Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

Comparaison de Topic Models pour l'extraction de lexique et la classification de courts textes

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Arthur BOUZARD

Directeur de mémoire :

Damien Nouvel

Encadrant :

Dominique Casanova

Année universitaire 2021/2022

Table des matières

Liste des figures

Liste des tableaux

AVANT-PROPOS

Remerciements

Je tiens à remercier l'ensemble des professeurs du Master Traitement Automatique des Langues de l'INALCO dont la qualité de l'enseignement m'a permis d'acquérir les connaissances mais aussi l'autonomie nécessaire afin de mener à bien cette étude. Je souhaite aussi remercier tous les élèves de la promotion 2020-2021 dont la bienveillance et le sérieux m'ont permis de progresser au cours de ces années d'études. Je remercie surtout mon professeur, monsieur Damien Nouvel, également directeur du présent mémoire pour son aide et ses conseils.
Je tiens également à remercier l'équipe scientifique du Français des affaires et plus particulièrement monsieur Dominique Casanova, mon tuteur de stage, pour sa bienveillance et sa pédagogie tout au long de mon stage ainsi que monsieur Julien Mouchnino pour ses nombreux enseignements. Je remercie enfin toutes les équipes du Français des affaires pour leur accompagnement et leur gentillesse.

Résumé

Cette étude fait suite à celles des précédents stagiaires du Français des affaires, qui s'articule autour de la construction d'un système de notation automatique de copies de Test d'Evaluation de Français, un examen à forts enjeux permettant l'obtention du droit de résidence ou de naturalisation en France ou au Canada. Ce système se basant sur des caractéristiques extraites des copies de l'examen, nous nous penchons sur l'automatisation de l'extraction d'une caractéristique qui était produite en partie manuellement. Cette dernière se repose sur la confection manuelle laborieuse de lexiques thématiques.

De nombreuses études utilisent le *topic modelling* pour l'alignement de lexiques bilingues [Liu et al., 2015] voire même l'extraction de lexiques par niveau de langue [Megumi et al., 2019], nous nous attarderons ici sur cette technique afin d'entretenir le système de notation automatique. L'étude montre que ces modèles sont capables de générer des lexiques pertinents à condition qu'une méthode adéquate aux données soit choisie avec un paramétrage soigneusement réglé. Ce travail est en cours d'intégration dans le système de notation automatique.

Mots-clés :

français langue étrangère, apprentissage automatique, topic modelling, lexique thématique, plongements lexicaux

Introduction

Présentation générale

On attend des organismes concentrant leurs activités sur la conception de tests en langue étrangère et le décernement de certifications d'aptitudes, de fournir des résultats aussi révélateurs que possible du niveau de langue des individus. Ces organismes entretenant d'étroites relations avec les états, ces derniers imposent un standard de qualité afin d'utiliser le résultat de ces tests dans le but de faciliter certaines procédures administratives. Le « Français des affaires », est un de ces organismes qui a pour rôle de promouvoir un français utile et professionnel via son Test d'Evaluation de Français et ses Diplômes de Français Professionnel. L'évolution des supports de test ayant permis la numérisation quasiment complète des productions des candidats, le Français des affaires y a entrevu l'opportunité d'améliorer la qualité et de réduire les coûts de leur évaluation, grâce au traitement de la masse de données textuelles récoltées. Ce dernier a pour cela élaboré un système de notation automatique pour l'épreuve d'expression écrite du test, présentant un modèle d'apprentissage automatique supervisé basé sur l'extraction de caractéristiques à partir de données textuelles. Ce système a déjà montré de bons résultats sur des corpus moins conséquents que ceux que l'on a à disposition aujourd'hui. Cependant, l'extraction de certaines de ces caractéristiques n'étant pas encore automatisée, elle présente aussi un coût humain, qui peut s'amplifier à l'avenir.

Ce travail a donc pour but d'automatiser l'extraction d'une caractéristique nécessitant la confection d'un lexique thématique pour chaque sujet. Le corpus n'étant composé que des productions des candidats et de la note qui leur a été attribué, il est nécéssaire de se tourner vers des approches non-supervisées, bien moins courantes pour effectuer cette tâche. Nous avons néanmoins pu identifier le *topic modelling* comme étant la méthode la plus apte à remplir nos objectifs, de par la sortie de ses modèles qui coïncide avec les résultats que nous désirons obtenir.
De plus, nous pensons que cela pourrait aussi nous permettre de regrouper automatiquement les sujets d'expression écrite trop similaires afin de réduire le nombre de lexiques à alimenter mais également de juger de la pertinence de proposer plusieurs sujets qui pourraient donner des productions trop semblables.

Première partie

Contexte général

CHAPITRE 1

CADRE DE L'ÉTUDE

Sommaire

1.1 Introduction

Nous présenterons le contexte général de cette étude et détaillerons les activités du Français des affaires dans son élaboration du TEF, puis nous aborderons un peu plus en détail ce système de notation automatique et ses enjeux pour enfin nous attarder sur notre choix d'utiliser le *topic modelling* afin d'améliorer ce système.

1.2 Contexte

1.2.1 Le Français des affaires, les tests de français et le département scientifique

Cette étude s'inscrit dans le cadre d'un stage de fin d'études de maîtrise en Traitement Automatique des Langues. Ce stage s'est déroulé au sein du département de la direction scientifique du Français des affaires, une activité de la Direction de l'attractivité internationale de la Chambre du Commerce de l'Industrie (CCI) de la région Paris Île-de-France. Ce stage s'est concentré sur le système de notation automatique du Test d'Evaluation de Français (TEF), développé depuis quelques années par les présents et anciens membres de l'équipe scientifique du Français des affaires, ainsi que par de nombreux stagiaires qui ont travaillé à améliorer progressivement ce système.

Fondé en 1958, le Français des affaires est un service de la CCI qui a pour mission principale de concevoir et de valider des tests de français général ainsi que de français professionnel tout en faisant profiter de son expertise pédagogique aux enseignants et aux entreprises.

Pour ses tests, on distingue deux catégories différentes. Tout d'abord, les Diplômes de Français Professionnel (DFP), qui se déclinent entre plusieurs domaines d'activité parmi lesquels figurent l'hôtellerie-tourisme-restauration, la santé, les affaires et les relations internationales. Puis, nous avons le Test d'Evaluation de Français mis en place depuis 1998 et dont l'attractivité réside sur le fait qu'il soit nécéssaire à l'obtention de la nationalité ou au droit de résidence en France ou au Canada. La passation de ces tests s'effectue dans des centres disséminés partout dans le monde. On compte 118 pays où il est possible de passer toutes les épreuves du TEF. Il existe plusieurs versions différentes du TEF dont les épreuves et la finalité différent :

— Le **TEF** générique. Il se compose de 5 épreuves et est notamment passé par les étudiants souhaitant poursuivre leurs études dans une université francophone.

— Le **TEF Intégration - Résidence - Naturalisation (IRN)**. Il évalue un candidat entre le niveau A1 et le niveau B1. Il a été élaboré à la demande du ministère de l'Intérieur pour proposer une évaluation aux différentes étapes du parcours d'intégration.

— Le **TEF Canada** nécéssaire pour faire valider une demande d'immigration ou de citoyenneté au Canada, hors Québec.

— Le **TEF pour l'accès au Québec** ou TEFAQ, comme pour le TEF Canada mais spécifique au Québec.

Le TEF Canada se compose de 4 épreuves différentes. Une compréhension écrite au format « questionnaire à choix multiples » (QCM), ainsi qu'une épreuve d'expression écrite. L'expression écrite présente deux tâches bien différentes, une première tâche (section A) présente le début d'un article insolite, le but étant de le terminer. La seconde tâche (section B) demande au candidat d'argumenter une courte affirmation en nuançant son propos. Cette seconde tâche était utilisée, jusqu'en décembre 2021, comme unique tâche d'expression écrite pour les versions du test qui ont précédé le TEF IRN (à savoir le TEF Résident et le TEF Naturalisation).

Les épreuves QCM sont évaluées automatiquement par ordinateur, les productions de l'épreuve d'expression écrite sont évaluées par deux spécialistes de français langue étrangère formés à la correction du TEF et enfin l'épreuve d'expression orale est évaluée par l'examinateur ayant servi d'interlocuteur au candidat ainsi que par un examinateur extérieur se basant sur l'enregistrement audio de l'épreuve.

C'est dans ce contexte qu'évolue l'équipe du développement scientifique qui a pour objectif de veiller à la qualité métrique des outils d'évaluation, à l'intégrité des résultats et à l'intégration des nouvelles technologies au service de l'évaluation. Elle s'occupe de diverses activités parmi lesquelles on compte le renforcement de la qualité métrique des tests et certifications, la sécurisation des conditions de passation, l'optimisation des ressources employées pour l'évaluation, l'innovation et enfin la communication autour des moyens mis en oeuvre pour remplir ses objectifs. L'équipe s'est donc progressivement spécialisée dans le développement d'outils visant à assurer l'intégrité des résultats et des notes délivrées. On note notamment le développement récent d'un système anti-fraude permettant d'analyser les anomalies au sein des formulaires QCM ainsi qu'une analyse en profondeur des données fournies par

les centres de passation du TEF à travers le monde en plus de la mise en place d'une détection automatique de plagiat et de frappes incohérentes pour l'épreuve d'expression écrite. L'autre préoccupation de l'équipe scientifique se trouve être son projet de notation automatique centré sur l'épreuve d'expression écrite du TEF que nous allons aborder plus en détails.

1.2.2 L'épreuve d'expression écrite du TEF

L'épreuve est limitée à une durée d'une heure. Le candidat doit répondre à deux sujets sélectionnés au hasard parmi tous les sujets disponibles dans la base de données et parmi les sujets en service. Un sujet est sélectionné parmi ceux de la section A, et un autre parmi ceux de la section B. Les sujets de la section A se présentent comme des débuts d'articles de journaux, qui ont la particularité de présenter un contexte insolite (voir exemple ci-dessous). Une consigne supplémentaire indique alors au candidat qu'il doit écrire la fin de l'article qu'il a eu pour sujet en au moins 80 mots. Les sujets de la section B présentent quant à eux une courte affirmation sur un thème comme l'éducation, la politique, la technologie... Une consigne supplémentaire indique un contexte au candidat (par exemple que cette phrase apparaît dans un journal) et lui demande de présenter des arguments en faveur ou contre cette affirmation en au moins 200 mots. A la manière d'un baccalauréat de français, on peut comparer la premère tâche à l'écrit d'invention et la seconde à une dissertation en version miniature. Donnons un expemple de sujet pour chacune des sections (le sujet n'étant pas authentique dans un souci de confidentialité) :

— **A** : Voici le début d'un article de journal :
« Safari à Paris. Un passant poursuivi par une meute d'éléphants au centre de la capitale française, des magasins saccagés et des voitures détruites, les autorités réagissent... » Ecrivez la suite. *80 mots*

— **B** : Vous avez lu l'affirmation suivante dans le journal :
« Les robots nous volent notre travail » .
Ecrivez une lettre au rédacteur de l'article pour exprimer votre point de vue en développant au moins 3 arguments. *200 mots*

Le format épistolaire n'est pas obligatoire et n'est pas pris en compte dans l'évaluation de la section B, de même pour le nombre de mots qui n'est pas pénalisé s'il est inférieur au nombre indiqué. D'ailleurs, en ce qui concerne l'évaluation, chacun des deux évaluateurs ne note pas directement la copie mais doit remplir une grille conçue par l'équipe pédagogique, en attribuant à chaque critère d'évaluation une note, suivant le système de notation du Cadre Européen commun de Référence pour les langues (CECR), c'est à dire entre <A1 et C2 (voir [CECR, 2022]). Les critères sont les suivants :

— Capacité à transmettre des informations
— Capacité à argumenter
— Syntaxe
— Lexique
— Cohérence et cohésion

On distingue les critères pragmatiques qui relèvent de capacités extra-linguistiques (les deux premiers), des critères purement linguistiques. Chaque critère renvoie à une échelle descriptive d'évaluation, où un descripteur précise, pour chaque niveau CECR, les observables-types attendus pour le critère considéré. Un score est calculé à partir du niveau attribué à chaque critère et la moyenne des scores des évaluateurs est reconvertie en niveau CECR. En outre, un arbitrage effectué par un responsable pédagogique du Français des affaires a lieu si une copie est appréciée très différemment par chacun des deux évaluateurs.

Le TEF faisant office de certification dans des contextes à enjeux élevés, le nombre de candidats au TEF et donc le nombre d'arbitrages augmentent chaque année. L'évaluation ainsi que les arbitrages de l'épreuve d'expression écrite ayant un coût humain assez important, et également dans le but d'améliorer la fiabilité des notes délivrées, l'équipe du développement scientifique s'est alors lancée dans un projet de notation automatique.

1.3 La notation automatique

L'examen du TEF ayant été complètement numérisé, l'accès simple et immédiat à des données textuelles en grande quantité a conforté l'équipe scientifique dans son idée de développer un système de notation automatique. En outre, les systèmes de notation automatique nécéssitant des données annotés, au moins du niveau de langue du document, le Français des affaires constitue alors un terrain idéal pour le développement d'un tel système. Néanmoins, les systèmes de notation automatique de production écrite ne datent pas d'hier. Dès 1999, le système « Intelligent Essay Grader » [Foltz et al., 1999] permet d'assister un correcteur humain dans son évaluation de productions en anglais. La plupart de ces systèmes associent des techniques de traitement automatique du langage naturel à l'apprentissage automatique afin d'extraire d'une production des caractéristiques (*Feature Engineering*) qui sont utilisées comme données pour effectuer une tâche de classification ou régression automatique. Plus récemment, certaines études ont suggéré que l'utilisation de l'apprentissage profond pourrait permettre d'aussi bons résultats tout en se passant de cette tâche d'extraction de caractéristiques [Filho et al., 2020], mais d'un autre côté, d'autres études ont démontré que l'utilisation de l'apprentissage profond pour l'extraction de caractéristiques donnait des résultats similaires aux anciennes méthodes pour un coût matériel et temporel bien plus important [Mayfield and Black, 2020].

Le système développé au sein de l'équipe scientifique du Français des affaires repose quant à lui sur l'extraction de caractéristiques utilisées dans un classifieur automatique (forêts aléatoires ou régréssion logistique ordinale). Au fur et à mesure de son développement, de nombreuses caractéristiques ont été ajoutées aux données d'entraînement, dont une grande partie repose sur la description des critères linguistiques. Il est en effet, plus simple de compter les fautes de syntaxe ou d'orthographe, de relever la diversité lexicale et la compléxité grammaticale d'une phrase que d'évaluer automatiquement la pertinence sémantique des propos. Les caractéristiques pragmatiques sont donc beaucoup moins outillées en terme d'algorithmes et c'est sur quoi l'équipe souhaite s'étendre davantage afin d'améliorer son système. D'autant plus que jusqu'alors, la classification se faisait sur l'ensemble des caractéristiques de tous les critères, mais le Français des affaires a décidé de faire évoluer son système de sorte à identifier les critères se prêtant le mieux ou le moins bien à une notation automatique. En effet, il est légitime de se demander si, au lieu d'appuyer

un évaluateur dans son évaluation globale de la copie, on ne pourrait pas simplement le soulager des tâches pénibles qui constituent la forme de la copie pour qu'il puisse se concenter sur le fond. C'est pour cela qu'on vient à se concentrer sur l'extraction de caractéristiques pour les deux premiers critères. Le second critère notamment, manque cruellement de caractéristiques. On lui compte tout de même certaines caractéristiques comme le nombre de mots marqueurs d'opinion, ainsi que le nombre de mots marqueurs d'opinion distincts. On rajoutera le nombre de mots qui font partie du champs lexical du sujet. C'est sur cette dernière caractéristique, étudiée pour le descripteur « Adéquation du texte à la tâche » que nous allons nous attarder.

Cette dernière indique le nombre de mots de la copie appartenant au champs lexical du sujet qui a été défini de deux façons. Une première fois par la constitution manuel d'un lexique pour les quelques sujets représentés dans les copies qui étaient disponibles à l'époque. Puis, dans un second temps ce lexique a été généré à partir des plongements lexicaux des mots du sujet. Les mots du modèle de plongements lexicaux word2vec ayant une similarité supérieure à un seuil de 0.80 avec un des mots de l'énoncé du sujet sont sélectionnés en tant que mot du lexique pour ce sujet. Néanmoins, cette méthode a ses limites. Tout d'abord elle génère du bruit, car de nombreux mots sélectionnés ne peuvent pas être considérés comme faisant partie du champs lexical du sujet, ensuite, pour certains sujets, aucun mot n'est sélectionné et abaisser le seuil génère également du bruit. Enfin, les modèles words2vec entraînés sur un corpus conséquent commencent à vieillir, et l'usage du vocabulaire actuel ne correspond pas au vocabulaire présent dans le modèle word2vec (par exemple le mot « influenceur » est très proche du mot « skyblogueur »).

Nous nous sommes alors demandés dans quelle mesure il était possible de constituer automatiquement un lexique pour chaque sujet relevant de son champs lexical. Les sujets d'expression écrite étant régulièrement renouvelés, nous nous somme également demandé comment il serait possible d'attribuer de manière suffisament fiable un lexique à un nouveau sujet sans avoir à appliquer une longue chaîne de traitement à chaque fois qu'un nouveau sujet est mis en service. C'est dans ce but que nous nous sommes penchés sur le *topic modelling*, une technique bien précise d'apprentissage non-supervisé, pour parvenir à nos fins.

1.4 Objectifs

« Le *topic modelling* est une technique en apprentissage automatique permettant de révéler les sujets latents dans un corpus de documents »
Derrière cette définition aguicheuse se cache un principe assez simple. Grâce aux statistiques, le *topic modelling* fait automatiquement le lien entre des documents et des thèmes (*topics* en anglais) qui se trouvent être un ensemble de mots liés sémantiquement. Dans le mesure où les sujets (que nous appelerons thèmes ou topics pour faire la différence avec les sujets d'expression écrite) de nos documents ne nous sont absolument pas inconnus vu qu'ils répondent à un sujet bien précis, il nous semble intéressant de voir s'il est possible d'obtenir un modèle présentant un nombre de thèmes raisonnable et dont le lexique s'accorde sémantiquement avec celui d'un certain nombre de sujets.

Précisons néanmoins que le *topic modelling* n'étant plus une technique d'apprentissage automatique très populaire, les ressources en ligne restent limitées. De plus, c'est une technique qui évolue plutôt vite, surtout depuis les percées récentes dans le domaine du traitement automatique du langage naturel. Tout cela

provoque l'obsolescence de certains programmes implémentant un algorithme de *topic modelling*, et limitant ainsi les choix possibles à expérimenter pour atteindre nos objectifs. À une implémentation laborieuse s'ajoute les coûts matériels et temporels assez importants dès qu'il s'agit d'optimiser un modèle de *topic modelling*, et l'incompatibilité de certaines librairies Python disponibles avec un environnement Google Colaboratory n'aidant en rien, nous nous sommes limités à expérimenter un nombre restreint d'algorithmes. Il faut aussi prendre en compte le fait que contrairement à la plupart des modèles d'apprentissage automatique, l'évaluation d'un modèle de *topic modelling* se fait principalement a priori grâce à l'appréciation humaine, les métriques ne permettant que d'aiguiller les choix sur les modèles à justement évaluer manuellement. Nous tenterons donc ici d'expérimenter quelques algorithmes de *topic modelling*, en misant sur la diversité des techniques existantes afin d'être en mesure de sélectionner le plus apte à former les lexiques correspondants au champs lexical de nos sujets. Nous nous pencherons également sur la possibilité d'utiliser un modèle de *topic modelling* (un topic model) en tant que classifieur automatique afin de regrouper nos sujets par groupes pour limiter le nombre de lexiques à entretenir.

Pour résumer, notre tâche est double, c'est à dire à la fois la classification des sujets de l'épreuve d'expression écrite du test d'évaluation de français et l'extraction de mots-clés pertinents pour chacune des classes générées. Il nous a alors semblé judicieux de tirer parti du *topic modelling* qui semble s'adapter parallèlement à nos deux objectifs.

CHAPITRE 2

ÉTAT DE L'ART

Sommaire

2.1 Introduction

Malgré les avancées récentes dans le domaine du traitement automatique du langage naturel, la constitution de lexiques spécialisés dans un domaine ou sur une thématique se fait majoritairement manuellement, ce qui constitue une tâche laborieuse et qui induit un certain coût en terme de temps ou de ressources humaines. En outre, la constitution d'un lexique relatif à un concept donné pose un autre problème quant à la sémantique des mots qui le composent. D'un côté, la polysémie lexicale nous complique la tâche en estompant les limites thématiques d'une unité lexicale. Bien que chaque mot ait ses « conventions d'emploi » socialement admises [Morgan, 1978], le français étant une langue largement répandue dans le monde, ces conventions sont sujettes à varier selon la zone géographique dans laquelle l'on se trouve, et de la même façon, elles évoluent aussi dans le temps ainsi que différemment selon les communautés linguistiques et sociales qui les utilisent [Mercier et al., 2003]. D'un autre côté, certains emplois ne sauraient être soumis à des conventions, notamment les figures de styles, qui par phénomène d'inclusion, peuvent faire glisser le sens d'un mot qui ne pourra alors pas être considéré individuellement mais dépendamment des mots qui l'accompagnent et de leur contexte d'énonciation [Jover, 2003].

C'est dans ce contexte que se pose la question de l'élaboration d'un système automatisé pour l'extraction d'un lexique relatif à son sujet dans le cadre du projet de notation automatique de l'épreuve d'expression écrite du test d'évaluation de français. La tâche d'extraction lexicale pour évaluer les compétences pragmatiques d'un apprenant étranger s'apparente, en traitement automatique des langues, à ce que l'on appelle communément l'extraction de mots-clés. Nous nous pencherons ci-dessous sur le *topic modelling* dans la mesure où il s'agit d'une technique d'apprentissage machine non-supervisée, qui s'avère donc plus apte à traiter nos données qui ne sont pas annotées manuellement. Nous verrons donc les différentes techniques qui ont été développées en faisant le parallèle avec les avancés en traitement du langage naturel puis nous évoquerons l'évolution des méthodes d'évaluation de topic models.

2.2 Une ribambelle de topic models

Pour pouvoir comprendre comment le *topic modelling* pourrait nous permettre d'améliorer un système de notation automatique de productions écrites en langue française par des apprenants étrangers, il faut dans un premier temps définir précisément ce qu'est le *topic modelling* ainsi qu'évoquer ses différentes applications en se concentrant sur celles qui entrent en relation avec notre tâche d'extraction lexicale. En traitement automatique du langage naturel tout comme en Recherche d'Information, le *topic modelling* est une technique d'apprentissage automatique non-supervisé ayant pour objectif de modéliser un corpus textuel selon ses principaux thèmes. Elle permet donc d'extraire les thèmes d'un corpus qui sont appelés « topic » et qui se composent de l'ensemble des mots lui étant liés. Le modèle issu de l'apprentissage comportera alors pour chaque document, les thèmes qui lui ont été assignés et leur probabilité, et de la même façon, pour chaque thème, les mots du corpus et leur probabilité, qui lui ont été assignés (voir figure A.1).

Le premier algorithme de *topic modelling* a fait un long chemin, depuis les travaux publiés en 2003 [Blei et al., 2003]. Nommé l'allocation latente de Dirichlet (en anglais *Latent Dirichlet Allocation* ou LDA) et basé sur les travaux de l'indexage sémantique latent (LSI et pLSI) [Deerwester et al., 1990][Hofmann, 1999], il a permis de formaliser une nouvelle classe de modèles probabilistes en apprentissage automatique. Ce dernier a su montrer son utilité en traitement automatique des langues de part l'interprétabilité élevée de son modèle de sortie utilisable tel quel ou dont les variables de sortie peuvent êtres utilisées pour peupler d'autres données destinées à un apprentissage automatique. Néanmoins, ce modèle a présenté certains inconvénients qui ont poussé les chercheurs à élaborer des alternatives. Ses limites viennent du fait que les données ont besoin d'être prétraitées en filtrant les mots non signifiants via un processus de lemmatisation et le retrait des mots les plus courants (*stopwords*) ainsi que des documents trop courts. De plus, lors de l'apprentissage, l'algorithme utilise en entrée une représentation appelée « sac-de-mots » dont la dimension est égale au nombre de mots du vocabulaire du corpus. Pour finir, le nombre de thèmes générés doit être défini en tant que hyper-paramètre pour l'apprentissage bien que chaque thème généré implique une nouvelle série de calculs [Sontag and M Roy, 2009]. Tout cela sous-entend un temps de calcul conséquent et limité techniquement par la mémoire vive disponible. En outre, son utilisation est aussi restreinte sur les documents de petite taille car l'algorithme a besoin d'un volume suffisant de données pour correctement apprendre sur chaque document. Les différents algorithmes qui ont été conçus par la suite sont des variantes de la LDA qui s'efforcèrent de combler les manques de leur aînée. Le processus hiérarchique de Dirichlet (HDP) [Teh et al., 2004] permet d'optimiser automatiquement l'hyper-paramètre du nombre de thèmes à trouver dans le corpus, afin de ne pas avoir besoin de préciser le nombre de thèmes présents à l'intérieur de ce dernier. Le Dynamic Topic Model (DTM) [Blei and Lafferty, 2006] ou le Topics over Time (TOT) [Wang and McCallum, 2006], en associant une date à chaque document ou groupe de documents, permettent de générer des thèmes pour différentes périodes dans le temps et ainsi visualiser l'évolution temporelle de ces thèmes. Les algorithmes précédents ont pu être modernisés grâce à la « Online » LDA (OLDA) [Banerjee and Basu, 2007] qui a permis d'entraîner des modèles LDA et de les mettre à jour en temps réel pour traiter les flux de données d'internet. Le Biterm Topic Model (BTM) [Yan et al., 2013], qui suréchantillonne les données d'entrée et le Self-

Aggregating Topic Model (SATM) qui regroupe pluieurs documents similaires en un seul document, [Quan et al., 2015] ont permis tous deux d'améliorer la LDA sur des textes courts, répondant au besoin de traiter de nouveaux formats de documents suite à l'émergence et la popularisation des réseaux sociaux.

La plus grande avancée de ces dernières années dans le domaine a été, comme dans la plupart des tâches en traitement automatique du langage naturel, l'avènement des plongements lexicaux, plus fréquemment désignés par le terme anglais *word embeddings*. Basés sur des études menées quelques années auparavant [Collobert and Weston, 2008], ce n'est qu'en 2013 que leur utilisation a été généralisée, d'une part grâce à l'augmentation de la puissance de calculs de nos machines mais surtout grâce à la publication de l'algorithme « Word2Vec » optimisant le processus de calcul des plongements lexicaux [Mikolov et al., 2013a] [Mikolov et al., 2013b]. En remplaçant la représentation vectorielle classique du sac-de-mots, ils ont permis de réduire drastiquement la dimensionnalité des données tout en fournissant des variables plus précises qui prennent en compte la syntaxe et la sémantique de chaque mot en utilisant les mots qui l'entourent. Cependant, les différents plongements lexicaux issus de Word2Vec et des algorithmes (e.g GloVe, fastText) qui ont suivi peu après sont dits « non-contextuels » dans la mesure ou chaque token du vocabulaire ne possède qu'une seule représentation peu importe le nombre de sens qu'il peut avoir. La majorité des techniques qui ont suivi ont évidemment utilisé ces plongements lexicaux. La plupart des précédents algorithmes de topic model ont été améliorés en y implémentant les plongement lexicaux. La Latente Feature LDA (LF-LDA) [Nguyen et al., 2015] ou bien lda2vec [Moody, 2016] reprennent le tout premier topic model dans ce but. La LDA-DREx [Bicalho et al., 2017] reprend l'idée de la SATM en utilisant les plongements lexicaux pour calculer la similarité des mots du vocabulaire du corpus afin de remplacer au sein d'un document les mots trop rares ou l'étendre avec des mots similaires à ceux qu'il contient déjà. Le Dynamic Embedded Topic Model [Dieng et al., 2019] reprend le DTM en y incorporant les plongements lexicaux.

En parallèle, dès 2005, on a vu s'accroître les recherches visant à améliorer les temps d'entraînement de modèles d'apprentissage profond [Steinkrau et al., 2005] [Chellapilla et al., 2006], par l'utilisation de processeurs graphiques dédiés [Ghorpade et al., 2012]. Elles ont permis par la suite d'intégrer leur utilisation aux algorithmes de *topic modelling* [Mian et al., 2013]. Ce sont ces mêmes recherches qui ont abouti à une autre avancée majeure en traitement automatique du langage naturel : les modèles de langue à base de transformeurs [Vaswani et al., 2017], dont l'algorithme le plus connu se nomme BERT [Devlin et al., 2019]. Ce sont des modèles issus de réseaux de neurones profonds entraînés sur une quantité conséquente de données, qui permettent de générer des plongements lexicaux contextuels, et qui contrairement aux plongements lexicaux appris par word2vec, permettent de rendre compte dans une certaine mesure de la polysémie lexicale en générant une représentation vectorielle différente pour deux mots identiques si les segments de texte dans lesquels se trouve le mot ciblé sont différents. Ces plongements lexicaux de plus haute qualité ont apporté de nouvelles pistes aux chercheurs. Certains ont totalement mis de côté la LDA et ont développé des algorithmes basés sur des techniques de réduction de dimensions, inspirés de travaux antérieurs aux plongements lexicaux [Shahnaz et al., 2006], alors appliquées à ces derniers, cela a donné des résultats similaires voire meilleurs que ceux d'une LDA [Thompson and Mimno, 2020]. Ces derniers ont également été améliorés en y ajoutant d'autres techniques, dont le

TF-IDF [Grootendorst, 2022].

Comme nous venons de le voir, il existe pléthore d'algorithmes différents (voir [Churchill and Singh, 2022] pour une liste presque exhaustive), et pourtant nous avons seulement évoqué les principaux nous permettant de dessiner la dynamique d'évolution du *topic modelling*. Il existe plusieurs dizaines d'algorithmes que l'on peut répartir parmi deux classes. Il y a ceux basés sur une distribution de probabilités (Dirichlet, multinomiale, Poisson...) parmi lesquels on compte tous ceux ayant été évoqués avant les plongements lexicaux. La seconde classe contient les algorithmes basés sur la factorisation vectorielle mais vu que cette application ne s'est popularisée que plus récemment avec les modèles de langue, elle compte moins de membres. Cette classe inclut également les topic models à base de graphes [Cataldi et al., 2010] qui est une piste de recherche qui n'a pas beaucoup été développée. Avec cette multitude de techniques de *topic modelling*, il est logique de se demander quelle est la technique qui donne les meilleurs résultats. Cette question reste ouverte dans la mesure où chaque algorithme a un objectif différent, certains visent à modéliser les thèmes de textes courts, d'autres cherchent à modéliser l'évolution de thèmes dans le temps. Certains autres ne cherchent qu'à optimiser les temps de traitement de précédents algorithmes et ne s'attardent pas sur l'évaluation de leurs modèles. Cela veut dire que l'efficacité d'un algorithme de topic model dépend surtout des données utilisées et de l'objectif derrière l'utilisation d'un topic model en particulier plutôt qu'un autre. Ainsi, dans certains cas, on peut par exemple obtenir de meilleurs résultats avec un algorithme utilisant une représentation sac-de-mots qu'une représentation utilisant des plongements lexicaux.

En outre, de part la nature de la tâche, le *topic modelling* ne peut faire usage de métriques d'évaluation aussi robustes que celles définies pour d'autres types de modèle d'apprentissage automatique. Le *topic modelling* est un modèle génératif car il produit ces deux matrices en sortie : pour chaque document, la probabilité de chaque thème généré et pour chaque thème, la probabilité pour chaque mot lui étant assigné. On cherche principalement à évaluer qualitivement les matrices produites en jugeant de la cohérence sémantique des mots de chaque thème mais aussi de la cohérence des thèmes les uns par rapport aux autres à partir de la sémantique des mots qui les composent. Il est cependant impossible d'évaluer objectivement ces deux points car on ne connait aucune métrique qualitative prennant en compte le sens des mots. On ne peut qu'exprimer les résultats via des métriques quantitatives, la qualité globale doit être jugée grâce à l'appréciation humaine. Pour évaluer les topic models, on a donc, dès le départ, souvent utilisé un corpus de référence, pour comparer les performances des différents algorithmes. Le très célèbre corpus TwentyNewsgroups [Lang, 1995] est le corpus de référence par excellence car il fournit une vérité de terrain en étant constitué de documents regroupés par thèmes et labélisés manuellement. Ainsi, on peut mesurer la proportion de thèmes présents dans le corpus qui ont été générés par le modèle (le *topic recall*) tout en mesurant également la proportion des documents correctement assignés aux bons thèmes (la *topic precision*). Néanmoins, avoir recours au même corpus pour évaluer différents algorithmes pose problème dans la mesure où certains (BTM, SATM...) par exemple, ont été optimisés pour traiter des textes bien plus courts que les articles de journaux du TwentyNewsgroups ce qui ne permettait pas une évaluation objective de leur potentiel. Il fallait alors dans ce cas, afin d'évaluer un modèle sur un corpus de textes courts, l'annoter manuellement, mais aussi évaluer plusieurs autres modèles, qui pouvaient ne pas être adaptés à la tâche. L'objectif était donc de trouver des métriques utilisables sans corpus de référence, ainsi

diverses métriques non-supervisées ont été proposées à cet effet.

La Pointwise Mutual Information (PMI) [Church and Hanks, 1989] est utilisée pour mesurer la cohérence d'un thème selon les mots qui le composent. On calcule alors la PMI d'un nombre défini de mots du thème par paire sur toutes les combinaisons et l'on fait la moyenne. Plus la PMI est elevée, plus l'on considère que le thème est cohérent dans la mesure où les mots qui le composent ont, les uns avec les autres, un nombre élevé de cooccurrences dans le corpus. On peut ensuite utiliser la moyenne de la PMI de tous les thèmes générés.

Viendra ensuite la Topic Diversity (TD) qui est une mesure globale de la diversité des thèmes générés et qui correspond au pourcentage de mots uniques en ne prenant en compte qu'un nombre défini de mots pour chaque thème.

Il existe nombre de mesures que les chercheurs ont mis au point pour évaluer leur modèle, parfois réutilisées sous d'autres noms ou légèrement modifiées mais il n'y avait pas toujours de consensus quant auxquelles il fallait utiliser. La publication de frameworks centrés sur les techniques de *topic modelling* a permis cependant d'utiliser communément la PMI normalisée [Bouma, 2009] et la TD [Rehurek and Sojka, 2011]. Plus récemment, des chercheurs ont publié un framework centré sur l'évaluation du *topic modelling* en y implémentant toutes les métriques pertinentes proposées dans l'état de l'art jusqu'à nos jours. [Terragni et al., 2021].

2.3 Conclusion

Implémenter sur nos données la plupart des techniques présentes dans les travaux cités ne serait que peu pertinent. Nos documents étant très courts, il nous semble judicieux de tirer parti des travaux se concentrant sur ce type de documents. Il nous semble également intéressant d'expérimenter comparativement les algorithmes des deux classes détaillées plus haut, dans la mesure ou les algorithmes utilisant des plongements lexicaux issus de modèles de langues ont tendance à générer des thèmes bien plus explicites malgré la longueur des documents alors que les modèles de la première classe devraient mieux performer quantitativement, d'après nos attentes. Cela nous permettra également de comparer les deux représentations vectorielles sur nos données.

Deuxième partie

Expérimentations

CHAPITRE 3

CORPUS

Sommaire

3.1 Introduction

Commençons par rappeler les objectifs de notre étude. Il s'agit ici d'une tentative d'extraction automatique de termes afin d'alimenter un lexique pré-existant qui a été constitué manuellement. Ce lexique se divise en sous-lexiques, qui sont chacun attribués à un sujet de l'épreuve d'expression écrite du TEF. Chaque sous-lexique se compose de mots appartenant au champs lexical ou à la thématique des mots du sujet. Ce lexique a pour principal objectif de produire une variable réelle pour le descripteur « adéquation du texte à la tâche » afin d'alimenter le système de notation automatique. Néanmoins, ce lexique déjà construit est incomplet dans le mesure où il ne couvre que quelques-uns des sujets présents dans les copies vu que la numérisation complète du TEF s'est réalisée au cours de sa création. Aujourd'hui, le corpus d'entraînement pour le modèle de notation automatique étant bien plus conséquent et prenant en compte beaucoup plus de sujets d'expression écrite, il est nécessaire de se pencher vers une approche automatisée. De plus, il faut prendre en compte la réactualisation des sujets d'expression écrite. Pour préserver l'intégrité des résultats, de nouveaux sujets sont régulièrement proposés, et d'autres sont supprimés. Il faut aussi penser au fait que le contenu des sujets évolue dans le temps : de nos jours,

on peut trouver des sujets portant sur l'intelligence artificielle ou bien l'écologie. De surcroît, il faut surtout prendre en compte le fait que le vocabulaire évolue aussi très vite, ce qui peut rendre la mise à jour manuelle du lexique laborieuse et peu fiable dans la mesure où un seul humain ne pourra pas rendre compte de toutes les évolutions de la langue, les ignorant lui-même [Mercier et al., 2003].

Il faut rajouter à cela notre second objectif, qui est la classification des sujets. Le but est de les regrouper par thème, afin de visualiser quels thèmes sont trop ou trop peu représentés. Nous espérons aussi qu'en dégageant des thèmes précis, il nous sera alors possible de peupler manuellement plus aisément un lexique général sur chacun de ces thèmes plutôt que sur chacun des sujets existants.

Tout cela nous a orienté vers l'idée de développer un système basé sur le *topic modelling* afin d'entretenir voire d'améliorer l'algorithme de notation automatique. Nous allons donc dans ce chapitre, présenter les données à notre disposition qui se répartissent en deux corpus distincts. Le premier corpus correspond à tous les sujets d'expression écrite du TEF, et le second correspond à un échantillon de copies produites à partir de ces sujets.

3.2 Sujets

Le corpus constitué des sujets de l'épreuve d'expression écrite du TEF comprend 256 documents différents dont 201 sujets du TEFAQ et TEFCanada et 55 sujets de l'ancien TEF Naturalisation-Residence. On compte 101 sujets représentant la section A et 155 sujets pour la section B. (voir tableau 3.2)

3.2.1 Trois échantillons à définir pour l'entraînement

Le corpus en lui-même semble a priori inopérable. Les corpus des sujets n'étant que de courtes phrases de quelques tokens, il serait difficile d'obtenir des résultats satisfaisants en entraînant un topic model sur une si petite quantité de données. Il nous semble alors plus judicieux de ne pas utiliser directement les sujets d'expression écrite comme documents pour l'entraînement mais plutôt de les conserver en tant que corpus d'évaluation sur notre tâche de classification. Notre topic model sera donc entraîné sur les copies répondant aux sujets, que l'on possède en plus grand nombre.

Nous pouvons néanmoins faire quelques observations. Le corpus contient les sujets des deux sections de l'épreuve d'expression écrite, c'est à dire la section correspondant à la tâche de narration (section A) et la section correspondant à la tâche d'argumentation (section B). Les deux sections correspondant à des tâches bien différentes, cela peut suggérer l'utilisation d'un vocabulaire différent entre les sections. Il nous semblerait donc pertinent d'entraîner trois topic models différents. L'un sur la section A, un autre sur la section B et un dernier sur les deux sections (section AB). Cela pourrait influencer la qualité des thèmes générés. Les topic models présentent l'avantage de ne pas nécessiter autant de données que d'autres approches, contrairement l'apprentissage profond par exemple, afin d'initialiser un modèle, mais en réduisant la quantité de données d'entraînement pour les modèles à une unique section, cela pourrait aussi affecter négativement la qualité des thèmes générés. De plus, il faut garder en tête que nous ne pouvons pas tronquer la taille de notre échantillon comportant les deux sections pour qu'elle puisse correspondre à celle des échantillons à section unique car nous devons maximiser le nombre de sujets d'expression écrite représentés dans nos données d'entraînement et réduire le nombre de documents

par sujet pourrait rendre l'identification des thèmes récurrents du corpus plus compliquée. Nous savons néanmoins que pour les topic models, la qualité des données d'entraînement prime sur leur quantité pour ce qui est d'obtenir des topics interprétables. Nous voulons aussi voir quels résultats ces trois modèles donneront sur la classification des sujets, bien que nous pensons que les résultats de la classification dépenderont en grande partie de la qualité des thèmes générés. Il faut également prendre en compte la nature de nos documents. La production demandée pour la section A correspond à un texte narratif faisant suite à un énoncé insolite, décrit de manière détaillée, et cadré dans un contexte bien précis. Il est déjà relativement complexe pour un être-humain de classer les énoncés de la section A parmi des thèmes trop généraux ce qui peut être également le cas pour la machine. En revanche, les productions issues de la section B sont plutôt simples à classer pour un être-humain, car elles relèvent de courts énoncés correspondants à une affirmation relevant d'une idée plus générale à propos d'un thème plus ou moins précis comme la politique, l'éducation ou encore l'écologie. Nous pouvons donc nous attendre à de meilleurs résultats sur les topics modèles entraînés uniquement sur la section B.

3.2.2 Les mots-clés

Pour pouvoir avoir une idée de la pertinence des thèmes qui seront générés sur les copies et assister le futur évaluateur humain dans l'appréciation des modèles, il nous semble important de commencer par visualiser les différents concepts qui caractérisent les documents de notre corpus. Pour ce faire, nous avons procédé à une extraction des mots signifiants et mots-clés en ayant recours à trois méthodes non-supervisées différentes que nous détaillerons ci-dessous. Il est nécessaire d'extraire trois ensembles de mots différents pour chaque méthode afin d'observer l'influence des deux sections mais également afin de pouvoir réutiliser ces mots, ou au moins une partie en tant que *stopwords*. En effet, nous désirons savoir si la production utilise des mots dans le champs lexical du sujet mais réutiliser directement les mots des sujets pour former nos lexiques ne présenterait que peu d'intérêt et limiterait grandement leur vocabulaire en plus d'anéantir l'intérêt statistique de l'étude. On peut noter qu'utiliser un mot du sujet sous une autre forme ou un autre genre grammatical peut être tout de même valorisé, on se limitera donc à filtrer uniquement les mots identiques à ceux des sujets. En outre, retirer une partie des mots faisant partie des thèmes représentés dans nos sujets pourrait sensiblement alterer la qualité des topics générés par nos topic models, du moins selon la nature des mots-clés extraits, ce qui fait que nous ne pouvons pas non plus nous permettre d'utiliser tous les mots des sujets sans distinction.

Pour ce qui est du pré-traitement, notre corpus étant un ensemble de courtes affirmations ou début d'articles de journaux, rédigés pour une épreuve de langue française, il contient un nombre probablement très faible voire nul de fautes de langue, d'orthographe ou même de frappe. Nous allons donc expérimenter trois méthodes non-supervisées afin d'extraire les mots-clés de nos sujets d'expression écrite. Nous détaillerons les traitements effectués dans les sections suivantes.

3.2.3 Extraction de mots signifiants par filtre syntaxique et retrait des stopwords

Notre première méthode est la plus simple. Afin d'extraire les mots signifiants, nous avons procédé à l'analyse automatique en partie du discours de nos documents, ce qui nous permet d'assigner une classe grammaticale à chaque mot. Pour cela, nous avons utilisé l'analyseur syntaxique de Stanford implémenté en Python, Stanza [Manning et al., 2014] qui utilise le modèle French-GSD entraîné sur le corpus GSD d'Universal Dependencies [de Marneffe et al., 2015]. Nous avons choisi de ne conserver que les noms communs, noms propres, adjectifs et verbes car ce sont les classes grammaticales les plus porteuses de sens. Nous avons ensuite filtré nos mots sur les lemmes en utilisant une liste de stopwords constituée des stopwords pour le français fournis par la bibliothèque Python SpaCy [Honnibal and Montani, 2020] que nous avons complété par la liste de stopwords français fournis par la bibliothèque Python NLTK [Bird et al., 2009]. Nous avons aussi manuellement complété la liste de stopwords en y ajoutant le lemme des verbes jugés peu signifiants ou trop vagues (par exemple les verbes : mettre, poser, faire, falloir, avoir, etc) puis avons retiré les doublons.

3.2.4 Extraction via le nombre d'occurrences : TF-IDF

Notre seconde méthode consiste à calculer par la méthode TF-IDF, chacun des mots de notre corpus et de sélectionner pour chaque document les mots dont le TF-IDF est le plus élevé. Pour cela, nous avons utilisé la biliothèque Python Scikit-learn [Pedregosa et al., 2011] qui contient tous les outils pour rapidement faire le calcul sur tous nos documents. Scikit-learn nous permet également de retirer les *stopwords* depuis une liste ainsi que ceux qui dépassent un seuil d'occurrence parmi tous les documents. Nous avons utilisé la liste de *stopwords* précédemment évoquée et avons choisi de ne pas conserver les mots présents dans plus de 90% des documents. Nous avons retiré la ponctuation de nos documents avec une expression régulière puis après avoir calculé le TF-IDF, nous avons filtré les doublons en ne conservant que le mot ayant le TF-IDF le plus élevé puis nous avons trié nos mots sur le TF-IDF par ordre décroissant. Comme nous le suspections, notre corpus possédant un nombre extrêmement faible de tokens et donc de cooccurrences, l'utilisation du TF-IDF n'est pas pertinente dans la mesure où beaucoup de mots ont alors le même TF-IDF et il nous est donc impossible de discerner les mots les plus importants parmi tous ceux relevés. Presque tous les mots ont obtenu la même valeur de TF-IDF et notre ensemble de mots final est alors presque identique à celui obtenu via la méthode précédente. Nous n'évaluerons donc pas cette méthode.

3.2.5 Extraction par similarité des plongements lexicaux : KeyBERT

Notre troisième méthode utilise cette fois-ci les plongements lexicaux d'un modèle de langue à base de transformeur. KeyBERT [Grootendorst, 2020] désigne comme mots-clés les mots du documents dont la similarité des plongements lexicaux est la plus proche de la moyenne des plongements lexicaux des mots de toute la phrase (leur centroïde). Pour calculer cette similarité, cet outil s'appuie sur la similarité du cosinus entre deux vecteurs de plongements lexicaux. KeyBERT nous permet de sélectionner n'importe quel modèle de langue pré-entraîné distribué par Hugging

Face [Wolf et al., 2019], d'utiliser une liste de stopwords mais aussi d'utiliser une liste de mots d'amorçage (*seeded keywords*). Cette liste influencera les résultats car les mots sélectionnés seront ceux dont la similarité des plongements lexicaux est à la fois la plus proche de ceux du document mais aussi de ceux des mots de la liste d'amorçage fournie en hyper-paramètre. En outre, keyBERT permet également de spécifier en hyper-paramètre la fenêtre de tailles de n-grams sur laquelle sera effectué le calcul de similarité ce qui pourrait permettre d'obtenir des mots-clés parfois plus pertinents, comme des collocations. Un autre avantage de keyBERT est qu'il s'applique sur un document de façon indépendante et non sur un corpus. On peut donc l'appliquer à nos documents de petite taille.

Nous avons appliqué l'algorithme sur nos sujets avec plusieurs configurations que nous allons détailler. Nous avons décidé d'utiliser une fenêtre de n-grams comprise entre 1 et 2, ce qui veut dire que nous obtiendront des mots-clés à la fois unigrammes et bigrammes. Nous avons décidé d'expérimenter différents modèles de langue. Nous avons choisi les deux principaux modèles, les plus communément utilisés pour le français, flauBERT [Le et al., 2020] et camemBERT [Martin et al., 2020] mais également un modèle affiné [1](*fine-tuned* en anglais) du modèle flauBERT. Nous voulons voir dans quelle mesure les plongements lexicaux d'un modèle affiné peuvent influencer les résultats. Nous supposons que les plongements lexicaux fournis par ce genre de modèle devraient conserver les propriétés linguistiques du modèle mais que les plongements lexicaux des mots présents dans les échantillons de l'affinage devraient avoir une similarité plus élevé avec le label qui leur est associé [Merchant et al., 2020]. Ainsi, en utilisant conjointement une liste de mots d'amorçage composée de labels décrivant des thèmes généraux, qui correspondent aux labels de l'affinage de notre modèle, nous pensons pouvoir extraire de nos document des mots plus pertinents et davantage similaires aux mots de cette liste.

Nous avons alors sélectionné un modèle affiné sur une tâche de classification de phrases. Ce dernier correspond au modèle flauBERT [Scialom et al., 2020b] qui a été affiné sur des articles de presse du dataset MLSUM [Scialom et al., 2020a] sur une tâche de classification où chaque phrase est annotée de l'un des labels suivants :

« économie, opinion, politique, société, culture, sport, environnement, technologie, éducation, justice » .

Pour finir, nous avons utilisé la même liste de stopwords que précédemment et notre liste de mots d'amorçage se compose des labels et sous-labels du dataset MLSUM. Nous avons donc expérimenté la liste de mots d'amorçage sur le modèle affiné.

1. Un modèle à base de réseaux de neurones profonds dont une ou plusieurs couches (la dernière le plus souvent) a été réentraînée sur une tâche spécifique avec de nouvelles données afin d'exploiter le modèle de langue générique et de le spécialiser.

3.2.6 Résultats

Afin d'évaluer la tâche d'extraction de mots-clés, il existe plusieurs métriques mais toutes reposent sur le fait que nous avons besoin de données annotées où les mots-clés ont été extraits manuellement afin de les comparer à ceux que nous avons extrait automatiquement. Cela sous-entend néanmoins que les mots-clés de référence ont été choisis subjectivement. Nous avons tout de même annoté un échantillon de 20 documents parmi les 256 documents disponibles afin de comparer nos résultats. Nous utilisons la précision, le rappel et la F-mesure afin d'évaluer nos méthodes.

Méthode	1			3											
Modèle	Ø			flauBERT			flauBERT-mlsum			flauBERT-mlsum-skw			camemBERT		
Section	A	B	AB	A	B	AB	A	B	AB	A	B	AB	A	B	AB
Précision	0.32	0.76	0.42	0.43	0.89	0.60	0.37	0.89	0.61	0.30	0.91	0.56	0.56	0.90	0.75
Rappel	1	1	1	0.45	0.97	0.65	0.30	0.97	0.56	0.28	0.97	0.55	0.29	0.88	0.52
F-Mesure	0.49	0.86	0.59	0.44	0.93	0.65	0.33	0.93	0.58	0.29	0.94	0.55	0.38	0.90	0.62

TABLE 3.1 – Précision, Rappel et F-mesure pour chaque méthode d'extraction de mots-clés, par échantillon.

En nous penchant tout d'abord sur la première méthode (voir tableau 3.1), nous pouvons observer qu'elle accuse un rappel parfait au détriment d'une précison très faible. Etant donné que nous extrayons tous les mots considérés signifiants sans distinction, cela correspond aux résultats attendus. Nous pouvons tout de même noter que la précision sur les documents appartenant à la section B reste relativement élevée. Cela s'explique de par la petite taille de nos documents (en moyenne une dizaine de tokens) qui implique une plus grande densité de mots-clés, qui seront davantage correctement identifiés si l'on sélectionne tous les mots signifiants. Nous pouvons également voir, pour les documents de la section B de notre troisième méthode, que la précision est bien supérieure à celle de la première méthode, tout en ayant un rappel aussi très élevé. KeyBERT semble donc très efficace sur les documents de très petite taille. Nous pouvons ensuite observer, pour les plongements lexicaux du modèle flauBERT sur les documents de la section A, une précision plutôt faible qui se dégrade dans sa version affinée et encore plus en y joignant une liste de mots d'amorçage. Cependant, nous pouvons voir une très légère amélioration sur la section B, probablement grâce à l'utilisation des mots d'amorçage. En somme, sur des documents plus longs, les plongements lexicaux des modèles flauBERT semblent permettre d'identifier trop de mots en tant que mots-clés, de la même façon que la première méthode. Seul les plongements lexicaux du modèle camemBERT semblent plus sélectifs en proposant moins de mots dans le sens où KeyBERT ne sélectionne pas les mots dont la similarité de leur plongements lexicaux avec celle des plongements lexicaux du document est trop faible (avec un seuil par défaut de 0.5).

Il nous semble alors plus intéressant de sélectionner des mots-clés pour leur qualité plutôt qu'en quantité. Nous utiliserons donc les mots-clés extraits via les plongements lexicaux du modèle camemBERT pour la section A, et ceux du modèle flauBERT affiné avec une liste de mots d'amorçage pour la section B. Nous avons sélectionné les 3 n-grams avec la similarité la plus élevée pour chaque document. Dans la perspective d'améliorer nos résultats, nous pourrions par exemple, au vu des résultats sur les documents, très courts, de la section B, diviser les documents les plus longs en plusieurs parties limitées à une proposition mais nous nous contenterons ici des mots-clés issus de l'expérimentation précédemment décrite.

Nous pouvons enfin visualiser nos mots-clés à travers une figure représentant un espace en deux dimensions où les plongements lexicaux de nos mots-clés ont été capturés en contexte sur les documents de notre corpus de sujets d'expression écrite. Nous avons eu recours à une analyse en composantes principales (ACP) [Hotelling, 1933] et à une t-SNE [van der Maaten and Hinton, 2008] afin de projeter sur deux dimensions nos plongements lexicaux. Cela ne nous servira pas à distinguer des thèmes précis dans nos sujets mais simplement à avoir une idée générale des thèmes qui peuvent y être évoqués.

Nous pouvons observer (voir figures 3.1, 3.2) l'absence de vocabulaire technique ou très spécifique, la plupart des mots sont assez communs (ce qu'on attend d'un énoncé destiné à un apprenant de langue française). Les thèmes entre les deux sections semblent assez différents, ce qui nous conforte sur l'idée d'entraîner un modèle pour chaque section. Nous pouvons également relever la présence de nombreux nom propres, noms de villes ou de pays (la visualisation ne montre pas tous les mots dans un souci de lisibilité), ce qui s'avérera peut-être utile pour filtrer le bruit que ces mots pourraient générer dans les copies, car aucun sujet ne demande au candidat de traiter d'une personnalité ou d'un lieu nommé.

FIGURE 3.1 – Mots-clés extraits des sujets de la section A

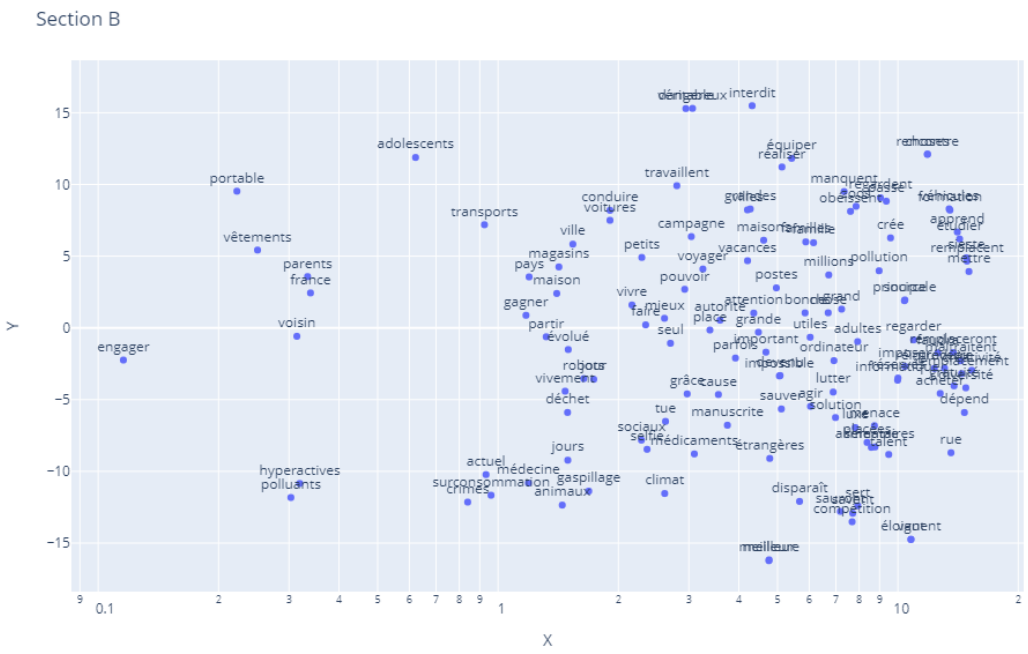


FIGURE 3.2 – Mots-clés extraits des sujets de la section B

3.2.7 Caractéristiques

	Section	A	B	AB
Données	Nombre de documents	101	155	256
	Nombre de tokens	3354	1585	4939
	Nombre moyen de tokens	33	10	19
	Minimum de tokens	21	5	5
	Maximum de tokens	49	28	49
Résultats	Nombre de mots-clés extraits	289	351	640
	Nombre d'unigrammes extraits	44	186	230
	Nombre de bigrammes extraits	245	165	410

TABLE 3.2 – Quelques données sur le corpus de sujets d'expression écrite

3.3 Description du corpus des copies

Notre corpus de copies se compose de 14 366 copies de l'épreuve d'expression écrite du TEF, chaque copie comportant les deux sections, nous possédons donc quasiment le double de documents destinés à l'entraînement. Parmi nos 256 sujets, 187 sujets différents sont représentés dans notre échantillon de copies. Le niveau de langue des productions s'étend du début de niveau B2 jusqu'au niveau C2 le plus avancé (voir tableau 3.3). On compte 110 évaluateurs différents ayant noté ces copies. Ces dernières ont toutes été produites entre août 2021 et août 2022 dans plusieurs centres de passation du TEF à travers le monde.

3.3.1 Pré-traitements

Le programme de notation automatique comprend un module de pré-traitement afin de normaliser le texte des productions écrites, permettant d'améliorer considérablement les performances de l'annotation automatique puis celles de la classification. Nous avons utilisé ce même module sur les copies que nous possédons et allons donc détailler ses différents traitements.

Les sujets de la section B sous-entendant souvent une production sous forme épistolaire, les copies sont anonymisées. Les noms, prénoms et adresses de messagerie électronique sont remplacés par un mot générique. Sont remplacés seulement les mots, situés à une distance de Damerau-Levenshtein [2] de 0 ou 1 du nom ou prénom de l'auteur de la copie, et qui ne sont pas présents dans un dictionnaire comportant les 10.000 mots les plus fréquents de la langue française. De nombreuses expressions régulières sont utilisées afin de corriger les erreurs de ponctuation, d'espacement et d'élision, mais également pour remplacer les mots écrits en langage « SMS » par leur équivalent. Les erreurs orthographiques sont corrigées par l'utilisation conjointe d'un correcteur orthographique automatique, Hunspell [Ooms, 2022] qui suggère différents mots possibles pour un mot inconnu, et des mots qu'il est possible d'écrire à partir de la phonétisation du mot à corriger. Ces deux méthodes suggérant plusieurs possibilités de correction pour un mot mal orthographié, un algorithme d'apprentissage automatique supervisé a été entraîné afin d'optimiser le choix à faire parmi les suggestions. Dans le cas où aucune suggestion ne serait délivrée, le mot est remplacé par « unknwn » pour indiquer que le mot n'a pas été trouvé dans les lexiques disponibles.

Afin d'entraîner un topic modèle sur nos données, nous devons tout d'abord lemmatiser le texte et lui retirer sa ponctuation puis ses *stopwords*. Pour cela, nous avons utilisé le même outil que pour nos sujets. Nous avons donc tokenifié et lemmatisé nos documents en utilisant Stanza, nous avons aussi utilisé la même liste de *stopwords* et nous avons également retiré les mots qui étaient présents dans plus de 90% de nos documents. Il n'est cependant pas forcément nécessaire d'effectuer ces derniers pré-traitements pour entraîner un topic model utilisant des plongements lexicaux.

2. La distance de Damerau-Levenshtein correspond au plus petit nombre d'opérations (insertion, suppression, substitution ou transposition) nécessaires pour changer un mot en un autre. Elle reprend la distance de Levenshtein en prenant en compte l'opération de transposition.

3.3.2 Accord inter-évaluateur et niveau de langue

Afin de dégager des thèmes pertinents, nous avons décidé de sélectionner seulement les copies dont le score final est compris entre les niveaux CECR se rapportant au niveau B2 jusqu'au niveau C2 car nous pensons que se limiter à un niveau de français plus élevé réduira le bruit produit par nos topic models et nous permettra d'extraire un lexique de meilleur qualité, qui d'autant plus a pu être valorisé par l'évaluateur. Néanmoins cela peut également inclure un certain biais dans la mesure où le lexique extrait pourrait davantage refléter celui mis en avant par un évaluateur humain qui juge subjectivement le qualité du vocabulaire d'une copie. Il faut tout de même noter que l'on compte 110 évaluateurs différents pour notre corpus de copies, ce biais est donc à relativiser. Il serait cependant difficile d'utiliser des copies d'un niveau inférieur de par les erreurs orthographiques trop nombreuses qui n'ont pas pu être résolues par l'algorithme de normalisation.

L'évaluation du TEF naturalisation et résidence limitant son évaluation aux niveaux A1 jusqu'à B1, seuls le TEF Canada et le TEFaq seront représentés à travers nos documents. De plus, pour être un minimum certain du niveau de langue attribué à nos échantillons, nous n'avons sélectionné que les copies dont la note attribuée par les deux évaluateurs diffère au maximum d'un niveau de CECR sans qu'il n'aille en dessous du niveau B2. Malgré cela, il existe des cas où la copie a été évaluée au dessus du niveau permettant sa sélection mais où l'une des deux section comporte un texte non achevé ou bâclé. Pour filtrer ces copies, nous ne sélectionnons pas celles dont le nombre de mots inconnus est supérieur à 2 et celles dont le nombre de tokens est inférieur au tiers de la moyenne de tokens pour toutes les copies de la même section.

3.3.3 Caractéristiques

Section	**A**	**B**	**AB**
Nombre de documents	14366	14074	28440
Nombre de documents de niveau B2	3668	3547	7215
Nombre de documents de niveau C1	4682	4584	9266
Nombre de documents de niveau C2	6016	5943	11959
Nombre de tokens	1921066	3779766	5700832
Nombre moyen de tokens (par document)	134	269	200
Nombre moyen de tokens (par sujet)	20656	40210	30486
Nombre minimum de tokens (par document)	44	89	44
Nombre maximum de tokens (par document)	680	974	974
Nombre minimum de tokens (par sujet)	15626	30400	15626
Nombre maximum de tokens (par sujet)	26512	52226	52226
Nombre de sujets représentés	93	94	187

TABLE 3.3 – Quelques données sur le corpus des copies

3.3.4 Potentiels obstacles à l'entraînement d'un topic model

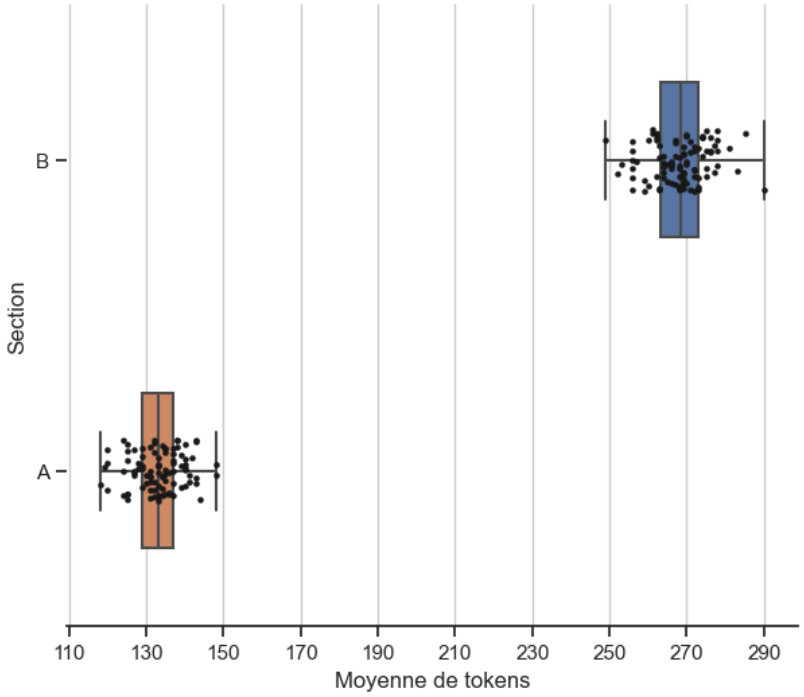


FIGURE 3.3 – Nombre moyen de tokens par document pour chaque sujet par section

Nous pouvons voir ici (figure 3.3) que pour chaque section, nous avons pour chacun de nos sujets une moyenne de tokens pour nos documents assez inégale. Nous pouvons en effet penser que certains sujets seraient plus difficiles, ce qui implique inéluctablement plusieurs problèmes auxquels peuvent être confrontés nos topic models. Tout d'abord, un sujet trop complexe peut sous-entendre qu'il ne soit pas classable dans un thème commun à d'autres sujets. De plus, même s'il peut être relié à un thème récurrent parmi nos sujets, ce sujet étant plus difficile, il pourrait inclure soit trop de documents comportant un vocabulaire spécifique, qui aura alors peu de poids pour classer correctement le document, soit davantages de documents au vocabulaire plus générique qui pourraient correspondre à une multitude de thèmes représentés parmi nos sujets. Néanmoins, nous pouvons tout aussi bien penser qu'un sujet pourrait aussi être simplement moins inspirant, ou moins d'actualité, ce qui peut causer les mêmes problèmes que ceux évoqués précédemment mais peut aussi juste réduire la taille du document produit sans affecter la qualité de son vocabulaire.

En outre, nous pouvons aussi nous demander dans quelle mesure le niveau de langue peut affecter le vocabulaire présent dans nos documents, vu que nous avons presque deux fois plus de documents de niveau C2 que de niveau B2, bien que ces derniers aient un niveau relativement proche. Pour tenter de comprendre cela, nous avons choisi d'extraire pour chaque section et niveau de langue, les 10 noms communs au TF-IDF le plus élevé du corpus en excluant les mots présents dans l'énoncé correspondant. Nous pensons que le TF-IDF peut être un bon indicateur afin de distinguer le vocabulaire le plus élaboré parmi nos documents. Le vocabulaire de nos documents d'un niveau de langue plus élevé possédera donc en théorie, davantage de mots dont le TF-IDF est proche de 1 et qui peuvent être jugés plus complexes. Pour

corroborer le jugement que nous pourrions émettre sur le vocabulaire d'un niveau de langue, nous nous sommes appuyés sur le dictionnaire Flelex [François et al., 2014] qui est une base de données du vocabulaire du français où chaque mot est annoté selon sa distribution parmi les niveaux CECR.

Copies	Section A	Flelex	TF-IDF moy.
B2	bambin, con, corbeau, superviseur, robe, collection, pizza, junior, poisson, challenge	B1	0.793
C1	assistance, respect, balcon, neveu, bracelet, as, civil, frigo, diamant, administrateur	B1	0.806
C2	bébé, malice, autocollant, bourreau, veste, os, constatation, vice, vétéran, statue	B2	0.736
—	Section B	—	—
B2	enfant, info, robot, âge, femme, selfie, vote, animal, voisin, salaire	B2	0.739
C1	magasin, ado, loi, île, machine, employé, médicament, transport, région, enfant	B1	0.709
C2	marchandise, crayon, relation, bonheur, compositeur, mutation, francophone, rêve, romain, hyperactivité	B2	0.730

TABLE 3.4 – Les 10 noms communs au TF-IDF le plus élevé par niveau et section avec leur moyenne de niveau selon le dictionnaire Flelex et TF-IDF moyen

Le niveau moyen du vocabulaire ainsi que son TF-IDF moyen sont extrêmement proches (voir tableau 3.4). Nous pouvons imaginer que les sujets ne suggèrent simplement pas le recours à un vocabulaire trop spécifique que seul un niveau très avancé pourrait maîtriser, bien que nous observons une très légère progression du niveau de complexité du vocabulaire dans les ensembles de mots identifiés. Il faut quand même garder en tête qu'avoir beaucoup de mots trop rares n'est pas propice à l'entraînement de certains topic models et que le TF-IDF peut-être utilisé conjointement à un topic model pour différencier les mots d'un niveau de langue faible avec ceux d'un niveau plus élevé [Megumi et al., 2019]. Dans notre cas, nos documents se limitant à des niveaux CECR plutôt avancés, compris entre B2 et C2, nous pouvons penser d'après les donnés précédentes que la répartition du niveau de langue n'influencera pas, ou très peu le diversité du vocabulaire. En revanche, les sujets vont quant à eux forcément supposer l'utilisation de mots plus spécifiques, dont la majorité ne devraient pas poser problème étant donné le nombre assez important de copies pour chaque sujet. Nous pourrions éventuellement filtrer les mots qui n'apparaissent qu'un certain nombre de fois dans le corpus si nous constatons du bruit dans les topics qui seront générés.

3.4 Conclusion

Dans ce chapitre, nous avons brièvement présenté nos deux corpus. Le premier comporte nos sujets d'expression écrite mais est trop petit pour pouvoir l'exploiter dans un topic model. Il nous a servi néanmoins à générer des mots-clés qui pourraient permettre de réduire le bruit auquel serait soumis nos topic models. Nous avons également relevé le fait que la qualité de nos modèles pourrait dépendre de la section sur laquelle ils auraient été entraînés de par la nature de la tâché demandée. Enfin, nous avons vu que certains thèmes pourraient ne pas être représentés dans nos modèles à cause d'une difficulté accrue de certains sujets ou bien qu'un vocabulaire trop rare dans un document pourrait perturber sa classification parmi les topics générés.

CHAPITRE 4

ENTRAÎNEMENT

Sommaire

4.1 Introduction

C'est dans ce chapitre que nous détaillons l'entraînement et les résultats de nos différents topic models où nous utiliserons plusieurs algorithmes sous plusieurs configurations sur trois jeux de données différents. Un pour la section A, un autre pour la section B et un dernier pour les deux sections à la fois. Nous verrons pour commencer quelques algorithmes de la première classe (les modèles probabilistes) puis certains de la seconde classe (les modèles à factorisation vectorielle). Avant cela, il nous semble qu'il sera opportun de détailler brièvement le fonctionnement de chacun de nos modèles.

Pour ce qui est de l'évaluation de nos modèles, nous avons choisi d'utiliser une mesure de cohérence appelée *Cv* [Röder et al., 2015] obtenue en évaluant un nombre prédéterminé de mots du vecteur de chaque topic généré (les mots dont la probabilité, ou la similarité cosinus, selon le modèle, est la plus élevée). Il nous faut rappeler qu'une mesure de cohérence sert à évaluer un topic de façon indépendante. Elle sert à mesurer le degré de similarité des mots qui composent un topic. Cette mesure combine donc l'utilisation de la NPMI (PMI normalisée) et de la similarité du cosinus. La NPMI indique dans quelle mesure deux mots cooccurrent par rapport à leur nombre d'occurrences respectives au sein d'un corpus. Ainsi, pour chaque vecteur, la NPMI est calculée pour chaque paire de mots parmi les mots sélectionnés dans une fenêtre glissante de 110 mots à travers tout le corpus. L'utilisation d'une fenêtre glissante produisant un vecteur d'un nombre identique de variables pour chaque paire

de mots du topic, la similarité cosinus est calculée entre chacun des vecteurs précédemments produits et la moyenne arithmétique entre les similarités cosinus donne la Cv-cohérence d'un topic. Pour obtenir la Cv-cohérence de notre modèle, on effectue une nouvelle fois le calcul de la moyenne arithmétique de la Cv-cohérence de chacun de nos topics. Nous avons choisi cette métrique par rapport aux autres métriques de cohérence car étant la seule comprise entre 0 et 1, elle reste simple à interpréter. Néanmoins, il a été relevé que dans certains cas, lorsque que les paires de mots d'un topic généré ont un nombre extrêmement faible de coocurrences, la corrélation de la mesure avec l'interprétation humaine de la qualité du topic généré a tendance à s'affaiblir. Pour éviter cela, nous avons filtré les mots trop rares de notre corpus lemmatisé. En outre, la Cv-cohérence est une mesure quantitative, mais avoir un résultat élevé ne témoigne pas forcément de topics de qualité. En effet, étant donnée que la mesure se base sur la NPMI, cela veut dire que plus les topics seront composés de mots cooccurrents, plus le score sera proche de 1 et à l'inverse, si les mots cooccurent peut, la Cv-cohérence se rapprochera de 0. Néanmoins, pour qu'un topic soit suffisamment qualitatif, il faut des mots sémantiquement proches ou que l'on peut retrouver dans le champs lexical d'un thème donné. Un score de Cv-cohérence proche de 1 releverait pourtant d'un mauvais topic car cela voudrait dire qu'il ne serait composé que de mots qui cooccurrent à chacune de leur occurence dans le corpus, comme certains n-grams. Par exemple, un topic évalué sur les mots « états, unis, amérique, france, espagne, japon » dans un corpus où l'on évoque souvent ces pays les uns à la suite des autres donnera un score très élevé alors que les mots 'états' et 'unis' devraient réduire la qualité du topic mais étant donné que les mots « États-unis d'Amérique » formeraient un trigram systématique, cela produirait un score élevé.

Notre seconde métrique s'appelle la Topic diversity [Dieng et al., 2019] et est calculée simplement en faisant la moyenne du pourcentage de mots spécifiques (que l'on ne trouve pas dans un autre topic) parmi un nombre prédéterminé de mots pour chaque topic (identiquement à la Cv-cohérence). Sa valeur est comprise entre 0 et 1. Idéalement, plus elle se rapproche de 1, plus le topic généré est qualitatif dans la mesure où l'on souhaite un thème différent pour chaque topic généré, bien qu'un mot puisse couvrir plusieurs thèmes différents. Nous avons choisi d'évaluer tous nos topics sur les 10 premiers mots leurs appartenant.

Il existe nombreuses métriques, notamment certaines destinées à être plus précises afin d'évaluer les topic models utilisant des plongements lexicaux. Cependant, étant donné que nous voulons comparer les performances de topic models utilisant les deux représentations vectorielles, nous utiliserons les deux métriques évoquées précédemment pour tous nos modèles. Notre but est donc de déterminer le modèle le plus performant, qui collera le mieux à nos données et à notre objectif. Nous chercherons donc à maximiser les scores des deux métriques et à observer les ensembles de mots des modèles ayant obtenus les meilleurs scores, analyser manuellement les ensembles de mots de chaque modèle étant une tâche beaucoup trop chronophage. Le score de cohérence devrait peser davantage que celui de la diversité dans la mesure où deux thèmes différents peuvent avoir des mots en commun. Maximiser cette mesure et obtenir des mots uniques pour chaque topic peut néanmoins permettre de mieux discriminer le niveau de langue des candidats une fois le lexique implémenté dans le système de notation automatique.

4.2 Modèles probabilistes

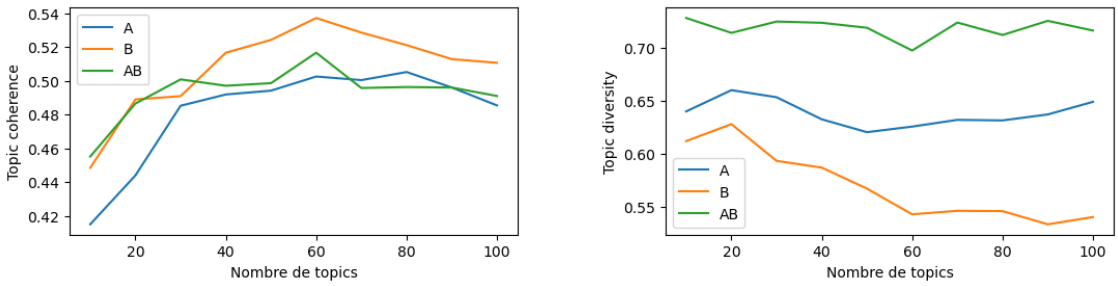
Nous avons commencé par expérimenter trois modèles distributionnels basés sur l'allocation latente de Dirichlet. Nous avons utilisé la librairie Python OCTIS [Terragni et al., 2021] afin d'entraîner les modèles LDA et CTM. Pour entraîner notre modèle BTM, nous avons utilisé la librairie Python Bitermplus [Terpilowski, 2022] en y implémentant les objets de la librairie OCTIS permettant l'évaluation du modèle afin d'utiliser une métrique identique pour évaluer tous nos modèles.

4.2.1 Latent Dirichlet Allocation

Bien que la LDA [Blei et al., 2003] ne soit pas totalement adaptée aux textes courts, les documents de la section B pourraient avoir une taille suffisante pour obtenir des résultats satisfaisants. De plus, elle constitue un bon sujet témoin afin de discerner les différences de performance entre les topic models que nous expérimenterons par la suite. Par ailleurs, nous détaillons la LDA et son processus itératif dans un schéma consultable en annexe (A.1). L'algorithme de la LDA ne prend que trois hyper-paramètres principaux afin d'optimiser le modèle :

- K, correspondant au nombre de topics en sortie,
- α, le poids qui ajuste le nombre de topics par document,
- β, le poids qui ajuste le nombre de mots par topic.

Etant donné que nous cherchons à définir un seul lexique pour chaque sujet d'expression écrite, et qu'un document est produit à partir d'un unique sujet, nous avons choisi de paramétrer le modèle avec α sur une faible valeur, c'est à dire 0.1. Nous avons ensuite choisi 0.01 comme valeur pour β, qui correspond à la valeur par défaut pour le modèle et nous assure un nombre suffisant de mots par topic. Pour ce qui est du nombre de topics, nous avons effectué plusieurs entraînements avec K compris entre 10 et 100, en augmentant K de 10 à chaque fois. Nous avons entraînés nos modèles sur 100 itérations sur le jeu de données. Nous avons répété ce processus pour nos trois échantillons et avons obtenu les résultats ci-dessous sur la moyenne des résultats de 5 entraînements par configuration (étant donné que ces modèles se basent sur une distribution de poids aléatoires, les résultats diffèrent à chaque nouvel entraînement). Précisons que nous avons entraîné nos modèles LDA sur notre corpus prétraité et lemmatisé.



(a) Cv-cohérence

(b) Topic diversity

FIGURE 4.1 – LDA : Moyenne de la Cv-cohérence et de la topic diversity selon nombre de topics, par section

Tout d'abord, nous pouvons voir en figure 4.1 que l'échantillon regroupant les sections A et B obtient le meilleur résultat pour la topic diversity. Ce n'est pas très étonnant étant donné qu'il contient presque deux fois plus de vocabulaire. Il faut cependant constater que l'écart de résultats entre les échantillons reste faible. Pour la topic diversity, la section B comportant beaucoup de sujets portant sur le même thème, nous pouvons imaginer que pour la taille du vocabulaire de l'échantillon, avoir peu de topics donne de meilleurs résultats. Cependant, avec un nombre trop élevé de topics, certains mots se répéteront forcément. La section B obtient des scores de cohérence plus élevés, ce qui est sûrement causé par la nature des sujets d'expression écrite de cette section, qui suggèrent une utilisation de termes précis et moins exotiques, comme évoqué précédemment. Il est étonnant de constater que la diversité ne baisse pas proportionnellement avec le nombre de topics générés pour l'échantillon A et AB. Nous pouvons penser que le vocabulaire est suffisamment riche et bien réparti parmi les documents, ce qui pourrait expliquer ces résultats.

Nous pouvons alors sélectionner un modèle présentant un bon ratio cohérence-diversité et visualiser les premiers mots des topics produits afin de juger de leur qualité (tableau 4.1).

1	2	3	4	5	6	7	8	9	10
enfant	**mariage**	**monsieur**	**internet**	**police**	**star**	**langue**	**petit**	**animal**	**pouvoir**
école	femme	journal	réseau	homme	pouvoir	pays	jeune	pouvoir	vie
parent	couple	pouvoir	social	policier	grand	étranger	an	nuit	jeune
devoir	rencontrer	article	livre	pouvoir	âgé	apprendre	prendre	sac	permettre
élève	vie	devoir	vie	prendre	dernier	monde	apprendre	maison	monde
pouvoir	marier	jean	lire	jeune	jour	culture	trouver	voir	devoir
télévision	célibataire	travail	journal	dernier	an	pouvoir	heure	chien	monsieur
apprendre	site	dupont	pouvoir	voir	monde	anglais	décider	prendre	journal
éducation	amour	chef	monsieur	agent	devenir	monsieur	demander	véhicule	travail
musique	marié	rédacteur	lecture	passer	sport	parler	homme	petit	humain

TABLE 4.1 – LDA : 10 mots des topics de l'échantillon AB avec K=10

Nous constatons deux problèmes dans les topics générés. Pour commencer tous les topics partagent un certain nombre de mots, identiques. On retrouve les mots « pouvoir » et « devoir » dans presque chaque topic, probablement du fait qu'il s'agisse de verbes modaux très présents en français. Nous ne les avons pas retiré lors du prétraitement car leur nom commun équivalent aurait beaucoup de sens dans un topic. Par exemple, nous pouvons attribuer le premier topic généré au thème de l'éducation mais nous ne pouvons pas savoir pour le mot « devoir » s'il est question du verbe ou du nom dont la présence ici aurait du sens, à l'opposé de la présence du mot « pouvoir » totalement erronée. Il aurait donc été préférable de joindre à chaque mot son annotation automatique en partie du discours pour pouvoir les différencier. Nous observons également que pour certains topics, plusieurs thèmes différents apparaissent, ce qui rend le topic difficilement interprétable voire pour certains topics, impossible à interpréter tant les mots qui les composent sont sémantiquement différents.

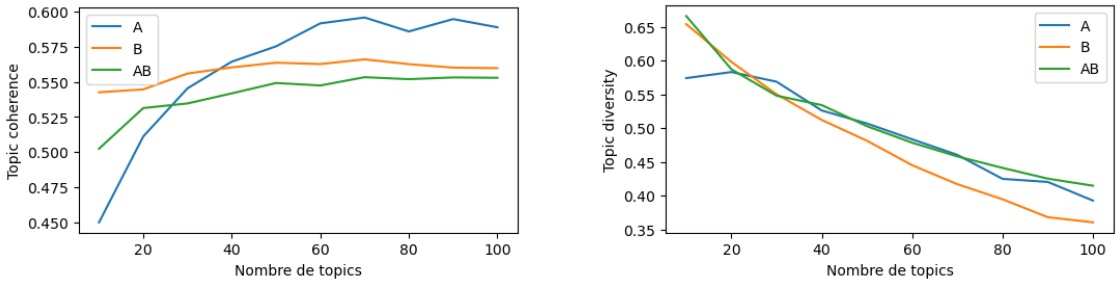
Dans la mesure où les échantillons des autres sections ont obtenu des résultats plus faibles pour nos deux mesures et que nous pouvons observer les mêmes problèmes dans leurs ensembles de mots respectifs, nous pouvons alors dire que les topics générés via une LDA sur notre corpus ne sont pas assez satisfaisants pour former plusieurs lexiques et que les modèles produits ne nous paraissent pas assez performants pour être utilisés dans notre tâche de classification.

4.2.2 Biterm Topic Model

L'algorithme BTM [Yan et al., 2013] est assez sembable à une LDA mais il est destiné aux textes courts. A la différence de la LDA, il n'itère pas sur chaque mot de chaque document mais sur chaque « biterm » de chaque document. Un biterm correspond à une paire parmi toutes les combinaisons possibles de termes présents dans une fenêtre de taille paramètrable. On fait glisser le début de la seconde fenêtre au mot suivant le dernier de la fenêtre précédente, ainsi on génère tous les biterms du corpus. Par exemple, les mots « A B C D E F » d'une fenêtre de taille 3 produira les biterms « A B » , « B C » et « A C » et la seconde fenêtre sera composée des mêmes combinaisons avec cette fois les mots « D E F » . Les données d'entrées ne sont donc plus au format sac-de-mots mais sac de biterms. Cette méthode est par ailleurs assez similaire aux méthodes n-grams. Le vocabulaire du corpus est donc l'ensemble des biterms produits sur chacune des fenêtres, augmentant alors considérablement la taille du vocabulaire du corpus, mais aussi la durée d'entraînement. La suite de l'algorithme suit la même logique d'une LDA.

Ses hyper-paramètres sont identiques à ceux d'une LDA, il faut cependant ajouter un paramètre permettant de choisir la taille de la fenêtre glissante qui produira les biterms.

Nous avons alors entraîné nos modèles BTM en utilisant les mêmes paramètres que pour les modèles de la LDA et avons effectué le même nombre d'entraînements en variant le nombre de topics. En outre, d'après les résultats obtenus dans l'article du concepteur de ce topic model, à partir de 30 mots dans la fenêtre glissante, les scores augmentent peu puis stagnent à partir de 60 mots. Vu qu'une fenêtre plus grande produit plus de biterms, nous avons donc choisi de paramétrer notre modèle sur une fenêtre de 40 mots afin de limiter la durée d'entraînement sans trop réduire les performances du modèle. Nous avons entraîné nos modèles sur notre corpus prétraité et lemmatisé. Nous avons obtenu les résultats visibles sur la figure 4.2.



(a) Cv-cohérence

(b) Topic diversity

FIGURE 4.2 – BTM : Moyenne de la Cv-cohérence et de la topic diversity selon nombre de topics, par section

Nous observons sur la figure 4.2 que la diversité est similaire pour les trois échantillons et évolue de la même façon. La fenêtre glissante étant de taille assez importante, la quantité de vocabulaire de chaque échantillon a probablement dépassé un seuil au dessus duquel le score de diversité baisse linéairement avec le nombre de topics. Pour ce qui est de la cohérence, seul l'échantillon de la section A voit son score augmenter significativement avec son nombre de topics, nous pouvons donc

nous concentrer sur les topics générés par un modèle entraîné sur l'échantillon de cette section (tableau 4.2).

1	2	3	4	5
raoul	**vie**	**patron**	**chauffeur**	**étudiant**
émission	pouvoir	entreprise	voiture	examen
animateur	année	employé	bus	directeur
surprise	monde	année	route	idée
grand	jeune	travail	véhicule	université
journaliste	homme	directeur	conduire	salle
samira	grand	grand	police	pouvoir
télévision	enfant	annoncer	prendre	voisin
porter	ancien	nouveau	arrêter	classe
dernier	nouveau	fin	volant	élève

TABLE 4.2 – BTM : Mots des 5 premiers topics de l'échantillon A avec K=40

En observant l'ensemble de mots complet, nous pouvons observer que peu de topics sont interprétables et quelques mots indésirables peuvent se glisser même au sein de topics plutôt qualitatifs. Nous pouvons noter que le modèle arrive tout de même à généraliser dans la mesure où l'on ne reconnaît pas souvent les sujets d'expression écrite dans les topics générés. Les ensembles de mots des échantillons ayant obtenu un score de cohérence plus faible présentent des topics encore plus difficiles à interpréter en plus de retrouver plusieurs fois le même thème à travers différents topics. Au vu de la qualité des topics générés et de la contrainte d'une durée d'entraînement élevée, nous ne retiendrons pas ce topic model pour notre tâche de classification.

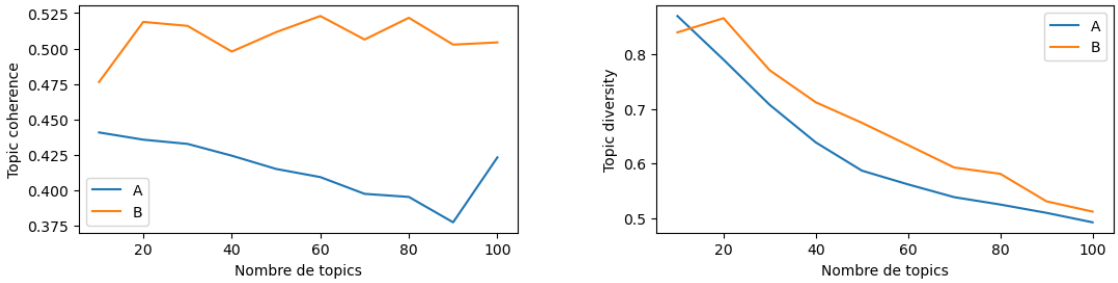
Cet algorithme a d'ailleurs été amélioré en y implémentant les plongements lexicaux [Li et al., 2019]. Cependant nous ne le testerons pas ici car son implémentation n'est pas disponible en ligne.

4.2.3 Contextualized Topic Model

Le contextualized topic model (CTM) [Bianchi et al., 2021] est basé sur la prodLDA [Srivastava and Sutton, 2017] à laquelle il rajoute l'utilisation des plongements lexicaux, ainsi que l'utilisation conjointe possible entre les sac-de-mots et les plongements lexicaux. Tout comme la prodLDA, le CTM est un réseau de neurones. La profondeur de ce dernier peut être définie en hyper-paramètre, tout comme le nombre de neurones qui le composent. Dans la mesure où le matériel à notre disposition ne permet pas un entraînement trop conséquent, nous avons limité notre réseau à deux couches et cent neurones. Nous utilisons, aussi pour cette même raison, le topic model qui ne prend que les plongements lexicaux en entrée, sans utiliser conjointement la représentation sac-de-mots. Il s'agit d'un modèle dit « *ZeroShot* » (zéro-coup en français), qui est un type de réseau de neurones basé sur les transformeurs et qui a pour particularité de pouvoir être entraîné sur des données à la fois supervisées et non supervisées sur une tâche de classification. Il pourra alors inférer les classes des documents non supervisés et si le document est jugé trop différent des classes existantes, il créera alors une nouvelle classe. Son application au domaine du *topic modelling* permet alors de créer des classes à partir de documents non supervisés afin d'obtenir une sortie similaire aux topics générés par une LDA (un topic correspondant à une classe).

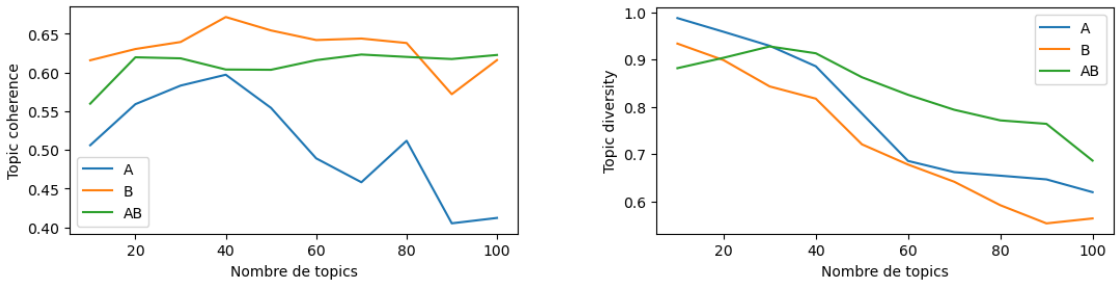
Ce modèle zeroshot présente d'autres avantages. En effet, un modèle entraîné permet de classer un nouveau document même s'il n'est pas dans la même langue (en partie grâce aux plongements lexicaux multilingues). Par extension, nous pourrions par exemple appliquer un modèle entraîné sur nos documents de niveau CECR supérieur ou égal à B2 sur des documents d'un niveau CECR inférieur afin de les classer parmi les classes (ou topics) déjà générés tout en affinant le vocabulaire qui composent ces classes mais aussi créer de nouvelles classes que l'on trouverait uniquement dans ces copies de plus faible niveau. Dans la mesure où nos ressources matérielles sont limitées, nous n'expérimentons pas cette possibilité.

Nous avons donc entraîné nos modèles en variant l'hyper-paramètre du nombre de topics en procédant de la même façon que pour nos modèles LDA. Nous avons cependant fait ces expérimentations avec deux corpus dont les pré-traitements diffèrent des modèles précédent, étant donné que ce modèle utilise des plongements lexicaux. Tout d'abord notre corpus sans aucun prétraitement, puis un corpus prétraité, dont la ponctuation a été retirée, mais non lemmatisé et dont les stopwords ont été supprimés. Nous avons constaté de meilleurs résultats avec le corpus prétraité, ce que l'on peut expliquer en partie par la gestion de la ponctuation par le tokenizer utilisé pour générer les plongements lexicaux. La ponctuation étant souvent mal utilisée en plus de la présence de caractères spéciaux non espacés, la qualité des plongements lexicaux du corpus non prétraité doit probablement en pâtir. De plus, le prétraitement retirant la ponctuation, cela nous a permis d'entraîner un modèle sur l'échantillon des sections conjointes A et B dans la mesure où la mémoire vive à disposition sur notre appareil (A.2) était insuffisante pour un entraînement sur le corpus non prétraité. Nous avons utilisé les plongements lexicaux du modèle de langue pré-entraîné sentence-camemBERT-large disponible sur le site HuggingFace [Wolf et al., 2019]. Ce topic modèle utilise les plongements lexicaux formés au niveau de la phrase afin de déduire quels documents sont similaires afin de leur attribuer un topic, c'est pour cela que seuls les transformeurs pour les phrases [Reimers and Gurevych, 2019] sont utilisables.



(a) Cv-cohérence (b) Topic diversity

FIGURE 4.3 – CTM : Moyenne de la Cv-cohérence et de la topic diversity selon nombre de topics, par section. Sans prétraitement.



(a) Cv-cohérence (b) Topic diversity

FIGURE 4.4 – CTM : Moyenne de la Cv-cohérence et de la topic diversity selon nombre de topics, par section. Avec prétraitement.

Nous observons en figures 4.3 et 4.4 un résultat différent des précédents modèles pour la cohérence, qui stagne ou baisse au lieu d'augmenter avec le nombre de topics. Les scores maximaux sont bien plus élevés que ceux des précédent modèles (voir tableaux A.1, A.2 pour comparer). Les modèles entraînés uniquement sur la section A frôlent une diversité parfaite, ce qui témoigne d'une certaine richesse dans le vocabulaire de cette section, ce qui peut cependant témoigner que les topics générés sont moins cohérents, tandis que la section B, bien que possédant des documents plus longs, obtient un score légèrement inférieur. Du côté de la cohérence, la section B est toujours bien au dessus de la section A. Globalement, la section B obtient le meilleur ratio diversité-cohérence. Les modèles entraînés sur l'échantillon mêlant les deux sections obtiennent des scores que l'on peut dire moyens dans le sens où ils se situent entre les résultats de la section A et ceux de la section B pour chaque métrique, ce que nous pourrions expliquer par l'utilisation de plongement lexicaux. Nous voyons aussi qu'augmenter le nombre de topics n'influence négativement que la section A alors même qu'un plus grand nombre de thèmes différents sont représentés dans les sujets de cette section.

Enfin nous pouvons visualiser les mots des 5 premier topics pour le modèle ayant obtenu les meilleurs résultats (tableau 4.3).

1	2	3	4	5
smartphones	**parents**	**langue**	**consommation**	**art**
salle	enfants	langues	environnement	romans
smartphone	enfant	maternelle	surconsommation	musique
informatique	discipline	pays	consommateurs	livres
stylo	maison	anglais	produits	lecture
jeunes	choix	apprendre	commerces	journaux
classe	mineurs	culture	alimentaires	livre
jeux	ados	cultures	gaspillage	lire
formation	sorties	chinois	transport	instrument
manuscrite	nuit	tourisme	limiter	films

TABLE 4.3 – CTM : Mots des 5 premiers topics de l'échantillon B avec K=10

En plus d'obtenir un score de cohérence et de diversité élevé, les modèles entraînés sur la section B proposent des topics plutôt qualitatifs, que nous trouvons cohérents et simples à interpréter. Nous pouvons facilement leur attribuer à chacun un thème très général (par exemple dans l'ordre jeunesse et technologie, éducation, apprentissage d'une langue étrangère, consommation, culture). Nous pouvons néanmoins constater que les topics ont tendance à inclure deux fois le même mot, le doublon étant le même mot au pluriel. Il nous est difficile de juger de la pertinence d'avoir un mot identique au pluriel mais dans la mesure où certains mots au pluriel ont un sens et un contexte différent de leur version au singulier, et étant difficile de faire la distinction automatiquement, il nous semble tout de même plus avisé de les conserver bien qu'ils présentent l'inconvénient d'augmenter facticement les scores de diversité. Nous pouvons également constater, en dehors de l'exemple présenté ci-dessus, un topic formé suite au format épistolaire souvent employé dans les copies de cette section. Un prétraitement visant à normaliser le document devrait empêcher le modèle de produire ce genre de topics. Dans la mesure où le score de cohérence est plutôt stable peu importe le nombre de topics mais que celui de diversité chute presque linéairement selon le nombre de topics, nous pouvons comprendre que les modèles avec plus de topics vont présenter des topics qui contiennent le même thème, composé des mêmes mots. Il nous semble alors plus intéressant de sélectionner un modèle avec peu de topics pour effectuer un classement des sujets d'expression écrite de la section B d'autant plus qu'il est nécéssaire de limiter le nombre de classes pour pouvoir effectuer une évaluation humaine.

Nous pouvons ensuite observer certains topics générés avec l'échantillon de la section A (tableau 4.4).

1	2	3	4	5
police	**maire**	**passagers**	**pompiers**	**tom**
poste	restaurant	motard	biche	adolescents
voleur	pizzas	bali	singe	instagram
bijoux	repas	milan	animaux	moqueries
boutique	chef	atterrissage	secours	photos
bijouterie	loterie	pilote	chasseurs	livres
magasin	organisateurs	atterrir	artiste	classe
policier	cuisinier	valise	rayon	adolescent
ville	couples	tokyo	animal	admiration
centre	organisateur	dinosaure	clients	professeur

1	2	3	4	5
course	**cadeaux**	**visiteur**	**loterie**	**biche**
retard	liste	exposition	ticket	clients
cycliste	maison	art	gagnants	singes
passager	noms	verre	organisateur	chasseurs
coureur	cadeau	critique	concours	protection
avion	sac	oeuvre	voyage	courses
destination	rempli	mexico	millions	panique
arriver	famille	maladroit	gagnant	rayon
chemin	enfants	contemporain	euros	chasse
cousin	acheter	mexique	somme	passage

TABLE 4.4 – CTM : Mots des 5 premiers topics de l'échantillon A avec K=10 et avec K=40

Nous ne pouvons pas dire que les topics générés soient totalement incohérents, mais en connaissant le contenu des sujets d'expression écrite de la section A, nous pouvons clairement discerner quels sujets ont été mélangés à l'intérieur de

chaque topic, ce qui produit également des mots totalement indésirables comme « dinosaure » ou bien les noms de ville ou pays. Augmenter le nombre de topics ne fait qu'estomper le phénomène. Avec 40 topics, nous avons presque déjà la moitié du nombre de sujets représentés dans notre corpus d'entraînement (93 sujets), nous pouvons observer pour chaque topic une mixture des mots de deux à trois sujets différents. Nous pouvons craindre que le modèle entraîné sur la section A soit sur-ajusté à ses données d'apprentissage, peu importe le nombre de topics générés. Nous pouvons observer la même chose pour les topics d'un modèle entraîné sur l'échantillon des deux sections.

En conclusion, nous pensons qu'un modèle entraîné sur la section B pourrait obtenir de bons résultats pour classifier nos sujets tandis que les résultats d'un modèle entraîné sur la section A ou A et B sont plus incertains. Nous pourrions néanmoins utiliser un modèle entraîné sur la section B pour classifier les sujets des deux sections dans la mesure où ce modèle généralisera davantage.

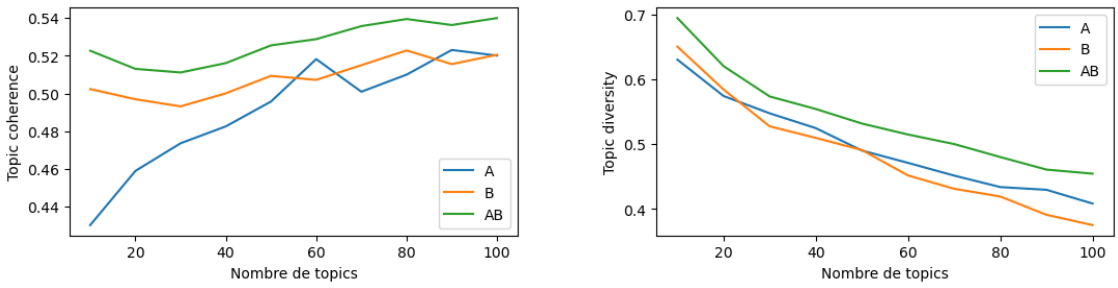
4.3 Modèles à base de factorisation vectorielle

Nous avons expérimenté deux modèles différents de cette classe, un modèle de factorisation par matrices non négatives (NMF) [Shahnaz et al., 2006] et un modèle BERTopic [Grootendorst, 2022]. Le modèle NMF a été implémenté avec OCTIS tandis que nous utilisons la bibliothèque Python dédiée pour entraîner les modèles BERTopic.

4.3.1 Non-negative Matrix Factorization

La factorisation par matrices non négatives est un algorithme [Lee and Seung, 1999] dont le principe se rapproche de celui d'une LDA et dont la sortie est en tout point similaire. Il a donc été modifié pour être adapté au *topic modelling*. Ce topic model à l'avantage d'être bien plus rapide à entaîner sur de longs documents dans la mesure où il s'adapte très bien à la représentation sac-de-mots en ignorant les valeurs inférieures ou égales à 0 de la matrice d'entrée. Le nombre de documents a cependant la même influence sur la durée d'entraînement.

Nous avons utilisé le corpus lemmatisé et nous avons donc entraîné nos modèles en variant l'hyper-paramètres du nombre de topics et avons obtenu les résultats présentés en figure 4.5.



(a) Cv-cohérence

(b) Topic diversity

FIGURE 4.5 – NMF : Moyenne de la Cv-cohérence et de la topic diversity selon nombre de topics, par section.

Contrairement aux modèles précédents, le score de cohérence est ici plus élevé pour les modèles entraînés sur l'échantillon des documents de la section A et B. Cela peut être dû à la plus grande quantité de données d'entraînement, ce qui peut améliorer les scores pour cette classe de topic models. Nous observons alors les ensembles de mots ayant obtenu les meilleurs résultats (tableau 4.5).

1	2	3	4	5
vie	**pouvoir**	**enfant**	**langue**	**internet**
professionnel	social	parent	étranger	livre
temps	réseau	école	apprendre	réseau
vivre	permettre	apprendre	maternel	social
privé	utiliser	petit	anglais	monde
important	humain	maison	culture	information
pouvoir	devoir	éducation	parler	lire
rêve	smartphone	discipline	apprentissage	journal
réussir	ordinateur	an	monde	permettre
famille	relation	télévision	permettre	site

TABLE 4.5 – NMF : Mots des 5 premiers topics de l'échantillon AB avec K=10

Les topics générés sont plutôt interprétables. Néanmoins, nous constatons l'absence de topics nous rappelant les sujets de la section A. Nous pensons alors que ce modèle est sous-ajusté aux données provenant de cette section. Nous pouvons observer quelques topics de la section A pour le confirmer (tableau 4.6).

1	2	3	4	5
police	**enfant**	**petit**	**fille**	**jeune**
policier	petit	femme	prendre	homme
agent	grand	heure	pouvoir	an
pouvoir	trouver	voir	vie	trouver
arrêter	parent	commencer	petit	demander
magasin	agent	retrouver	grand	jour
prendre	garçon	parent	mère	ami
lieu	cadeau	garçon	vivre	prendre
enquête	famille	an	moment	dernier
poste	mère	magasin	rendre	grand

TABLE 4.6 – NMF : Mots des 5 premiers topics de l'échantillon A avec K=10

En effet, les topics générés avec l'échantillon de la section A sont difficilement interprétables et les topics générés sur l'échantillon des deux sections ne se retrouvent pas dans celui-ci. Les résultats obtenus avec ce topic model nous semblent alors insuffisants pour pouvoir nous permettre de l'utiliser dans notre tâche de classification.

La NMF a d'ailleurs aussi été améliorée avec l'utilisation des plongements lexicaux [Ailem et al., 2017] mais son implémentation n'a pas été mise en ligne. Afin d'expérimenter un topic model de cette classe utilisant les plongements lexicaux nous utiliserons BERTopic.

4.3.2 BERTopic

BERTopic a la particularité d'associer une multitude de techniques afin de générer des topics à partir de documents textuels. Il a tout d'abord besoin d'effectuer une réduction par factorisation de matrices de la taille des plongements lexicaux extraits de nos documents. Cela va ensuite lui permettre de partitionner ces données (*clustering*) en plusieurs groupes qui formeront nos topics. Il va enfin également utiliser la représentation sac-de-mots des groupes générés et y appliquer un calcul du TF-IDF adapté (appelé class-TF-IDF) afin de sélectionner les mots les plus pertinents.

BERTopic est aussi très modulaire dans le sens où il est possible d'utiliser les plongements lexicaux contextuels issus de n'importe quel modèle de langue pré-entraîné, et qu'il est aussi possible d'implémenter n'importe quel algorithme de réduction vectorielle pour la première tâche et n'importe quel algorithme de partitionnement pour la seconde.

Nous avons donc utilisé les plongements lexicaux du modèle de langue sentence-camemBERT. Afin de réduire la taille de nos plongements lexicaux, nous avons utilisé UMAP [McInnes et al., 2018] et pour partitionner nos données, nous avons utilisé HDBSCAN [McInnes et al., 2017].

Nous avons tout d'abord paramétré UMAP afin d'optimiser la réduction de dimensions de nos plongement lexicaux. Les plongements lexicaux extraits avec notre modèle de langue contiennent 764 dimensions, le but est donc de réduire au maximum ce nombre car appliquer un algorithme de partitionnement sur des vecteurs d'aussi grande taille prendrait un temps considérable sans pour autant améliorer significativement les résultats du partitionnement. Le but étant de trouver une taille constituant un bon compromis entre la quantité d'informations à conserver et la durée d'entraînement de la tâche suivante. Nous sommes partis d'une réduction à 5 dimensions et avons incrémenté cette valeur jusqu'à ne plus observer d'amélioration de la qualité des topics générés. Nous avons donc utilisé des plongements lexicaux réduits à 20 dimensions.

Le principal obstacle pour obtenir des topics de qualité s'avère être la tâche de partitionnement. Nous avons déjà observé qu'il était difficile d'obtenir des topics satisfaisants pour la section A via les méthodes expérimentées précedemment. Nous avons tenté d'entraîner plusieurs modèles en implémentant l'algorithme de partitionnement K-means [Lloyd, 1982] et en définissant un nombre de topics arbitraire (entre 10 et 100) mais cela ne nous a pas permis d'obtenir de résultats satisfaisants. La multiplicité des thèmes présents au sein des documents de la section A ne nous permet donc pas de définir un nombre arbitraire de topics, car cela cause trop de bruit. Nous avons donc utilisé HDBSCAN qui permet de définir automatiquement le nombre de classes générées et produire ainsi le nombre de topics représentant au mieux nos données. Cependant, HDBSCAN a tendance à définir beaucoup de documents en tant que valeurs aberrantes qui seront toutes classées dans un topic « poubelle ». Les mots de ces documents ne seront alors pas utilisés pour générer les topics. Il est donc important de chercher à optimiser le modèle HDBSCAN de sorte à minimiser le nombre de documents non comptabilisés tout en réduisant également le nombre possible de topics générés. Pour cela, il est possible d'ajuster deux hyper-paramètres.

— Le nombre entier minimum de documents par classe.
— Le nombre entier minimum de voisins qu'un document doit avoir pour qu'ils forment une classe, qui doit être inférieur ou égal au nombre minimum de documents par classe.

En théorie, réduire la valeur du second paramètre (sample_size) devrait réduire le nombre de valeurs jugées aberrantes et ajuster la valeur du premier paramètre (min_cluster_size) devrait nous permettre de contrôler le nombre de topics générés. Cependant, chaque paramètre influence également le nombre de valeurs aberrantes et de topics générés sans que leur relation ne soit explicite. Nous sommes donc obligés d'adopter une approche plus empirique en testant un nombre élevé de combinaisons. Pour cela, nous avons utilisé un module Python nommé TopicTuner [Robinson, 2022] permettant d'optimiser les hyper-paramètres de HDBSCAN. Nous avons donc exécuté HDBSCAN selon plusieurs combinaisons de ses hyper-paramètres. Afin de déterminer la combinaison adéquate, nous avons expérimenté le premier paramètre pour toutes les valeurs comprises entre 2 et 500 pour toutes les valeurs du second paramètre correspondant à 10%, 25%, 50%, 75% et 100% de la valeur du premier paramètre (voir figure 4.6). Cela nous a permis de déterminer plusieurs valeurs intéressantes pour le premier paramètre avec lesquelles nous avons testé toutes les valeurs pour le second paramètre, comprises entre 1 et la valeur du premier paramètre (voir figure 4.7). Après avoir déterminé la valeur qui présentait le meilleur compromis entre nombre de topics et nombre de valeurs aberrantes, nous avons testé toutes les valeurs du premier paramètre avoisinant ce résultat, encore une fois avec toutes les valeurs du second paramètre afin de déterminer la combinaison minimisant le plus le nombre de valeur aberrantes tout en proposant un nombre de topics cohérent. Il faut cependant prendre en compte que cette combinaison d'hyper-paramètres n'est optimale que pour nos données issues des plongements lexicaux dont le nombre de dimensions a été réduit par UMAP. Si l'on change notre entrée pour un autre résultat d'UMAP, il faudra rechercher de nouveaux à optimiser ces deux hyper-paramètres.

En procédant ainsi, il est possible d'obtenir des topics qui semblent de très bonne de qualité. Cependant, plus nous restreignons le nombre de topics, plus le nombre de valeurs aberrantes est important, au dessus de 5000 documents ignorés pour 40 topics sur les documents de la section A. Cela influence négativement la couverture des topics générés. Il nous semble alors être plus judicieux de choisir un nombre plus réaliste de topics avec moins de documents ignorés. Finalement, pour la section A, les paramètres ayant généré le moins de valeurs aberrantes ont formé un modèle de 74 topics pour 2203 valeurs aberrantes. Le nombre de topics se rapprochant du nombre de sujets, le tâche de classification perd de l'intérêt. De plus, nous constatons un sur-ajustement du modèle aux données d'entraînement tant chaque topic généré se rapproche d'un sujet précis. Nous pouvons néanmoins dire que contrairement au CTM, nous n'observons pas de mélange de plusieurs sujets au sein d'un même topic ce qui nous laisse tout de même envisager l'utilisation des lexiques des topics générés pour le système de notation automatique. En procédant identiquement pour la section B, nous avons obtenu un modèle présentant 40 topics pour 341 valeurs aberrantes. En entraînant un modèle sur les deux sections nous obtenons un minimum de 121 topics pour 5301 valeurs aberrantes.

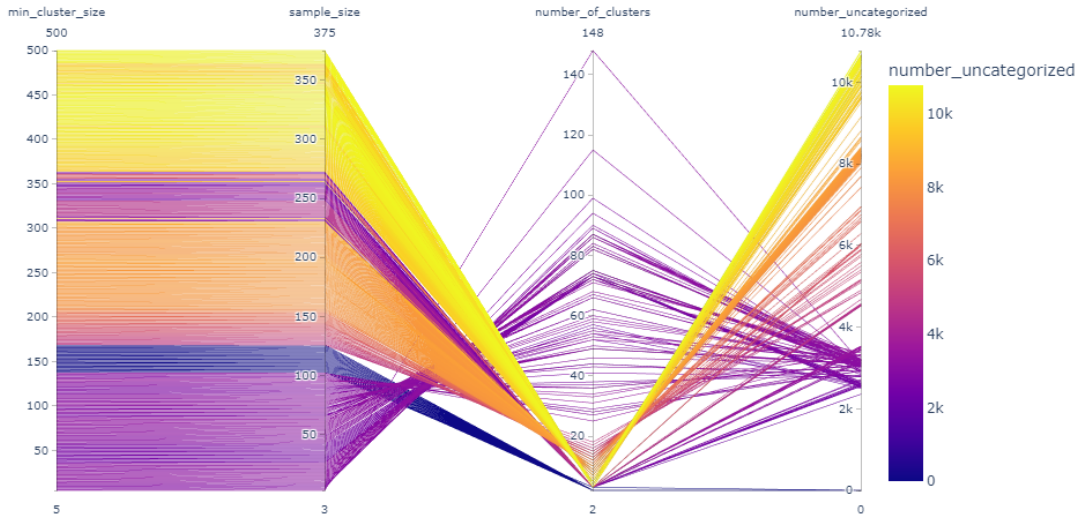


FIGURE 4.6 – HDBSCAN : nombre de clusters et valeurs aberrantes pour min_cluster_size={5 ;500} et sample_size=min_cluster_size*0.75, section A

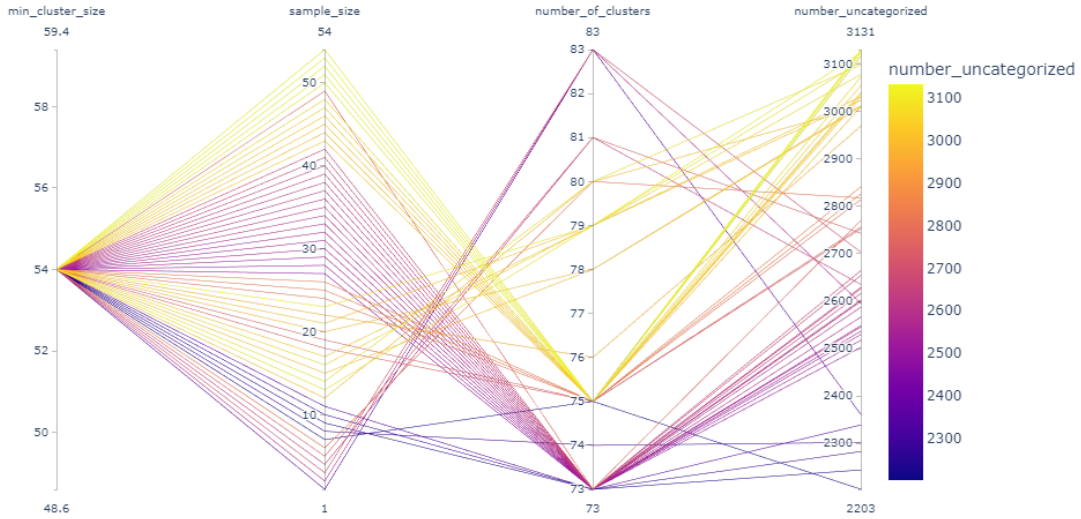
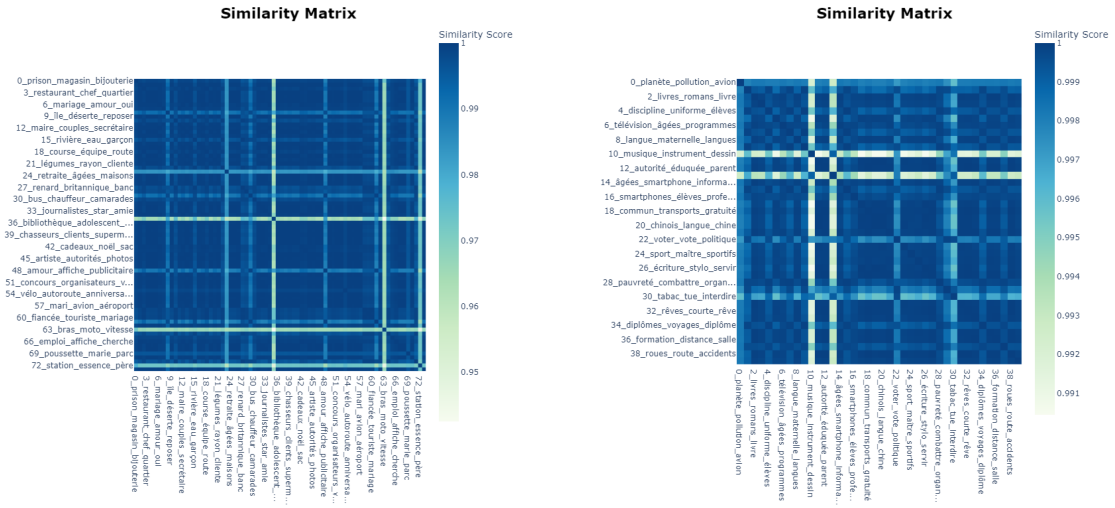


FIGURE 4.7 – HDBSCAN : nombre de clusters et valeurs aberrantes pour min_cluster_size=54 et sample_size={1 ;min_cluster_size} , section A

Dans la mesure où le nombre de valeurs aberrantes, pour un modèle entraîné sur les deux sections est bien plus élevé, et supérieur à l'addition de celles des modèles d'une section unique, tout en ayant également plus de topics, il nous semble peu pertinent d'utiliser un tel modèle d'autant plus que les topics semblent moins cohérents et en plus de cela, nous observons du bruit dans des topics formés à partir de documents de la section B causé par des mots spécifiques de la section A.

Globalement, les modèles entraînés sur une section unique produisent des topics assez qualitatifs. Chaque modèle arrive à généraliser un minimum en synthétisant les documents de sujets similaires en un seul topic sans que ce dernier n'ait à perdre de sa cohérence. Ainsi, plus particulièrement pour le modèle de la section B, nous pouvons obtenir un nombre bien inférieur de topics par rapport au nombre de sujets représentés dans nos documents mais nous restons loin de l'objectif que nous avions de limiter notre modèle à peu de topics afin de classifier nos sujets. Il n'est cependant pas possible de réduire davantage le nombre de topics du modèle dans la mesure où le nombre de valeurs aberrantes que nous avons réussi à obtenir pour une quinzaine de topics est supérieur au tiers du nombre de documents de l'échantillon d'entraînement.

Pour ce qui est de l'évaluation quantitative de nos modèles, nous n'avons pas les moyens d'évaluer ces modèles avec les métriques utilisées précédemment. Tout d'abord, le module python BERTopic n'implémente aucune métrique d'évaluation, puis les outils d'évaluation de la bilbiothèque OCTIS et BERTopic ne partagent pas la même version de leur dépendance Scikit-learn ce qui ne permet pas d'implémenter simplement les objets fournis par OCTIS. Enfin, contrairement aux topic models précédent, la sortie de BERTopic ne comporte pas les mêmes formats de données, ce qui obligerait à réécrire les objets de la bibliothèque afin de pouvoir utiliser nos métriques d'évaluation. Nous pouvons néanmoins utiliser les visualisations mises à disposition par cet outil afin d'évaluer la qualité de nos modèles. La première chose intéressante à observer est la matrice de similarité de nos topics (figure 4.8), qui nous donne une idée sur la diversité de nos topics les uns par rapport aux autres.



(a) Section A (b) Section B

FIGURE 4.8 – BERTopic : Matrices de similarité

Nos topics semblent très similaires les uns par rapport aux autres, cela peut s'expliquer du fait que l'on sélectionne un total de 30 mots pour obtenir un lexique plus conséquent. Le modèle utilisant le c-TF-IDF pour déterminer les mots les plus aptes à composer un topic, cela signifie que plus le nombre de mots sélectionné est important, moins les mots sélectionnés au dela d'un certain score sont spécifiques. Nous pouvons observer qu'au dela des 10 premiers mots ayant obtenu le meilleur c-TF-IDF (voir figure 4.9), les autres mots de chaque topic sont plus communs et donc plus susceptibles d'être présents dans plusieurs topics (sans pour autant que leur présence n'y soit pas justifiée).

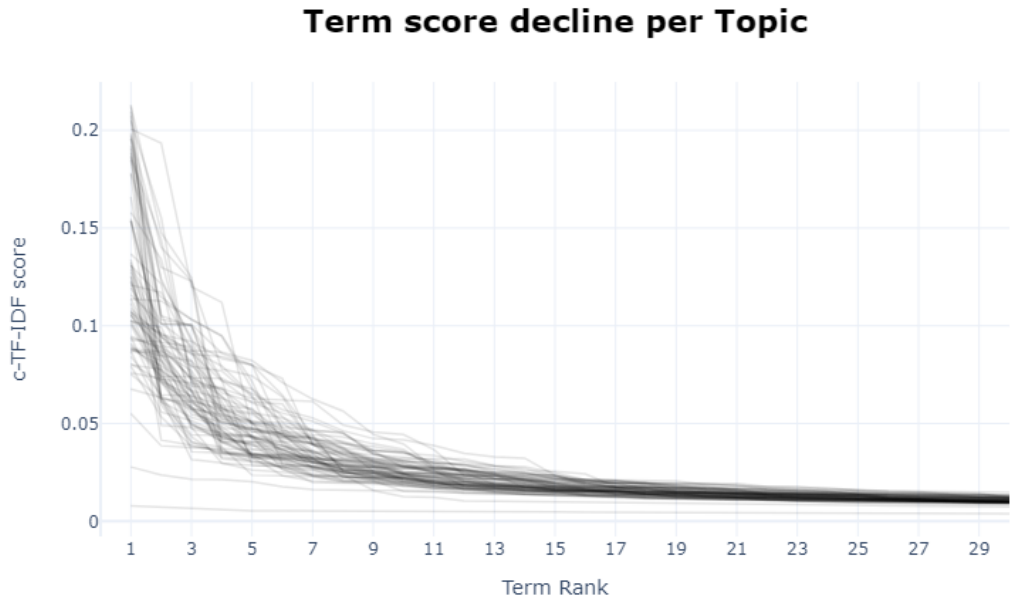


FIGURE 4.9 – BERTopic : c-TF-IDF décroissant par mot pour chaque topic

Pour ce qui est de nos lexiques (voir figures 4.10, 4.11), nous observons le même problème que pour les modèles CTM, c'est à dire que certains mots sont en double avec leur version au pluriel (uniquement quand le pluriel du mot se forme avec la lettre s, vu que ces mots produisent presque les mêmes plongements lexicaux à partir de notre modèle de langue) au sein d'un topic. Nous pouvons également voir que les mots qui ont le plus de poids pour un topic sont souvent les mots présents dans l'énoncé du sujet, ce qui néanmoins nous pose problème dans la mesure où réutiliser les mots du sujet dans sa copie n'est pas forcément valorisable. A partir de ces deux éléments, nous avons décidé de filtrer les lexiques de nos modèles en supprimant donc les doublons et les mots du sujet de nos lexiques, ce qui nous permettra également d'estimer la quantité de bruit moyen dans les topics générés pour chacun de nos modèles.

FIGURE 4.10 – BERTopic : quelques mots des 8 premiers topics de la section A

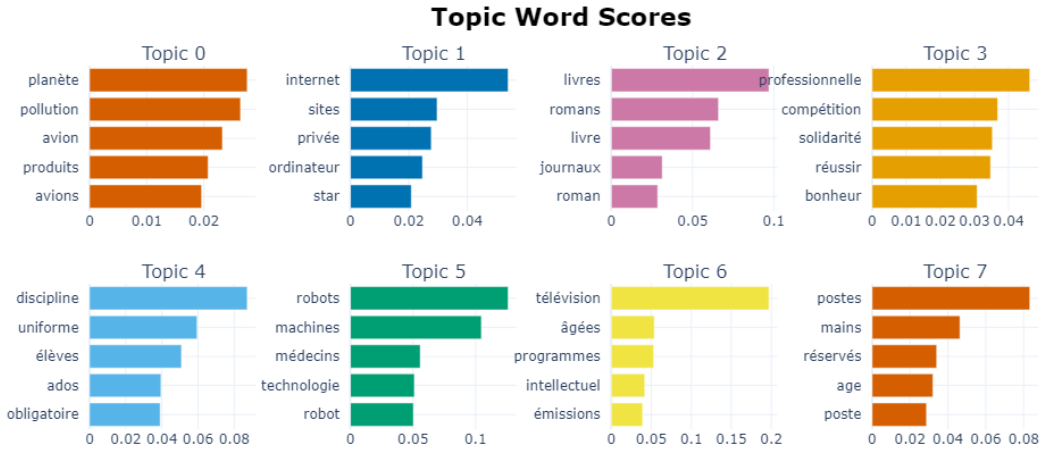


FIGURE 4.11 – BERTopic : quelques mots des 8 premiers topics de la section B

4.4 Lexiques et classification

Il va sans dire que les topic models utilisant des plongements lexicaux nous ont semblé générer des topics à la fois plus cohérents et plus diversifiés, pourvu que le nombre de topics générés corresponde à la réalité de nos données d'entraînement. En outre, de part la nature de nos données et en s'appuyant sur les expérimentations précédentes, nous pouvons dire qu'il serait impossible de réduire le nombre de lexiques pour les sujets de la section A. Nous pensons qu'il est préférable d'utiliser soit un topic model avec un nombre élevé de sujets soit un topic model déterminant automatiquement le nombre de topics à générer, même si cela peut impliquer qu'un sujet peut se voir attribuer un seul lexique alors qu'il peut s'identifier à deux topics. Nous envisageons donc d'utiliser BERTopic pour former nos lexiques de la section A, le partitionnement via HDBSCAN permettant de déterminer un nombre adéquat de topics pour les données de cette section. Pour la section B, nous constatons cependant qu'il serait possible de limiter le nombre de lexiques différents. Nous expérimentons donc un modèle CTM dans la mesure où c'est celui qui a montré les résultats les plus prometteurs avec un nombre de topics limité. Nous pouvons aussi le comparer au modèle BERTopic de cette section, car malgré un nombre de topics élevé, leur qualité l'est également.

Pour classer nos sujets d'expression écrite parmi nos topics générés, nous n'utilisons pas de classifieur spécifique, nous utilisons les fonctions de prédiction de nos topic models. Bien qu'aucun sujet n'ait été utilisé en tant que document pour l'entraînement, il est intéressant d'évaluer séparemment les sujets dont les copies sont présentes dans notre corpus (*Train* dans le tableau 4.7) et les sujets n'y étant pas représentés (*Test* dans le tableau 4.7). Cela nous permet d'évaluer dans quelle mesure le modèle entraîné peut être utilisable pour des sujets qui paraîtront à l'avenir (voir tableau 4.7).

Comme évoqué précédemment, nous considérons en tant que bruit un mot du lexique également présent dans le sujet auquel il a été attribué par la prédiction du modèle. En effet, le système de notation automatique génère une variable correspondant au nombre de mots de la copie présents dans le lexique attribué à son sujet. Cependant, la réutilisation des mots du sujet par le candidat ne serait pas une preuve de ses compétences linguistiques. Ces mots doivent alors être filtrés pour pouvoir utiliser les lexiques dans l'algorithme de notation automatique, mais avant cela, ils nous permettent d'estimer une marge d'erreur sur la prédiction de nos modèles. Il est peu probable qu'une prédiction soit correcte si le lexique désigné par le modèle ne contient aucun mot du sujet. La prédiction du modèle est effectuée par un calcul de similarité cosinus entre la moyenne des plongements lexicaux des mots d'un topic et la moyenne des plongements lexicaux des mots du documents sur lequel on effectue la prédiction. On obtient en sortie la similarité de chaque topic avec ce document. Après avoir constaté que la prédiction était toujours erronée pour les sujets n'ayant aucun mot en commun avec leur lexique, nous avons décidé de sélectionner le topic le plus similaire comportant au moins un mot du sujet. Le cas échéant, nous sélectionnons le topic le plus similaire et nous l'indiquons afin de faciliter une correction manuelle. Nous avons aussi filtré nos lexiques de sortie des mots identiques au singulier et au pluriel, formés avec la lettre s, pour ne conserver que les mots au singulier si les deux temps étaient présents dans le lexique.

Pour évaluer la prédiction de nos modèles, nous n'avons pas utilisé de métrique d'évaluation multi-classes, comme la F-mesure par exemple car au vu du nombre très élevé de classes, les scores relevés n'auraient pas été explicites d'autant plus que certains topics se trouvent être peu représentés parmi les sujets. Nous avons donc effectué une évaluation plus globale qui correspond à la proportion de sujets dont le topic été correctement attribué. Cette évaluation doit forcément être supervisée étant donné qu'il faut lire et comprendre un sujet pour savoir si le lexique qui lui a été assigné automatiquement lui correspond. Nous avons donc évalué manuellement nos paires sujet-lexique obtenues en sortie et avons calculé le pourcentage de sujets dont le lexique attribué automatiquement fait sens. Nous avons également évalué le bruit global de nos topics en prenant en compte pour chaque paire, la proportion de mots du lexique présents dans le sujet par rapport au nombre de mots que compte le lexique.

Pour le modèle CTM, nous avons constaté la récurrence d'un topic unique parmi nos 10 topics, pour chaque sortie, qui va toujours être ininterprétable et mélanger plusieurs thèmes. Ce dernier faussait également la prédiction car contenant des mots très divers, la plupart des sujets lui étaient attribués. Nous avons décidé de supprimer ce topic de la sortie pour obtenir des prédictions cohérentes. Enfin le nombre de sujets non représentés parmi les documents de la section A (marqué *) est malheureusement trop faible (7 documents) pour pouvoir l'évaluer solidement.

4.5 Résultats

Modèle	CTM		BERTopic			
Section	B		A		B	
Topics	9		74		40	
Sujets	Train	Test	Train	Test	Train	Test
Précision (%)	87.2	77	95	71*	97.9	87
Bruit (%)	8	5.1	27.6	11.9	9.3	4.6

TABLE 4.7 – Scores de classification des sujets d'expression écrite

Nous observons sur le tableau 4.7 que le bruit pour la section A est bien plus important que celui de la section B. Cela peut s'expliquer du fait que les sujets de la section A sont à la fois beaucoup plus longs, ce qui augmente les chances qu'ils possèdent un mot du lexique, mais aussi que les sujets de cette section possèdent des mots assez spécifiques. Cela produit pour BERTopic un nombre de topic bien plus élevé que pour la section B, avec des topics dont les mots seront en plus grand nombre identiques à ceux du sujet. On peut dire que le modèle peine à généraliser sur cette section même si ce dernier a réussi à regrouper les 93 sujets représentés dans les copies dans 74 topics. De plus, il nous semble que la proportion de bruit, que nous avons défini, de chaque modèle soit corrélée avec sa précision. Cela nous fait dire que l'opération consistant à sélectionner pour un sujet le topic le plus similaire, et

comportant un certain nombre de mots de ce dernier, permet d'améliorer la précision de la classification. Enfin, le modèle CTM ayant été paramétré sur un faible nombre de topics, une précision plus faible ne nous étonne pas dans la mesure où certains thèmes présents dans très peu de sujets de cette section ne sont pas représentés dans les topics de ce modèle.

4.6 Conclusion

Nous avons comparé différents algorithmes de topic models et constatons les limites des modèles utilisant une représentation sac-de-mots, dont le bruit dépend grandement de la nature des données d'entraînement et dont les scores de cohérence et de diversité restent faibles. En revanche, les topic models plus récents, utilisant les plongements lexicaux ont donné de meilleurs résultats. Cependant, l'impossibilité de réduire le nombre de topics générés pour nos données nous a poussé à nous concentrer finalement sur une approche utilisant les plongements lexicaux et permettant de déterminer automatiquement le nombre optimal de topics. Nous résumons certains points des algorithmes expérimentés dans le tableau 4.8.

Modèle	LDA	BTM	CTM	NMF	BERTopic
Pré-traitements (après normalisation, par le système de notation automatique)	- Lemmatisation et tokenisation - Retrait des stopwords - Retrait de la ponctuation - Retrait des mots trop rares ou trop courants	Identiques à la LDA	1 : Extraction des plongements lexicaux 2 : - Retrait de la ponctuation - Retrait des stopwords - Extraction des plongements lexicaux	Identiques à la LDA	- Retrait de la ponctuation - Retrait des stopwords - Extraction des plongements lexicaux
Temps approximatif des prétraitements	4 à 5 heures	Identique à la LDA	Quelques minutes	Identique à la LDA	Identique à la CTM
Pré-traitements à expérimenter	- Utilisation du mot avec son annotation automatique en partie du discours - Idéalement, expérimenter toutes les combinaisons des valeurs du seuil d'occurrence où l'on considère un mot trop rare ou trop courant	Identique à la LDA	Idéalement, expérimenter comparativement tous les pré-traitement possibles	Identique à la LDA	Identique à la CTM
Représentation vectorielle	Sac-de-mots	Sac-de-mots	Plongements lexicaux	Sac-de-mots	Plongements lexicaux
Temps approximatif d'entraînement section A (pour l'ensemble des valeurs de K et 5 exécutions)	Moins d'une heure	Une dizaine d'heures	3 à 4 heures	2 à 3 heures	Moins d'une heure pour définir les meilleurs hyper-paramètres pour le partitionnement et quelques minutes pour l'entraînement
Temps approximatif d'entraînement section B (pour l'ensemble des valeurs de K et 5 exécutions)	Moins d'une heure	Une douzaine d'heures	4 à 5 heures	2 à 3 heures	Identique à la section A
Temps approximatif d'entraînement section AB (pour l'ensemble des valeurs de K et 5 exécutions)	Moins de deux heures	Une vingtaine d'heures	Une dizaine d'heures	3 à 4 heures	Identique à la section A
Problèmes observés	- Mélange de plusieurs thèmes en un seul topic - Mots récurrents partagés par plusieurs topics - Mots indésirable au sein de certains topics - Interprétabilité difficile de certains topics	Identique à la LDA	- Présence de mots identiques au pluriel dans un même topic - Présence récurrente d'un topic ininterprétable - Carte graphique obligatoire - Très coûteux en mémoire vive	- Sous-ajustement aux donnés de la section A. - Interprétabilité difficile de certains topics	- Présence de mots identiques au pluriel dans un même topic - Demande un niveau de programmation avancé pour optimiser le modèle et les temps d'entraînement. - Impossible de réduire le nombre de topics à une valeur arbitraire sans réduire de façon conséquente l'interprétabilité des topics - Sur-ajustement aux données d'entraînement. - Carte graphique recommandée - Pas de métrique non-supervisée pour l'évaluation
Points forts observés	Apprentissage rapide.	Aucun	- Meilleurs scores et topics interprétables pour la section B uniquement - Généralise suffisamment sur la section B	- Majorité de topics interprétables mais uniquement pour la section B - Généralise trop ou n'apprend pas assez pour les autres sections	Topics de qualité et interprétables Apprentissage très rapide

TABLE 4.8 – Résumé comparatif des algorithmes expérimentés dans cette étude

CHAPITRE 5

DISCUSSION

Sommaire

5.1 Introduction

Les différents topic models que nous avons pu expérimenter nous ont permis à la fois d'observer l'évolution des résultats produits par cette classe d'algorithmes d'apprentissage automatique mais aussi de mieux comprendre les spécificités de nos données. Autant les lexiques produits nous paraissent satisfaisants, autant les méthodes utilisées pour la classification pourraient être améliorées.

5.2 Réflexions et perspectives d'améliorations

Tout d'abord, il faut préciser que l'utilisation que nous faisons d'un topic model est quelque peu détournée dans le sens où en recherche d'information, les topic models ont pour but d'informer sur les thèmes présents au sein d'un corpus, alors qu'ici nous connaissons déjà plus ou moins les thèmes mais ce qui nous intéresse s'avère être la sortie des topic models, propices à la confection de lexiques thématiques. De cette façon, il nous est plus facile d'interpréter un topic produit par un modèle et de juger de sa qualité, de par son lien direct avec les sujets d'expression écrite. Cependant, cette façon de procéder génère un biais dans l'appréciation des topics générés. On aura tendance à donner une appréciation négative à un topic s'il ne correspond pas aux attentes créées par les connaissances que l'on possède sur les sujets, ce qui nous pousse à sélectionner un modèle sur-ajusté à ses données d'entraînement et qui ne pourra donc pas traîter les sujets à paraître. Se prêter à l'évaluation qualitative des topics de sortie sans avoir connaissance des sujets, ou en se limitant aux mots-clés extraits pourraient permettre de sélectionner un modèle capable de mieux généraliser pour ensuite classer nos sujets.

Un autre problème posé s'avère être le niveau de prétraitement à effectuer sur notre corpus. Les topic models utilisant une représentation sac-de-mots ont besoin d'un nombre d'occurrence élevée pour chaque terme afin de constituer des topics cohérents, mais malgré le retrait de stopwords, la lemmatisation et le retrait de mots trop présents et trop rares, la plupart de ces modèles ont produit des topics

bruités par des mots récurrents mais pas suffisamment pour dépasser le seuil défini et abaisser ce seuil réduisant drastiquement la cohérence des topics, il semble difficile de filtrer ces cas. Identiquement, pour les modèles utilisant les plongements lexicaux, bien qu'il soit souvent recommandé par les concepteurs des différents modèles les utilisant de ne pas pré-traiter son corpus, nous observons tout de même de nettes améliorations en utilisant un corpus dont le pré-traitement couvre au moins la ponctuation et les stopwords. En revanche, filtrer les mots trop présents ou trop rares pour ces modèles réduit la cohérence des topics produits. Nous avons tenté d'utiliser les plongements lexicaux formés à partir de notre corpus non-prétraité tout en utilisant le vocabulaire de notre corpus pré-traité afin d'obtenir des plongements lexicaux plus fidèles à nos données tout en produisant des topics au vocabulaire moins bruité mais nous avons constaté que les plongements lexicaux formés sur notre corpus pré-traité donnaient tout de même de meilleurs résultats. Etudier comparativement le niveau de pré-traitement nécessaire pour chaque modèle pourrait permettre d'améliorer sensiblement la qualité des topics générés.

Le modèle CTM étant un réseau de neurones, nous avons essayé de modifier quelques hyper-paramètres comme le nombre de de neurones, de couches, le taux d'apprentissage ou la fonction d'activation mais ce type de modèle étant très long à entraîner, il nous était impossible d'expérimenter un nombre suffisamment élevé de configurations dans des temps et pour des coûts de calcul raisonnables. Il serait probablement possible d'améliorer significativement les topics générés en expérimentant chaque hyper-paramètre.

Enfin, pour ce qui est de la classification des sujets, nous avons utilisé les classes générées par nos topic models, mais il serait probablement plus efficace, du moins pour réduire le nombre de classes, de ne pas utiliser la valeur la plus élevée de la fonction de prédiction de nos modèles mais d'expérimenter différents classifieurs où nos documents auront pour caractéristiques les probabilités d'appartenance à chaque topic [Alcoforado et al., 2022].

Conclusion générale

Dans ce mémoire, nous avons étudié et expérimenté le *topic modelling* afin d'alimenter les caractéristiques destinées à entraîner un système de notation automatique. Notre tâche a plus spécifiquement porté sur la confection automatique de lexiques thématiques. Il s'agissait donc d'une tâche d'automatisation à laquelle nous avons tenté de répondre en proposant tout d'abord une visualisation de mots-clés susceptibles d'apparaître dans ces lexiques. Puis, nous avons identifié plusieurs algorithmes qu'il serait possible d'expérimenter. Nous nous sommes donc penchés sur cinq algorithmes différents publiés entre 2003 et 2022. Cela nous a permis d'étudier l'avancée des techniques de *topic modelling* qui ont su évoluer avec leur temps en implémentant tous les progrès récents en traitement automatique du langage naturel.

En outre, nos données textuelles, c'est à dire les productions écrites d'apprenants avancés en français langue étrangère d'une épreuve d'expression écrite d'un test officiel de niveau de langue française, ont présenté plusieurs spécificités compliquant l'apprentissage de topic models. Ces dernières ont fait que les techniques anciennes ont donné des résultats mitigés alors que les techniques plus récentes, celles utilisant les plongements lexicaux, ont permis de remplir en grande partie notre objectif en produisant des lexiques interprétables qui collent à nos données. L'utilisation de ces modèles pour classifier les sujets de cette expression écrite ont également donné de bons résultats même s'il serait possible d'avoir recours à une classification plus classique pour les améliorer.

Bien que se pencher sur l'automatisation de la tâche avait aussi pour but de réduire le biais apporté par une conception manuelle, le *topic modelling* impliquant aussi l'humain en tant que principal évaluateur du modèle, on peut alors remettre en question l'objectivité de la technique utilisée. La dernière interrogation qui peut-être soulevée se porte sur les résultats que donneront cette étude sur le poids et la qualité de la caractéristique correspondante dans le système de notation automatique. Si nos lexiques permettent d'obtenir des résultats au moins similaires à ceux confectionnés manuellement par le passé, notre objectif final sera en partie rempli. Il nous faudra alors étudier dans quelle mesure le critère que nous décrivons via cette caractéristique pèse dans la discrimination du niveau de langue d'un apprenant étranger.

BIBLIOGRAPHIE

[Ailem et al., 2017] Ailem, M., Salah, A., and Nadif, M. (2017). Non-negative matrix factorization meets word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1081–1084, New York, NY, USA. Association for Computing Machinery. – Cité page 50.

[Alcoforado et al., 2022] Alcoforado, A., Ferraz, T. P., Gerber, R., Bustos, E., Oliveira, A. S., Veloso, B. M., Siqueira, F. L., and Costa, A. H. R. (2022). Zeroberto: Leveraging zero-shot text classification by topic modeling. *CoRR*, abs/2201.01337. – Cité page 62.

[Banerjee and Basu, 2007] Banerjee, A. and Basu, S. (2007). Topic models over text streams: A study of batch and online unsupervised learning. In *SDM*. – Cité page 20.

[Bianchi et al., 2021] Bianchi, F., Terragni, S., Hovy, D., Nozza, D., and Fersini, E. (2021). Cross-lingual contextualized topic models with zero-shot learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1676–1683, Online. Association for Computational Linguistics. – Cité page 45.

[Bicalho et al., 2017] Bicalho, P. V., Pita, M., Pedrosa, G., Lacerda, A. M., and Pappa, G. L. (2017). A general framework to expand short text for topic modeling. *Inf. Sci.*, 393:66–81. – Cité page 21.

[Bird et al., 2009] Bird, Steven, Loper, E., and Klein, E. (2009). Natural language processing with python. https://www.nltk.org/. – Cité page 30.

[Blei and Lafferty, 2006] Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. *Proceedings of the 23rd international conference on Machine learning*. – Cité page 20.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022. – Cité pages 20 et 41.

[Bouma, 2009] Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. – Cité page 23.

[Cataldi et al., 2010] Cataldi, M., Di Caro, L., and Schifanella, C. (2010). Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, MDMKDD '10, New York, NY, USA. Association for Computing Machinery. – Cité page 22.

[CECR, 2022] CECR (2022). *Echelle globale - Tableau 1 (CECR 3.3) : Niveaux communs de compétences*. https://www.coe.int/fr/web/common-european-framework-reference-languages/table-1-cefr-3.3-common-reference-levels-global-scale. – Cité page 15.

[Chellapilla et al., 2006] Chellapilla, K., Puri, S., and Simard, P. (2006). High Performance Convolutional Neural Networks for Document Processing. In Lorette, G., editor, *Tenth International Workshop on Frontiers in Handwriting Recognition*, La Baule (France). Université de Rennes 1, Suvisoft. http://www.suvisoft.com. – Cité page 21.

[Church and Hanks, 1989] Church, K. W. and Hanks, P. (1989). Word association norms, mutual information, and lexicography. In *27th Annual Meeting of the Association for Computational Linguistics*, pages 76–83, Vancouver, British Columbia, Canada. Association for Computational Linguistics. – Cité page 23.

[Churchill and Singh, 2022] Churchill, R. and Singh, L. (2022). The evolution of topic modeling. *ACM Comput. Surv.*, 54(10s). – Cité page 22.

[Collobert and Weston, 2008] Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery. – Cité page 21.

[de Marneffe et al., 2015] de Marneffe, M.-C., Guillaume, B., Grioni, M., Dickerson, C., and Perrier, G. (2015). *UD French GSD*. https://universaldependencies.org/treebanks/fr_gsd/index.html. – Cité page 30.

[Deerwester et al., 1990] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.*, 41:391–407. – Cité page 20.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. – Cité page 21.

[Dieng et al., 2019] Dieng, A. B., Ruiz, F. J. R., and Blei, D. M. (2019). The dynamic embedded topic model. *CoRR*, abs/1907.05545. – Cité pages 21 et 40.

[Filho et al., 2020] Filho, A. H., Concatto, F., Antonio do Prado, H., and Ferneda, E. (2020). Comparing feature engineering and deep learning methods for automated essay scoring of brazilian national high school examination. – Cité page 16.

[Foltz et al., 1999] Foltz, P. W., Laham, D., and Landauer, T. K. (1999). The intelligent essay assessor: Applications to educational technology. – Cité page 16.

[François et al., 2014] François, T., Gala, N., Watrin, P., and Fairon, C. (2014). *FLELex: a graded lexical resource for French foreign learners.* In the 9th International Conference on Language Resources and Evaluation (LREC 2014). – Cité page 38.

[Ghorpade et al., 2012] Ghorpade, J., Parande, J., Kulkarni, M., and Bawaskar, A. (2012). GPGPU processing in CUDA architecture. *CoRR*, abs/1202.4347. – Cité page 21.

[Grootendorst, 2020] Grootendorst, M. (2020). Keybert: Minimal keyword extraction with bert. – Cité page 30.

[Grootendorst, 2022] Grootendorst, M. (2022). Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*. – Cité pages 22 et 49.

[Hofmann, 1999] Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, page 50–57, New York, NY, USA. Association for Computing Machinery. – Cité page 20.

[Honnibal and Montani, 2020] Honnibal, M. and Montani, I. (2020). spacy3 : Industrial-strength natural language processing in python. https://spacy.io/. – Cité page 30.

[Hotelling, 1933] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. journal of educational psychology, 24(6), 417–441. – Cité page 33.

[Jover, 2003] Jover, M. B. (2003). Contraintes en sémantique lexicale. *Langages*, 37(150):75–87. Included in a thematic issue : La constitution extrinsèque du référent. – Cité page 19.

[Lang, 1995] Lang, k. (1995). 20 newsgroups dataset. http://people.csail.mit.edu/jrennie/20Newsgroups. – Cité page 22.

[Le et al., 2020] Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., and Schwab, D. (2020). Flaubert: des modèles de langue contextualisés pré-entraînés pour le français. In *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2: Traitement Automatique des Langues Naturelles*, pages 268–278. ATALA. – Cité page 31.

[Lee and Seung, 1999] Lee, D. and Seung, S. (1999). Algorithms for non-negative matrix factorization. – Cité page 49.

[Li et al., 2019] Li, X., Zhang, A., Li, C., Guo, L., Wang, W., and Ouyang, J. (2019). Relational biterm topic model: Short-text topic modeling using word embeddings. *Computer Journal*, 62:359–372. – Cité page 44.

[Liu et al., 2015] Liu, X., Duh, K., and Matsumoto, Y. (2015). Multilingual topic models for bilingual dictionary extraction. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 14(3). – Cité page 7.

[Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137. – Cité page 51.

[Manning et al., 2014] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60. – Cité page 30.

[Martin et al., 2020] Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., and Sagot, B. (2020). CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics. – Cité page 31.

[Mayfield and Black, 2020] Mayfield, E. and Black, A. W. (2020). Should you fine-tune BERT for automated essay scoring? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 151–162, Seattle, WA, USA → Online. Association for Computational Linguistics. – Cité page 16.

[McInnes et al., 2017] McInnes, L., Healy, J., and Astels, S. (2017). hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11). – Cité page 51.

[McInnes et al., 2018] McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. – Cité page 51.

[Megumi et al., 2019] Megumi, Y., Umemura, N., and Kawano, H. (2019). Proposal of japanese vocabulary difficulty level dictionaries for automated essay scoring support system using rubric. *Journal of the Operations Research Society of China*, 8. – Cité pages 7 et 38.

[Merchant et al., 2020] Merchant, A., Rahimtoroghi, E., Pavlick, E., and Tenney, I. (2020). What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics. – Cité page 31.

[Mercier et al., 2003] Mercier, L., Verreault, C., and Lavoie, T. (2003). *Le francais, une langue à apprivoiser - Textes des conférences prononcées au Musée de la civilsation (Québec, 2000-2001) dans le cadre de l'exposition Une grande langue : le français dans tous ses états*. – Cité pages 19 et 28.

[Mian et al., 2013] Mian, L., Ge, B., Qiong, L., Jie, T., and Jiuxin, Z. (2013). *Accelerating Topic Model Training on a Single Machine*. – Cité page 21.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. – Cité page 21.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc. – Cité page 21.

[Moody, 2016] Moody, C. E. (2016). Mixing dirichlet topic models and word embeddings to make lda2vec. *CoRR*, abs/1605.02019. – Cité page 21.

[Morgan, 1978] Morgan, J. L. (1978). *Two Types of Convention in Indirect Speech Acts*, pages 261 – 280. Brill, Leiden, The Netherlands. – Cité page 19.

[Nguyen et al., 2015] Nguyen, D. Q., Billingsley, R., Du, L., and Johnson, M. (2015). Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313. – Cité page 21.

[Ooms, 2022] Ooms, J. (2022). *hunspell: High-Performance Stemmer, Tokenizer, and Spell Checker*. https://docs.ropensci.org/hunspell/ (docs), https://github.com/ropensci/hunspell (devel) https://hunspell.github.io (upstream). – Cité page 35.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. – Cité page 30.

[Quan et al., 2015] Quan, X., Kit, C., Ge, Y., and Pan, S. (2015). Short and sparse text topic modeling via self-aggregation. In Yang, Q. and Wooldridge, M., editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, International Joint Conference on Artificial Intelligence (IJCAI), pages 2270–2276. AAAI Press/International Joint Conferences on Artificial Intelligence. 24th International Joint Conference on Artificial Intelligence, IJCAI 2015 ; Conference date: 25-07-2015 Through 31-07-2015. – Cité page 21.

[Rehurek and Sojka, 2011] Rehurek, R. and Sojka, P. (2011). Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2). – Cité page 23.

[Reimers and Gurevych, 2019] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. – Cité page 45.

[Robinson, 2022] Robinson, D. (2022). *TopicTuner*. https://github.com/drob-xx/TopicTuner. – Cité page 52.

[Röder et al., 2015] Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 399–408, New York, NY, USA. Association for Computing Machinery. – Cité page 39.

[Scialom et al., 2020a] Scialom, T., Dray, P., Lamprier, S., Piwowarski, B., and Staiano, J. (2020a). MLSUM: the multilingual summarization corpus. *CoRR*, abs/2004.14900. – Cité page 31.

[Scialom et al., 2020b] Scialom, T., Dray, P.-A., Lamprier, S., Piwowarski, B., and Staiano, J. (2020b). Mlsum: The multilingual summarization corpus. – Cité page 31.

[Shahnaz et al., 2006] Shahnaz, F., Berry, M. W., Pauca, V., and Plemmons, R. J. (2006). Document clustering using nonnegative matrix factorization. *Inf. Process. Manage.*, 42(2):373–386. – Cité pages 21 et 49.

[Sontag and M Roy, 2009] Sontag, D. and M Roy, D. (2009). Complexity of inference in topic models. in advances in neural information processing: Workshop on applications for topic models: Text and beyond. – Cité page 20.

[Srivastava and Sutton, 2017] Srivastava, A. and Sutton, C. (2017). Autoencoding variational inference for topic models. – Cité page 45.

[Steinkrau et al., 2005] Steinkrau, D., Simard, P. Y., and Buck, I. (2005). Using gpus for machine learning algorithms. In *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, ICDAR '05, page 1115–1119, USA. IEEE Computer Society. – Cité page 21.

[Teh et al., 2004] Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2004). Hierarchical dirichlet processes. – Cité page 20.

[Terpilowski, 2022] Terpilowski, M. (2022). *Bitermplus*. https://github.com/maximtrp/bitermplus. – Cité page 41.

[Terragni et al., 2021] Terragni, S., Fersini, E., Galuzzi, B. G., Tropeano, P., and Candelieri, A. (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*, pages 263–270, Online. Association for Computational Linguistics. – Cité pages 23 et 41.

[Thompson and Mimno, 2020] Thompson, L. and Mimno, D. (2020). Topic modeling with contextualized word representation clusters. *CoRR*, abs/2010.12626. – Cité page 21.

[van der Maaten and Hinton, 2008] van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605. – Cité page 33.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc. – Cité page 21.

[Wang and McCallum, 2006] Wang, X. and McCallum, A. (2006). Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 424–433, New York, NY, USA. Association for Computing Machinery. – Cité page 20.

[Wolf et al., 2019] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771. – Cité pages 31 et 45.

[Yan et al., 2013] Yan, X., Guo, J., Lan, Y., and Cheng, X. (2013). A biterm topic model for short texts. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 1445–1456, New York, NY, USA. Association for Computing Machinery. – Cité pages 20 et 43.

ANNEXE A

DOCUMENTATION

A.1 Liste des abréviations

Abréviation	Forme complète
TEF	Test d'évaluation de français
CCI	Chambre du commerce de l'industrie
QCM	Questionnaire à choix multiples
TF-IDF	Term frequency * Inverse document frequency
PMI	Pointwise mutual information
NPMI	Normalized pointwise mutual information
TD	Topic diversity
ACP	Analyse en composantes principales
t-SNE	t-distributed stochastic neighbor embedding
LDA	Latent dirichlet allocation
BTM	Biterm topic model
CTM	Contextualized topic model
NMF	Non-negative matrix factorization
BERT	Bidirectional encoder representations from transformers
UMAP	Uniform manifold approximation and Projection
HDBSCAN	Hierarchical density-based spatial clustering of applications with noise

A.2 Matériel

Tous les entraînements décrits dans la section « Entraînement » (4) ont été effectué sous le système d'exploitation Windows 10 Professionnel, en utilisant Python dans sa version 3.9. La machine dispose d'un processeur AMD Ryzen 5 3600x, d'une carte graphique NVIDIA 2070SUPER avec 8 gigaoctets de mémoire vidéo et de 32 gigaoctets de mémoire vive.

A.3 LDA

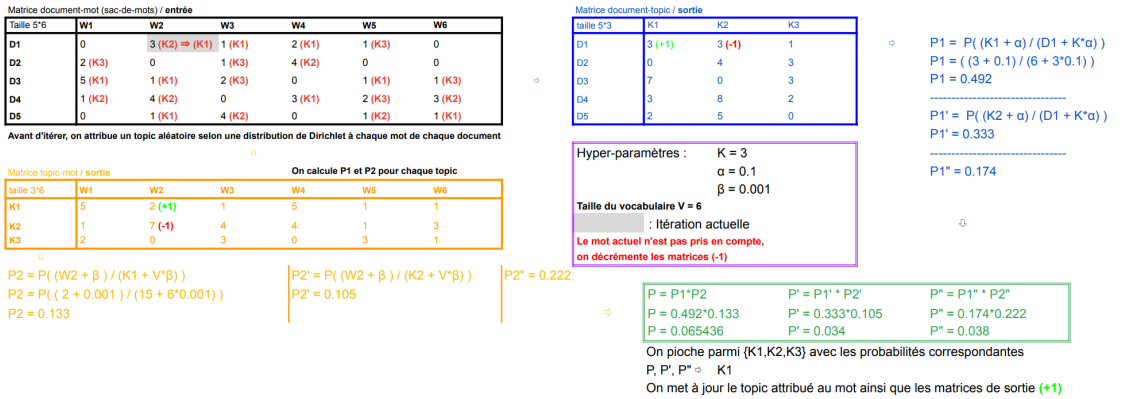


FIGURE A.1 – LDA : Schématisation des entrées, sorties et d'une itération pendant l'entraînement

A.4 Exemples de copies

Les exemples ci-dessous sont des copies normalisées via le module du système de notation automatique mais qui ne sont pas encore pré-traitées pour être utilisées dans nos topic models.

Section A :

Niveau B2 :

Un scandale dans le super marché les clients surpris de voir l'entré d'une biche et qui a trouvé comme refuge, le supermarché en craignant d'être capturé par un groupe des chasseurs Ce groupe de chasseurs n'ont pas eu l'accès dans le super marché pour la capturé. La biche se sent en sécurité après avoir obtenue la protection des clients. Ce groupe désespérer, n'ont pas accomplie leur mission de chasse. Les clients ont mis à l'abri la vie de la biche.

Niveau C1 :

Heureusement que c'est la veille de leurs anniversaire de mariage et comme par hasard aussi c'est exactement le restaurant ou ils ont fait connaissance, le couple on décidé de reporté le dîner et de ne pas aussi changer le lieu a fin de fête leurs date précieuse le monsieur a demandé a la personne qui s'occupe de la relation publique si c'est possible de reporté la réservation pour demain et d'avoir la mÊme la table, et vu que c'est la base saison et le restaurant peu fréquentable la demoiselle leurs a changer la réservation avec toute gentillesse.

Niveau C2 :

La petite fille trouvait qu'il fessait nuit et le ciel unknwn, ses patents ont parti pour lui acheter une classe, au retour un chien attaquait la maman, le mari a essayé de la protéger mais ce dernier a mordu sa femme, Ensuite le papa a pris sa femme a l'hôpital par contre Leila attendait dans le zoo seul avec des animaux, la police trouvait la petite seul au milieu de la nuit qui criait a peut voix enfin ces derniers ont contacté le père de la fille pour la récupérer.

Section B :

Niveau B2 :

Dans la vie et« notre monde » sure tout pour les humain il faut faire une balance pour la nourriture ; dé queue tu est un bébé jusque âgé ; ça veut dire que chaque période il est différent ; en commence a boire du lait «bèbè» et petite a petite en passe au fruit plus des légumes, et comme ça le cours va être santé et les os solide ; ça c'est comme exemple, notre sujet c'est « queue les gens ne savent plus manger sainement » ça veut dire quille faut donner un changement a notre nourriture si on prépare quelque chose a notre cuisine ; pas comme la nourriture de la restaurant ex « macdo» il est a des produit ou on c'est pas de quoi a faire ce sandwich la. Il faut encore manger des légumes différents sure tout « unknwn », la même chose pour les légumes, ça nous donnes des vitamines plus d'énergie queue nos cours a besoin de ça. Il a des inconvénient aussi pour la nourriture, comme cholestérol ça viens de la frais, et des produit sucré qui fait mal aux cours. Si on mange le poisson ça ne donne plus des vitamines « les fruit de mer » ex, il a le blé aussi riche on vitamine, comme ça on donne des exemples l'autre gens

Niveau C1 :

a mon avis et parlant écologiquement avec quelques exemples si on trouve des solution et de ne plus utiliser les choses qui ne sont pas recyclable par exemple les hôtels qui utilisent les vers jetable pourquoi pas ne pas les changer en unknwn, premièrement on diminue les vers en plastique que l'on trouve partout a la mer aussi a unknwn il trouvera l'hôtelier que c'est rentable moins de dépense aussi annuellement, aussi les sacs en plastique réfléchissez si on fait les courses 2 fois par jours on utilises les sachet voyant en un mois combien chaque personnes utilise ? Mais si on achète une fois par mois un sac filet et on sait très bien que le fibre et recyclable comme ça on diminue les sacs en plastique aussi c'est plus rentable comme ça on pourra diminuer le plastique que l'on trouve partout et on pourra garder la nature propre.

Niveau C2 :

Cher Monsieur le rédacteur, Je me permets de vous adresser en tant qu'une lectrice fidèle de votre journal. Mais aujourd'hui je vous écris en tant qu'une mère pour deux enfants. J'ai été surprise par l'affirmation lue dans votre article L'écriture manuscrite disparaît ; bientôt, nos enfants ne sauront plus se servir d'un stylo ! Tout d'abord, au contraire nos enfants devraient utiliser abondement un stylo pour s'exprimer. Ce fait

les aide à bien mémoriser leurs études, En écrivant ses pensées, ils seront mémoriser dans leur cerveau. De plus, l'écriture aidera le fonctionnement normal des yeux, les enfants ne seront pas bloqués toute la journée devant un écran ce qui va leur causer mal aux yeux. Les études scientifiques ont indiqués que se concentrer une longue durée devant un écran causera du mal aux yeux. Aussi, inscrire nos notes et nos pensés sur un papier nous aide à bien réfléchir et à avoir une logique comme vous faites en tant que journalier avant de rédiger un article. En effet, j'ai éduqué mes enfants d'inscrire sur une feuille leurs études avant leurs examens afin de facile la mémorisation. J'ai été ravie de vous faire part de mon avis et j'espère que vous accepterez de le publier dans votre journal. Je vous prie d'agréer, Monsieur le rédacteur, l'expression de mes sincères salutations. Jean

A.5 Exemples de lexiques finaux

Section A :

— *classement, âge, vidéos, application, garçon, génie, jeux, informatique, réseaux, sociaux, adolescent, youtube, millions, secret, instagram, talent, entier, parents, devenu, programmation, doué, jeunes, abonnés, domaine, entrer, vidéo, influence, école*

— *prison, magasin, bijouterie, prisonnier, bijoux, voleur, propriétaire, boutique, sac, billets, banque, voleurs, argent, vitrine, fuite, coincé, policier, sécurité, évasion, vol, voler, sport, enquête, arme, braquage, caisse, plan, employée, individu*

— *loterie, ticket, organisateur, couple, gagnant, euros, millions, dollars, argent, somme, gagné, vérification, gain, échange, proposé, espagne, gagner, date, verser, proposition, jeux, numéro, présenté, cagnotte, délai*

Section B :

— *planète, pollution, avion, produits, magasins, écologie, manger, consommation, vêtements, déchets, repas, limiter, surconsommation, commerces, voyages, climat, transport, viande, acheter, véhicules, internationaux, neufs, aliments, alimentation, écologique, alimentaires, ouverture, climatique, gaspillage*

— *commun, gratuité, transport, imposer, gratuit, bus, payer, service, voiture, déplacer, décision, véhicules, gratuite, coût, carburant, budget, déplacement, taxes, trains, frais, tarifs, charges, revenu, impôts, villes*

— *langue, maternelle, étrangère, renier, apprentissage, correctement, anglais, apprenons, communiquer, cultures, communication, étranger, origines, maîtrise, identité, maîtriser, ouverture, française, mondialisation, origine, parlent, échanges, appris, apprenant, apprend, cerveau*

A.6 Récapitulatif des résultats

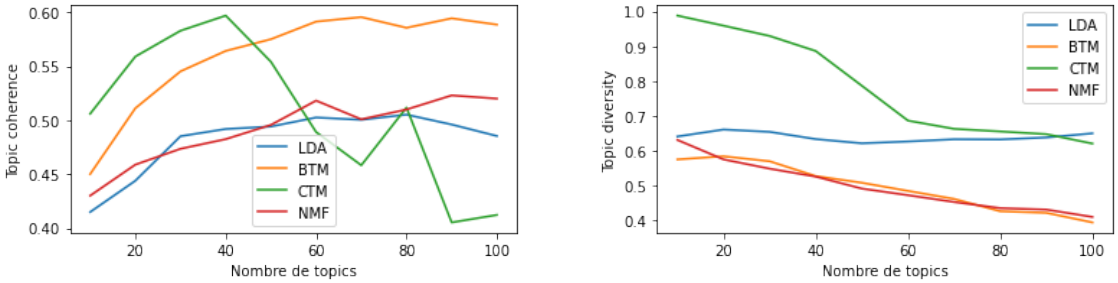
Résultats sur la moyenne de 5 exécutions arrondis à deux décimales.

Section	A				B				AB			
K\Model	**LDA**	**BTM**	**CTM**	**NMF**	**LDA**	**BTM**	**CTM**	**NMF**	**LDA**	**BTM**	**CTM**	**NMF**
10	0.41	0.45	0.51	0.43	**0.45**	0.54	0.62	0.50	**0.46**	0.50	0.56	0.52
20	0.44	0.51	0.56	0.46	0.49	0.54	0.63	0.50	0.49	0.53	0.62	0.51
30	0.49	0.55	0.58	0.47	0.49	0.56	0.64	0.49	0.50	0.53	0.62	0.51
40	0.49	0.56	**0.60**	0.48	0.52	0.56	**0.67**	0.50	0.50	0.54	0.60	0.52
50	0.49	0.58	0.55	0.50	0.52	0.56	0.65	0.51	0.50	0.55	0.60	0.53
60	0.50	0.59	0.49	0.52	0.54	0.56	0.64	0.51	0.52	0.55	0.62	0.53
70	0.50	0.60	0.46	0.50	0.53	0.57	0.64	0.52	0.50	0.55	0.62	0.54
80	0.50	0.59	0.52	0.51	0.52	0.56	0.64	0.52	0.50	0.55	0.62	0.54
90	0.50	0.59	**0.40**	0.52	0.51	0.56	0.57	0.52	0.50	0.55	0.62	0.54
100	0.49	0.59	0.41	0.52	0.51	0.56	0.62	0.52	0.49	0.55	**0.62**	0.54

TABLE A.1 – Moyenne de la Cv-cohérence par section et modèle selon nombre de topics

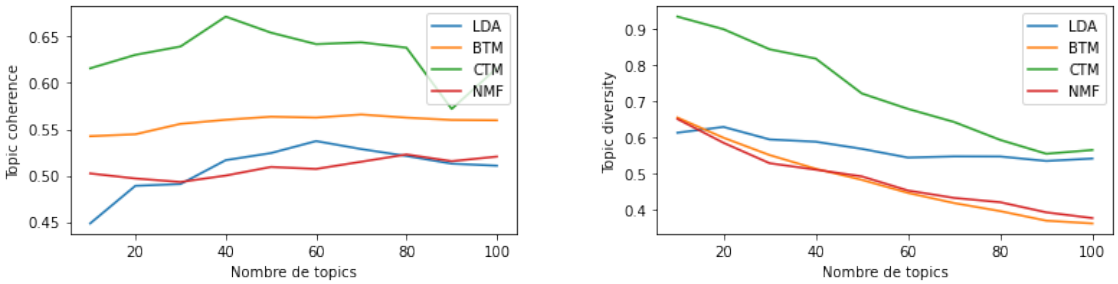
Section	A				B				AB			
K\Model	**LDA**	**BTM**	**CTM**	**NMF**	**LDA**	**BTM**	**CTM**	**NMF**	**LDA**	**BTM**	**CTM**	**NMF**
10	0.64	0.57	**0.99**	0.63	0.61	0.65	**0.93**	0.65	0.73	0.67	0.88	0.69
20	0.66	0.58	0.96	0.57	0.63	0.60	0.90	0.58	0.71	0.59	0.90	0.62
30	0.65	0.57	0.93	0.55	0.60	0.55	0.84	0.53	0.72	0.55	**0.93**	0.57
40	0.63	0.53	0.89	0.52	0.59	0.51	0.82	0.51	0.72	0.53	0.91	0.55
50	0.62	0.51	0.79	0.49	0.57	0.48	0.72	0.49	0.72	0.50	0.86	0.53
60	0.63	0.48	0.69	0.47	0.54	0.45	0.68	0.45	0.70	0.48	0.83	0.51
70	0.63	0.46	0.66	0.45	0.55	0.42	0.64	0.43	0.72	0.46	0.79	0.50
80	0.63	0.42	0.65	0.43	0.55	0.39	0.59	0.42	0.71	0.44	0.77	0.48
90	0.64	0.42	0.65	0.43	0.53	0.37	0.55	0.39	0.73	0.43	0.76	0.46
100	0.65	**0.39**	0.62	0.41	0.54	**0.36**	0.56	0.38	0.72	**0.41**	0.69	0.45

TABLE A.2 – Moyenne de la topic diversity par section et modèle selon nombre de topics



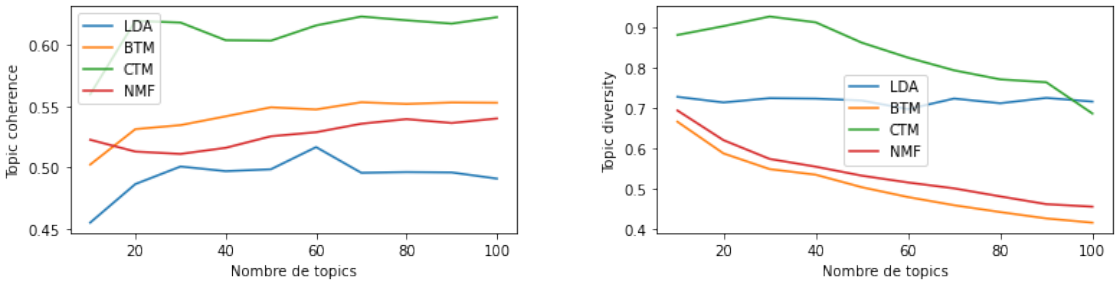
(a) Cv-cohérence (b) Topic diversity

FIGURE A.2 – Moyenne de la Cv-cohérence et de la topic diversity par modèle selon nombre de topics, section A



(a) Cv-cohérence (b) Topic diversity

FIGURE A.3 – Moyenne de la Cv-cohérence et de la topic diversity par modèle selon nombre de topics, section B



(a) Cv-cohérence (b) Topic diversity

FIGURE A.4 – Moyenne de la Cv-cohérence et de la topic diversity par modèle selon nombre de topics, section AB

INDEX