Cela permettra aux spécialistes de rechercher des relations causales sur le développement linguistique de l'enfant autiste par l'exploitation des données recueillies. À long terme, ce système contribuera à installer dans le milieu éducatif une vision du développement des enfants autistes adapté à leurs modes d'apprentissage spécifiques.

8.3 Travaux futurs

Afin d'atteindre l'objectif du programme de recherche, les travaux futurs doivent se concentrer sur l'analyse multimodale des vidéos afin d'augmenter les différentes caractéristiques qui permettront à l'engin de recommandations de bonifier ses capacités à interpréter le contenu des vidéos. Cela aura pour conséquence d'améliorer la qualité des recommandations ainsi que l'interprétation des informations constituant l'historique des enfants. De plus, puisque ces enfants ne manifesteront pas leur appréciation à l'aide d'un mécanisme de notation en ligne, il serait possible d'envisager l'utilisation d'appareils électroniques qui permettraient de traduire leurs émotions afin de déduire l'appréciation ou non de la séquence vidéo. Cependant, cette avenue pourrait être considérée comme trop invasive, voire agressive pour ce cas d'utilisation. Nous pourrions alors proposer d'investiguer au sein de la communauté des praticiens de l'autisme s'il existe une base de données spécifiant des vidéos d'intérêt pour chaque enfant conçue à partir d'évaluations cliniques.

Concernant la carte sémantique, il a été démontré que celle-ci permet, à la suite de la mise en évidence des thématiques décrivant les centres d'intérêt, de décrire le contenu des vidéos

regardées. Pour un tuteur, cette représentation lui permettrait donc d'interpréter les vidéos que l'enfant a regardées. Outre les différentes améliorations concernant cette carte préalablement discutées au à la section 7.2, il serait approprié d'y ajouter les informations concernant les vidéos qui auraient été classifiées comme non pertinentes et d'ajuster la représentation des liens à l'aide d'un gradient de couleur afin de représenter le pourcentage d'appartenance du centre d'intérêt aux différents sujets.

Finalement, puisque nous sommes dans une période où l'apprentissage profond domine la scène de l'apprentissage machine, il serait intéressant de comparer les résultats obtenus de notre modélisation des sous-titres à une modélisation à l'aide d'un modèle comme BERTopic.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Figure 1.2 Jay Alammar 2022, dans Talk Topic Modeling with BERTopic - Talking Language AI Ep#1, sur le site co:here. Image téléchargée le 22 mars 2023. https://txt.cohere.ai/topic-modeling-with-bertopic/

Figure 1.3 Jay Alammar 2022, dans Talk Topic Modeling with BERTopic - Talking Language AI Ep#1, sur le site co:here. Image téléchargée le 22 mars 2023. https://txt.cohere.ai/topic-modeling-with-bertopic/

Figure 1.4 Jay Alammar 2022, dans Talk Topic Modeling with BERTopic - Talking Language AI Ep#1, sur le site co:here. Image téléchargée le 22 mars 2023. https://txt.cohere.ai/topic-modeling-with-bertopic/

BIBLIOGRAPHIE

Agrawal, Amritanshu, Wei Fu et Tim Menzies. 2018. « What is wrong with topic modeling? And how to fix it using search-based software engineering ». *Information and Software Technology,* vol. 98, p. 74-88.

Arun, R., V. Suresh, C. Madhavan et M. Murty. 2010a. *On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations*. 391-402 p.

Arun, R., V. Suresh, C. E. Veni Madhavan et M. N. Narasimha Murthy. 2010b. « On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations ». In. (Berlin, Heidelberg). 10.1007/978-3-642-13657-3_43. p. 391-402. Coll. « Advances in Knowledge Discovery and Data Mining »: Springer Berlin Heidelberg.

Asuncion, Arthur, Max Welling, Padhraic Smyth et Yee Whye Teh. 2012. « On smoothing and inference for topic models ». *arXiv preprint arXiv:1205.2662*.

Bard, Gregory V. 2007. « Spelling-error tolerant, order-independent pass-phrases via the damerau-levenshtein string-edit distance metric ». In *Proceedings of the fifth Australasian symposium on ACSW frontiers - Volume 68*. (Ballarat, Australia), p. 117–124. Australian Computer Society, Inc.

Basu, Subhasree, Yi Yu, Vivek K Singh et Roger Zimmermann. 2016. « Videopedia: lecture video recommendation for educational blogs using topic modeling ». In *International Conference on Multimedia Modeling*. p. 238-250. Springer.

Bennett, James, et Stan Lanning. 2007. « The netflix prize ». In *Proceedings of KDD cup and workshop*. Vol. 2007, p. 35. New York.

Biel, Joan-Isaac, et Daniel Gatica-Perez. 2014. « Mining crowdsourced first impressions in online social video ». *IEEE Transactions on Multimedia,* vol. 16, nº 7, p. 2062-2074.

Binkley, David, Daniel Heinz, Dawn Lawrie et Justin Overfelt. 2016. « Source code analysis with LDA ». *Journal of Software: Evolution and Process,* vol. 28, nº 10, p. 893-920.

Blei, David M, Andrew Y Ng et Michael I Jordan. 2003. « Latent dirichlet allocation ». *the Journal of machine Learning research,* vol. 3, p. 993-1022.

Boyd-Graber, Jordan, Yuening Hu et David Mimno. 2017. « Applications of topic models ». *Foundations and Trends® in Information Retrieval,* vol. 11, nº 2-3, p. 143-296.

Brunet, Jean-Philippe, Pablo Tamayo, Todd Golub et Jill Mesirov. 2004. « Metagenes and molecular pattern discovery using matrix factorization ». *Proceedings of the National Academy of Sciences of the United States of America,* vol. 101, p. 4164-9.

Burke, Robin. 2002. « Hybrid Recommender Systems: Survey and Experiments ». *User Modeling and User-Adapted Interaction,* vol. 12.

Chang, Jonathan, Sean Gerrish, Chong Wang, Jordan Boyd-Graber et David Blei. 2009. « Reading tea leaves: How humans interpret topic models ». *Advances in neural information processing systems,* vol. 22.

Cheng, Xueqi, Xiaohui Yan, Yanyan Lan et Jiafeng Guo. 2014. « Btm: Topic modeling over short texts ». *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, nº 12, p. 2928-2941.

Denny, Matthew J., et Arthur Spirling. 2018. « Text Preprocessing For Unsupervised Learning: Why It Matters, When It Misleads, And What To Do About It ». *Political Analysis,* vol. 26, nº 2, p. 168-189.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee et Kristina Toutanova. 2018. « Bert: Pre-training of deep bidirectional transformers for language understanding ». *arXiv preprint arXiv:1810.04805*.

Dieng, Adji B, Francisco JR Ruiz et David M Blei. 2020. « Topic modeling in embedding spaces ». *Transactions of the Association for Computational Linguistics,* vol. 8, p. 439-453.

DiMaggio, Paul, Manish Nag et David Blei. 2013. « Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of U.S. government arts funding ». *Poetics,* vol. 41, nº 6, p. 570-606.

Elgesem, Dag, Ingo Feinerer et Lubos Steskal. 2016. « Bloggers' responses to the Snowden affair: Combining automated and manual methods in the analysis of news blogging ». *Computer Supported Cooperative Work (CSCW),* vol. 25, p. 167-191.

Evans, Michael S. 2014. « A computational approach to qualitative analysis in large textual datasets ». *PloS one,* vol. 9, nº 2, p. e87908.

Goldberg, David, David Nichols, Brian M Oki et Douglas Terry. 1992. « Using collaborative filtering to weave an information tapestry ». *Communications of the ACM,* vol. 35, nº 12, p. 61-70.

Goldberg, Ken, Theresa Roeder, Dhruv Gupta et Chris Perkins. 2001. « Eigentaste: A constant time collaborative filtering algorithm ». *information retrieval,* vol. 4, p. 133-151.

Gomaa, Wael H, et Aly A Fahmy. 2013. « A survey of text similarity approaches ». *international journal of Computer Applications,* vol. 68, nº 13, p. 13-18.

Greene, Derek, et Pádraig Cunningham. 2006. « Practical solutions to the problem of diagonal dominance in kernel document clustering ». In *Proceedings of the 23rd international conference on Machine learning*. p. 377-384.

Greene, Derek, Derek O'Callaghan et Pádraig Cunningham. 2014. « How Many Topics? Stability Analysis for Topic Models ». In. (Berlin, Heidelberg). 10.1007/978-3-662-44848-9_32. p. 498-513. Coll. « Machine Learning and Knowledge Discovery in Databases »: Springer Berlin Heidelberg.

Griffiths, Thomas L, et Mark Steyvers. 2004. « Finding scientific topics ». *Proceedings of the National academy of Sciences,* vol. 101, nº suppl 1, p. 5228-5235.

Grootendorst, Maarten. 2022. « BERTopic: Neural topic modeling with a class-based TF-IDF procedure ». *arXiv preprint arXiv:2203.05794*.

Guo, Lei, Chris J Vargo, Zixuan Pan, Weicong Ding et Prakash Ishwar. 2016. « Big social data analytics in journalism and mass communication: Comparing dictionary-based text analysis and unsupervised topic modeling ». *Journalism & Mass Communication Quarterly,* vol. 93, nº 2, p. 332-359.

Heffler, Karen Frankel, Danielle M Sienko, Keshab Subedi, Kathleen A McCann et David S Bennett. 2020. « Association of Early-Life Social and Digital Media Experiences With Development of Autism Spectrum Disorder–Like Symptoms ». *JAMA pediatrics*.

Hill, Will, Larry Stead, Mark Rosenstein et George Furnas. 1995. « Recommending and evaluating choices in a virtual community of use ». In *Proceedings of the SIGCHI conference on Human factors in computing systems*. p. 194-201.

Hughes, Michael, Dae Il Kim et Erik Sudderth. 2015. « Reliable and scalable variational inference for the hierarchical dirichlet process ». In *Artificial Intelligence and Statistics*. p. 370-378. PMLR.

Jaccard, Paul. 1901. « Étude comparative de la distribution florale dans une portion des Alpes et des Jura ». *Bull Soc Vaudoise Sci Nat,* vol. 37, p. 547-579.

Jacobi, Carina, Wouter Van Atteveldt et Kasper Welbers. 2018. « Quantitative analysis of large amounts of journalistic texts using topic modelling ». In *Rethinking Research Methods in an Age of Digital Journalism*. p. 89-106. Routledge.

Jacques, Claudine, Valérie Courchesne, Andrée-Anne S Meilleur, Suzanne Mineau, Stéphanie Ferguson, Dominique Cousineau, Aurélie Labbe, Michelle Dawson et Laurent Mottron. 2018. « What interests young autistic children? An exploratory study of object exploration and repetitive behavior ». *PloS one,* vol. 13, nº 12, p. e0209251.

Jaro, Matthew A. 1989. « Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida ». *Journal of the American Statistical Association,* vol. 84, nº 406, p. 414-420.

Jones, Thomas, William Doane et Maintainer Thomas Jones. 2016. « Package 'textmineR' ». *Functions for Text Mining and Topic Modeling*.

Jones, Tommy. 2019. « A Coefficient of Determination for Probabilistic Topic Models ». *arXiv preprint arXiv:1911.11061*.

Kaldy, Zsuzsa, Catherine Kraper, Alice S Carter et Erik Blaser. 2011. « Toddlers with autism spectrum disorder are more successful at visual search than typically developing toddlers ». *Developmental science,* vol. 14, nº 5, p. 980-988.

Kientz, Julie A, Matthew S Goodwin, Gillian R Hayes et Gregory D Abowd. 2013. « Interactive technologies for autism ». *Synthesis Lectures on Assistive, Rehabilitative, and Health-Preserving Technologies,* vol. 2, nº 2, p. 1-177.

Kissine, Mikhail, Xavier Luffin, Fethia Aiad, Rym Bourourou, Gaétane Deliens et Naoufel Gaddour. 2019. « Noncolloquial Arabic in Tunisian children with autism spectrum disorder: A possible instance of language acquisition in a noninteractive context ». *Language learning,* vol. 69, nº 1, p. 44-70.

Levy, Karen EC, et Michael Franklin. 2014. « Driving regulation: Using topic models to examine political contention in the US trucking industry ». *Social Science Computer Review,* vol. 32, nº 2, p. 182-194.

Lin, Tianyi, Wentao Tian, Qiaozhu Mei et Hong Cheng. 2014. « The dual-sparse topic model: mining focused topics and focused terms in short text ». In *Proceedings of the 23rd international conference on World wide web*. p. 539-550.

Lopez-Herrejon, Roberto E, Oishi Poddar, Gerardo Herrera et Javier Sevilla. 2020. « Customization support in computer-based technologies for autism: A systematic mapping study ». *International Journal of Human–Computer Interaction*, p. 1-18.

Maier, Daniel, A. Waldherr, P. Miltner, G. Wiedemann, A. Niekler, A. Keinert, B. Pfetsch, G. Heyer, U. Reber, T. Häussler, H. Schmid-Petri et S. Adam. 2018. « Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology ». *Communication Methods and Measures,* vol. 12, nº 2-3, p. 93-118.

Mantyla, Mika V., Maelick Claes et Umar Farooq. 2018. « Measuring LDA topic stability from clusters of replicated runs ». In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. (Oulu, Finland), p. Article 49. Association for Computing Machinery. < https://doi.org/10.1145/3239235.3267435 >.

Marshall, Emily A. 2013. « Defining population problems: Using topic models for cross-national comparison of disciplinary development ». *Poetics,* vol. 41, nº 6, p. 701-724.

Mehrotra, Rishabh, Scott Sanner, Wray Buntine et Lexing Xie. 2013. « Improving lda topic models for microblogs via tweet pooling and automatic labeling ». In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. p. 889-892.

Mei, Tao, Bo Yang, Xian-Sheng Hua et Shipeng Li. 2011. « Contextual Video Recommendation by Multimodal Relevance and User Feedback ». *ACM Trans. Inf. Syst.,* vol. 29, nº 2, p. Article 10.

Mimno, David, Hanna Wallach, Edmund Talley, Miriam Leenders et Andrew McCallum. 2011. « Optimizing Semantic Coherence in Topic Models ». In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. (Edinburgh, Scotland, UK., jul). mimno-etal-2011-optimizing. p. 262-272. Association for Computational Linguistics. < https://www.aclweb.org/anthology/D11-1024 >.

Minka, Thomas P, et John Lafferty. 2012. « Expectation-propogation for the generative aspect model ». *arXiv preprint arXiv:1301.0588*.

Newman, David, Chaitanya Chemudugunta, Padhraic Smyth et Mark Steyvers. 2006. « Analyzing entities and topics in news articles using statistical topic models ». In *Intelligence and Security Informatics: IEEE International Conference on Intelligence and Security Informatics, ISI 2006, San Diego, CA, USA, May 23-24, 2006. Proceedings 4*. p. 93-104. Springer.

Newman, David, Jey Han Lau, Karl Grieser et Timothy Baldwin. 2010. « Automatic evaluation of topic coherence ». In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*. p. 100-108.

Nguyen, Dong, A. Seza Doğruöz, Carolyn P. Rosé et Franciska de Jong. 2016. « Computational Sociolinguistics: A Survey ». *Computational Linguistics,* vol. 42, nº 3, p. 537-593.

Ostrolenk, Alexia, Baudouin Forgeot d'Arc, Patricia Jelenic, Fabienne Samson et Laurent Mottron. 2017. « Hyperlexia: Systematic review, neurocognitive modelling, and outcome ». *Neuroscience & Biobehavioral Reviews,* vol. 79, p. 134-149.

Panichella, A., B. Dit, R. Oliveto, M. Di Penta, D. Poshynanyk et A. De Lucia. 2013. « How to effectively use topic models for software engineering tasks? An approach based on Genetic Algorithms ». In *2013 35th International Conference on Software Engineering (ICSE)*. (18-26 May 2013), p. 522-531.

Panichella, Annibale. 2021. « A Systematic Comparison of search-Based approaches for LDA hyperparameter tuning ». *Information and Software Technology,* vol. 130, p. 106411.

Pennington, Jeffrey, Richard Socher et Christopher Manning. 2014. « GloVe: Global Vectors for Word Representation ». In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (Doha, Qatar, oct). pennington-etal-2014-glove. p. 1532-1543. Association for Computational Linguistics. < https://www.aclweb.org/anthology/D14-1162
http://dx.doi.org/10.3115/v1/D14-1162 >.

Puschmann, Cornelius, et Tatjana Scheffler. 2016. « Topic modeling for media and communication research: A short primer ».

Qiang, Jipeng, Zhenyu Qian, Yun Li, Yunhao Yuan et Xindong Wu. 2020. « Short text topic modeling techniques, applications, and performance: a survey ». *IEEE Transactions on Knowledge and Data Engineering,* vol. 34, nº 3, p. 1427-1445.

Rajaraman, V. 2016. « Big data analytics ». *Resonance,* vol. 21, nº 8, p. 695-716.

Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom et John Riedl. 1994. « Grouplens: An open architecture for collaborative filtering of netnews ». In *Proceedings of the 1994 ACM conference on Computer supported cooperative work.* p. 175-186.

Rich, Elaine. 1979. « User modeling via stereotypes ». *Cognitive science,* vol. 3, nº 4, p. 329-354.

Rieger, Jonas, Jörg Rahnenführer et Carsten Jentsch. 2020. « Improving Latent Dirichlet Allocation: On Reliability of the Novel Method LDAPrototype ». In. (Cham).

10.1007/978-3-030-51310-8_11. p. 118-125. Coll. « Natural Language Processing and Information Systems »: Springer International Publishing.

Scrucca, Luca. 2013. « GA: a package for genetic algorithms in R ». *Journal of Statistical Software,* vol. 53, nº 1, p. 1-37.

Scrucca, Luca. 2016. « On some extensions to GA package: hybrid optimisation, parallelisation and islands evolution ». *arXiv preprint arXiv:1605.01931*.

Shardanand, Upendra, et Pattie Maes. 1995. « Social information filtering: Algorithms for automating "word of mouth" ». In *Proceedings of the SIGCHI conference on Human factors in computing systems*. p. 210-217.

Stevens, Keith, Philip Kegelmeyer, David Andrzejewski et David Buttler. 2012. « Exploring topic coherence over many models and many topics ». In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. p. 952-961.

Syed, S., et M. Spruit. 2018. « Exploring Symmetrical and Asymmetrical Dirichlet Priors for Latent Dirichlet Allocation ». *Int. J. Semantic Comput.,* vol. 12, p. 399-423.

Terragni, Silvia, Elisabetta Fersini, Bruno Giovanni Galuzzi, Pietro Tropeano et Antonio Candelieri. 2021. « Octis: comparing and optimizing topic models is simple! ». In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. p. 263-270.

Wallach, Hanna M, David M Mimno et Andrew McCallum. 2009. « Rethinking LDA: Why priors matter ». In *Advances in neural information processing systems*. p. 1973-1981.

Wallach, Hanna M, Iain Murray, Ruslan Salakhutdinov et David Mimno. 2009. « Evaluation methods for topic models ». In *Proceedings of the 26th annual international conference on machine learning*. p. 1105-1112.

Wang, Jiapeng, et Yihong Dong. 2020. « Measurement of text similarity: a survey ». *Information,* vol. 11, nº 9, p. 421.

Webber, William, Alistair Moffat et Justin Zobel. 2010. « A similarity measure for indefinite rankings ». *ACM Trans. Inf. Syst.,* vol. 28, nº 4, p. Article 20.

Yarnguy, Thanakorn, et Wanida Kanarkard. 2018. « Tuning latent Dirichlet allocation parameters using ant colony optimization ». *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),* vol. 10, nº 1-9, p. 21-24.

Zhao, Weizhong, James J Chen, Roger Perkins, Zhichao Liu, Weigong Ge, Yijun Ding et Wen Zou. 2015. « A heuristic approach to determine an appropriate number of topics in topic modeling ». In *BMC bioinformatics*. Vol. 16, p. 1-10. Springer.

Zhu, Qiusha, Mei-Ling Shyu et Haohong Wang. 2013. « Videotopic: Content-based video recommendation using a topic model ». In *2013 IEEE International Symposium on Multimedia*. p. 219-222. IEEE.