Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

Classification automatique de documents : application aux exercices de manuels scolaires

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Elise LINCKER

Directeur de mémoire :

Cyril Grouin

Encadrants :

Camille Guinaudeau, Olivier Pons, Caroline Huron

Année universitaire 2021/2022

TABLE DES MATIÈRES

Liste des figures

Liste des tableaux

LISTINGS

AVANT-PROPOS

Remerciements

Je remercie mes encadrants Camille GUINAUDEAU, Olivier PONS et Caroline HURON pour leur confiance et leur accompagnement bienveillant tout au long de mon stage. Je suis fière de continuer mon parcours au sein de leur équipe dans le cadre d'une thèse sur le projet MALIN.

Je remercie l'ensemble de l'équipe enseignante du master TAL, particulièrement Cyril Grouin pour son suivi, ses conseils et ses encouragements.

Je tiens également à remercier toute l'équipe du projet MALIN et les membres du Cartable Fantastique pour leur implication, mes camarades de master, et mes collègues TAListes du LISN.

Résumé

Dans une démarche d'inclusion scolaire, le projet de recherche MALIN (MAnuels scoLaires INclusifs) a pour objectif l'automatisation de l'adaptation des manuels scolaires numériques pour les rendre accessibles (accès, traitement et interaction avec les contenus) aux élèves en situation de handicap.

Ce mémoire s'inscrit dans le projet MALIN et se focalise sur l'adaptation de manuels de français de niveau élémentaire pour des élèves dyspraxiques. La première partie de ce travail porte sur la classification des exercices selon leur type d'adaptation. En raison d'un fort déséquilibre des classes dans le jeu de données, la deuxième partie traite de la génération de données artificielles. Diverses approches de classification et de génération sont expérimentées et discutées. Les résultats obtenus sont très encourageants, malgré des données multimodales peu étudiées et présentant une structure et un langage qui leur sont propres.

Mots clés :
Classification, génération automatique de textes, apprentissage automatique, manuel scolaire, dyspraxie

Introduction

Si le manuel scolaire est un outil quasi-systématiquement utilisé en classe depuis toujours, il ne convient pas à tous les élèves. Cela se vérifie particulièrement auprès des enfants dyspraxiques car ils ont besoin d'aménagements pour pouvoir contourner les difficultés liées à leur pathologie. La dyspraxie est un trouble développemental de la coordination trop souvent minimisé, bien qu'elle touche plus de 5% des enfants.

Dans une démarche d'inclusion scolaire, nous travaillons avec l'association *Le Cartable Fantastique* sur le projet MALIN (MAnuels scoLaires INclusifs). Dans l'objectif d'automatiser l'adaptation des manuels scolaires, une chaîne de traitement est mise au point. Ce mémoire traite particulièrement de la classification d'exercices de manuels de français selon leur adaptation à la dyspraxie. Cette tâche constitue un challenge de par la source de données, encore trop peu étudiée en traitement automatique des langues, malgré un besoin important. En effet, les manuels scolaires sont une source de données multimodale, composée de texte et d'images. Ils présentent un langage, une structure et une mise en page spécifiques, bien que la forme et le contenu des manuels varient d'un ouvrage à l'autre.

Ce travail détaille le processus de classification, de la construction d'un corpus d'exercices à l'entraînement et l'application de modèles d'apprentissage automatique. Plusieurs méthodes statistiques et neuronales seront testées et discutées. Pour contourner les difficultés causées par le manque de données, une augmentation de données sera également envisagée.

Plan de lecture

L'ensemble du travail est organisé comme suit :

— Dans le chapitre 1, intitulé « Cadre de l'étude », nous exposons le contexte général de l'étude. Après une présentation de la dyspraxie, nous nous intéressons aux aménagements et adaptations déjà proposés pour l'inclusion scolaire d'élèves présentant des troubles des apprentissages. Enfin, nous introduisons le projet de recherche dans lequel s'inscrit notre travail et nos objectifs.

— Dans le chapitre 2, nous présentons un état de l'art des travaux de simplification de textes dans un objectif d'accessibilité, des algorithmes de classification automatique de textes et des méthodes de génération de données.

— Dans le chapitre 3, nous décrivons le processus de construction de notre corpus, à partir des manuels scolaires numériques fournis par les éditeurs, jusqu'au jeu de données final d'exercices annotés.

— Le chapitre 4, intitulé « Protocole expérimental » résume la chaîne de traitement appliquée à notre corpus et décrit en détails les mesures d'évaluation utilisées pour l'évaluation de nos modèles.

— Dans le chapitre 5, nous présentons l'ensemble des expérimentations mises en œuvres pour la classification des exercices et les résultats.

— Le chapitre 6 est dédié aux conclusions et perspectives de ces travaux.

Première partie

Contexte général

CHAPITRE 1

CADRE DE L'ÉTUDE

Sommaire

Introduction

Le travail de recherche présenté dans ce mémoire porte sur un cas particulier de classification de documents : la classification d'exercices de manuels scolaires selon leur adaptation pour les élèves dyspraxiques. Il a été réalisé à l'occasion d'un stage dans le cadre du projet MAnuels scoLaires INclusifs (MALIN), au Laboratoire Interdisciplinaire des Sciences du Numérique (LISN) et en collaboration avec l'association *Le Cartable Fantastique*.

Ce chapitre commence par une introduction à la dyspraxie. La section 1.2 se constitue comme un panorama des outils d'accessibilité pour les dys actuellement existants. Enfin, nous présentons, dans la section 1.3, l'association et le projet dans lequel s'inscrit notre travail, et dans la section 1.4, la chaîne de traitement mise en place pour le projet et les objectifs que nous nous sommes fixés.

1.1 Dyspraxie

La dyspraxie est un trouble développemental de la coordination. Mal connue en France, elle est pourtant aussi fréquente que la dyslexie et touche 5% des enfants.

La dyspraxie se traduit par un déficit de la coordination motrice : absence d'automatisation du geste, troubles de l'organisation du regard et de la perception de l'espace. Les enfants dyspraxiques se trouvent ainsi en difficulté dans leur vie quotidienne et scolaire. Ils ont du mal à gérer leur temps et leur matériel, à s'habiller et à lacer leurs chaussures, à manier des couverts, à faire du vélo et à attraper un ballon. A l'école, ils sont forcément amenés à découper, coller, dessiner et surtout écrire, ce qui leur demande un effort considérable par rapport à la plupart des enfants de leur âge. L'écriture manuscrite est basée sur des gestes complexes que les enfants dyspraxiques ne parviennent pas à coordonner et automatiser. Ils ne peuvent pas réaliser plusieurs tâches à la fois : le traçage de chaque lettre leur demande une grande concentration, au détriment de l'attention et des apprentissages. Si les élèves

dyspraxiques ne sont pas accompagnés, leur trouble les empêche d'apprendre et de suivre le rythme scolaire.

Il faut comprendre qu'un élève dyspraxique est compétent, intelligent et peut réussir à l'école. Cela implique de mettre en place des aménagements et des adaptations pour contourner les difficultés liées à son trouble, en premier lieu l'écriture manuscrite.

Caroline Huron, chercheuse en sciences cognitives et présidente de l'association *Le Cartable Fantastique*, explique davantage les enjeux de l'aide à l'enfant dyspraxique dans [Huron, 2017] et dans son ouvrage [Huron, 2011].

1.2 Accessibilité

Tous les élèves atteints d'un trouble de l'apprentissage nécessitent des aménagements et des adaptations pédagogiques à l'école, pour leur permettre d'apprendre au même rythme que leurs camarades. Pour la dyspraxie, cela consiste principalement à réduire les efforts moteurs, dont l'écriture, en privilégiant l'oral et en introduisant des outils numériques. Les documents de travail doivent être non seulement préparés pour faciliter la concentration de l'enfant sur la tâche, mais aussi aérés, clairement structurés, et suivre des normes de repérage.

Des logiciels, des applications et des éditeurs de livres numériques permettent de mettre en place certaines compensations pour les enfants souffrant de troubles dys-. Il est possible de zoomer sur certains éléments, modifier la police, la taille de la police, les couleurs, ou encore les espaces et les interlignes. Les éditeurs de texte grands publics proposent également des fonctionnalités de lecture. Par exemple, l'extension d'aide à la lecture « Lire Couleur » d'Open Office permet de colorer les lettres par son, d'atténuer les lettres muettes, de mettre en évidence les syllabes ou encore de surligner une ligne sur deux. L'association *Le Cartable Fantastique* propose quant à elle des outils numériques spécifiquement adaptés à la dyspraxie, dont un plug-in Libre Office. Une panoplie d'options sont disponibles, pour la lecture et l'écriture ainsi que pour chaque matière scolaire comme les mathématiques, la physique et la chimie. On y retrouve des options de pose d'opérations mathématiques, de tracé de tableaux et de frises chronologiques, ou encore de réalisation de circuits électriques.

Certains éditeurs comme Belin publient des cahiers d'exercices interactifs avec une police spécialement étudiée pour les dyslexiques (Open Dyslexic) et un contraste réduit. L'association *Le Cartable Fantastique* crée également les *Fantastiques Exercices*, un ensemble d'exercices adaptés accompagnés de leur version classique pour le reste de la classe. Plus spécifiquement pour la lecture, un corpus parallèle de textes de niveaux CE1, CE2 et CM1 a été construit dans le cadre du projet ALECTOR (Aide à la LECTure pour améliORer l'accès aux documents pour enfants dyslexiques) [Gala et al., 2020]. Les textes originaux ont été simplifié au niveau du lexique, de la morpho-syntaxe et du discours. MOBiDYS propose aussi des livres de littérature étudiés en classe au format « FROG », un format numérique conçu pour les élèves dyslexiques, ainsi que « DAISY », des versions audio des manuels scolaires traditionnels.

Enfin, des ouvrages destinés aux enseignants, comme *Dys : outils et adaptations dans ma classe - Cycles 2 et 3* [Loty and Mazeau, 2020], les aident à adapter leur contenu pédagogique aux élèves porteurs d'un trouble dys.

S'il existe quelques travaux et outils portant sur la dyslexie et l'aide à la lecture, l'accessibilité à l'école pour les dyspraxiques n'a encore fait l'objet d'aucune étude. Il

existe des guides à destination des enseignants et des parents, mais peu de ressources spécifiquement conçues pour les élèves. Les manuels scolaires ne sont notamment pas accessibles et doivent être remis à des services d'adaptation-transcription ou bien pris en charge par l'enseignant de la classe ou l'accompagnant de l'élève en situation de handicap (AESH), afin de produire des versions numériques adaptées.

1.3 MALIN

L'association *Le Cartable Fantastique* a pour objectif l'aide à l'inclusion scolaire des enfants en situation de handicap, particulièrement les enfants dyspraxiques. Depuis 2010, l'équipe crée des outils numériques à utiliser en classe afin de permettre aux enfants de compenser leur déficit et aux enseignants d'adapter leur contenu. Ces outils permettent aux élèves dyspraxiques de suivre le cours au même rythme et de faire les mêmes exercices que leurs camarades, sans être bloqués par leur handicap.

En 2021, l'association propose le projet MALIN (MAnuels scoLaires INclusifs) financé par l'Agence Nationale de la Recherche (ANR). Le manuel scolaire est un support pédagogique quasi systématiquement utilisé en classe, mais il n'existe actuellement aucune version numérisée accessible aux enfants qui ont des difficultés pour écrire. Une adaptation tenant compte de leurs besoins est nécessaire. Dans un manuel adapté, les exercices sont modifiés de manière à contourner l'écriture manuscrite mais sans en changer le contenu ni l'objectif pédagogique.
Exemple : La consigne « *Recopie chaque liste sans l'intrus.* » devient « *Dans chaque liste, cache l'intrus.* », et l'élève doit cliquer sur un élément de l'énoncé.

Jusqu'à présent, seules des parties d'ouvrages sont adaptées manuellement sur une plateforme élaborée par l'association, et les délais d'adaptation peuvent être de plusieurs mois. Le projet de recherche MALIN a donc pour objectif de développer des solutions techniques et innovantes afin d'aboutir à l'automatisation de l'adaptation des manuels scolaires numériques pour les rendre accessibles aux élèves en situation de handicap.

1.4 Objectifs

Notre travail s'inscrit dans le projet MALIN et consiste à innover pour automatiser l'adaptation des manuels scolaires. Au sein d'une équipe de quatre stagiaires menés par des chercheurs en informatique, traitement automatique du langage naturel et sciences cognitives, nous nous penchons dans un premier temps sur l'adaptation de manuels de français de classes élémentaires pour les élèves dyspraxiques.

Il s'agit de mettre en place une chaîne de traitement qui, en partant d'un manuel numérique fourni par son éditeur, renvoie sa version interactive et adaptée. Le processus d'adaptation imaginé par l'équipe, schématisé en figure 1.1, comprend les étapes suivantes :

- **Conversion** du manuel en un document structuré (XML) ;
- **Extraction** des composantes pragmatiques du manuel : cours et exercices - et au sein de chaque exercice : numéro ou nom d'exercice, consigne, énoncé, exemple, conseil, images ;
- **Classification** des exercices selon leur type d'adaptation ;
- **Adaptation** des exercices selon leur annotation ;
- **Fusion** des exercices adaptés et éventuellement d'autres éléments du manuel (i.e. fiches de cours, etc.) en un unique document HTML.

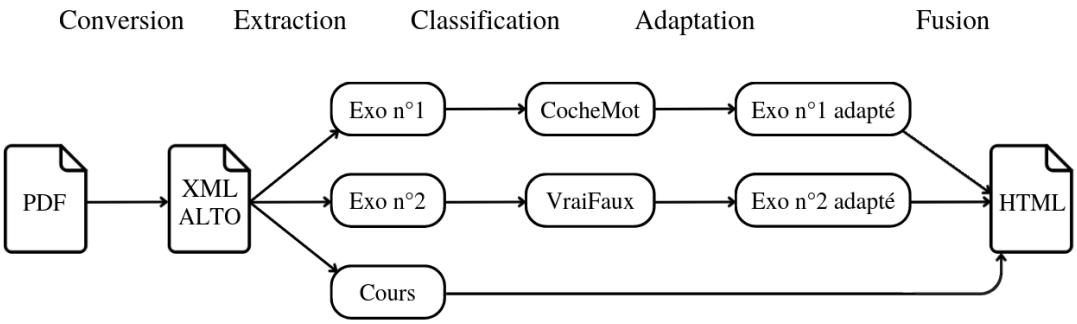


FIGURE 1.1 – Processus d'adaptation

Le travail présenté dans ce mémoire porte sur la **classification des exercices en fonction de leur adaptation**. Il sera l'occasion non seulement d'appliquer des algorithmes de classification existants à un nouveau type de données et d'évaluer ainsi leur robustesse, mais aussi de proposer de nouvelles méthodes et d'emprunter d'autres pistes pouvant être bénéfiques à ce projet. En particulier, une **augmentation des données d'apprentissage** sera envisagée, étant donné le manque d'études similaires et donc le manque de données.

CHAPITRE 2

ETAT DE L'ART

Sommaire

Introduction

Dans ce chapitre, nous nous intéressons d'abord aux travaux menés sur la simplification de textes dans un objectif d'accessibilité. Ensuite, nous introduisons les deux tâches de traitement automatique des langues traitées dans cette étude. D'une part, il s'agit de la classification de documents : dans la section 2.2, nous présentons un inventaire des méthodes de représentation de documents ainsi qu'un état de l'art des algorithmes de classification automatique de textes. D'autre part, nous présentons dans la section 2.3 un état de l'art des techniques de génération de données artificielles.

2.1 Simplification de textes

La simplification de textes a déjà fait l'objet de plusieurs études, [Siddharthan, 2014, Alva-Manchego et al., 2020] en font un inventaire. Certains chercheurs reprennent des méthodes utilisées dans d'autres domaines comme la traduction automatique [Wubben et al., 2012], d'autres se focalisent sur un niveau linguistique précis comme la syntaxe [Brouwers et al., 2014]. D'autres travaux portent sur la simplification de textes spécialisés, par exemple du domaine médical [Cardon and Grabar, 2020] ou juridique [Cemri et al., 2022], pour une accessibilité au grand public.

La simplification automatique de textes est aussi étudiée pour faciliter la lecture et la compréhension de textes pour des enfants, des personnes présentant un handicap ou un trouble mental, ou encore les apprenants d'une langue étrangère.

Par exemple, [Canning and Tait, 1999] et [Carroll et al., 1999] développent, à partir d'une approche par règles, des systèmes de simplification de textes pour des personnes présentant des troubles du langage ou des apprenants de l'anglais. Ils appliquent des analyseurs grammaticaux et syntaxiques, puis proposent des opérations de simplification comme la substitution lexicale, la résolution d'anaphores et la simplification syntaxique. Concernant l'apprentissage à l'école, [Scarton and Specia, 2018] construisent des modèles séquence à séquence pour simplifier des textes en fonction du niveau scolaire.

Des travaux similaires ont également été menés sur la langue française. Le FALC (Facile A Lire et à Comprendre) est une écriture utilisée pour simplifier des documents à destination de personnes ayant une déficience intellectuelle. Elle repose sur un ensemble de recommandations définies dans [AUDIAU, 2009]. Le projet Cap'FALC a ainsi pour but le développement d'outils numériques de transcription de textes en FALC. Dans le cadre de ce projet de recherche, [Martin et al., 2020a] proposent une méthode de simplification de phrases paramétrable par l'utilisateur, reposant sur un modèle d'apprentissage profond de type séquence à séquence. Peuvent être spécifiés la longueur des phrases, ou encore le degré de complexité lexicale ou syntaxique.

Dans le même objectif d'accessibilité, l'outil HECTOR (Hybrid tExt simplifiCation TOol for Raw texts in French) [Todirascu et al., 2022] est créé dans le cadre du projet ALECTOR (Aide à la LECTure pour améliORer l'accès aux documents pour enfants dyslexiques). HECTOR simplifie les textes aux niveaux lexical, syntaxique et discursif. Le système est conçu pour des personnes dyslexiques, mais peut servir de base pour une adaptation à un autre public.

Si ces travaux traitent l'accessibilité des textes, ils ne sont pas directement liés à notre travail d'adaptation des manuels. Le but n'est pas de simplifier les textes, mais de les transposer. Dans la phase d'adaptation et suivant les directives du projet MALIN, il s'agit effectivement d'apporter quelques modifications aux exercices pour permettre à l'élève de se repérer dans l'espace et de réduire ses gestes moteurs. Des études encore en cours et des expériences réalisées sur le public dyspraxique montrent par exemple l'influence de l'utilisation des couleurs pour le repérage spatial. Cependant, le contenu et l'objectif pédagogique de l'exercice ne sont en aucun cas altérés.

2.2 Classification de documents

La classification de textes est une tâche supervisée qui consiste à assigner une ou plusieurs étiquettes (ou catégories ou classes) prédéfinies à des documents. Elle peut s'appliquer à tous types de textes et répondre à plusieurs objectifs, comme l'analyse de sentiments, la classification en thème, ou la détection de spams.

Dans cette section, nous exposons un état de l'art des algorithmes de classification automatique de textes. Il s'appuie essentiellement sur les articles [Kowsari et al., 2019, Minaee et al., 2021, Gasparetto et al., 2022] qui se présentent comme des panoramas des méthodes de classification automatique de texte existantes. Ils décrivent les approches, les pré-traitements, les méthodes d'évaluation et les limites de chaque technique et de ses applications.

2.2.1 Pré-traitements

Tous décrivent l'importance des pré-traitements et de la représentation du texte dans un système de classification. [Kowsari et al., 2019] découpent la tâche de classification en quatre phases : l'extraction de caractéristiques, la réduction de dimensionnalité, le choix du classifieur adéquat et l'évaluation. Ils appuient particulièrement sur la première étape. Cela commence par l'application d'opérations de pré-traitements pour nettoyer et normaliser les données de sorte à réduire le bruit.

Les pré-traitements les plus appliqués dans le traitement de données textuelles sont la normalisation, la tokenisation, la lemmatisation, la racinisation et la suppression des mots vides.

La tokenisation est une opération fondamentale qui consiste à découper une chaîne de caractères en unités lexicales appelées tokens.

La normalisation varie selon la tâche et le corpus. Elle peut par exemple inclure la suppression des caractères d'espacement en début et fin de chaîne, la réduction des espaces multiples consécutives, la suppression ou l'ajout d'espaces autour des signes de ponctuation ou des caractères spéciaux, et la transformation du texte en minuscules ou en majuscules.

La lemmatisation consiste à rapporter les différentes formes d'un même mot à une forme unique : le lemme. En français, il s'agit généralement de l'infinitif pour les verbes, du singulier pour les noms communs, du masculin singulier pour les adjectifs et de la forme non élidée pour les formes élidées.

La racinisation consiste à rapporter les mots à leur racine, en supprimant les affixes et les désinences.

La suppression des mots vides (stopwords) repose sur la suppression des mots du texte qui ne portent pas d'information sémantique.

Il est aussi courant d'expanser les abréviations, de réécrire les expressions argotiques ou familières en langage formel, et d'appliquer au document une correction orthographique.

Les opérations de pré-traitements varient selon les particularités du type de documents du jeu de données et surtout selon la technique d'apprentissage utilisée. Les modèles neuronaux incluent généralement des pré-traitements dans leur chaîne de traitement, comme la normalisation en minuscule, la suppression de caractères spéciaux et la tokenisation. Les documents tokenisés sont ensuite normalisés à la même taille, en ajoutant du padding ou par troncation.

2.2.2 Représentation du document

L'information textuelle, contrairement aux images, ne possède pas intrinsèquement une représentation numérique. Après pré-traitement du texte, l'extraction de caractéristiques permet une représentation des données interprétable par les machines. Cela peut se faire de différentes manières :

- représenter les liens syntaxiques entre les mots :
  - n-grams ;
  - analyse syntaxique en dépendances ;
- donner du poids aux mots (*weighted words*) :
  - sac de mots ;
  - TF-IDF ;
- plongements de mots :

— non contextualisés ;
— contextualisés ;

Ces méthodes sont présentées dans les sous-sections suivantes.

Représentation par sac de mots

La représentation sac de mots est la méthode de représentation des mots d'un document la plus simple. Elle consiste à calculer la fréquence des mots au sein de chaque document. Soit $W = \{w_1, ...w_n\}$ l'ensemble des mots du corpus, un document est représenté par le vecteur $d = \{d_1, ...d_n\}$, avec d_i le nombre d'occurrences du mot w_i qu'il contient.

Cette représentation encode chacun des mots du vocabulaire en un vecteur one-hot, dont la dimension est la taille du vocabulaire de l'ensemble du corpus. Ce nombre est généralement très important, le vocabulaire pouvant compter plusieurs millions de mots. Les vecteurs des documents sont souvent constitués de zéros, surtout si les documents du corpus sont courts. Les sacs de mots sont généralement des ensembles de grande dimension et peuvent ainsi poser des problèmes d'efficacité de par leur dimension et leur sparcité.

[Jones, 2004] propose la fréquence inverse du document (IDF) comme un complément du modèle sac de mots. L'idée est de trouver une formule qui donne moins de poids aux mots trop fréquents et plus de poids aux termes clés peu fréquents. La combinaison de la fréquence des mots (TF) et de la fréquence inverse du document (IDF) amène au TF-IDF dont la formule mathématique est :

$$tf - idf_{w_i,d} = TF(w_i, d) * IDF(w_i)$$

avec $TF(w_i, d)$ le nombre de fois où le mot w_i apparaît dans le document d, et $IDF(w_i)$ la fréquence inverse du document. Celle-ci est calculée selon l'équation :

$$IDF(w_i) = \log \frac{|D|}{DF(w_i)}$$

avec $|D|$ le nombre total de documents dans le corpus et $DF(w_i)$ le nombre de documents dans lesquels le mot w_i apparaît.

De la même manière que dans une représentation sac de mots, la TF d'un mot w_i dans un document d est élevée si w_i est présent fréquemment à l'intérieur de celui-ci. Finalement, l'IDF est une métrique de pondération : les mots qui apparaissent dans de nombreux documents du corpus sont des termes moins importants du fait de leur faible IDF.

Plongements de mots non contextualisés

Les méthodes de représentation des mots présentées ci-dessus produisent des matrices très grandes et de faible densité et ne prennent en compte ni l'ordre des mots, ni les relations syntaxiques et sémantiques entre les mots. Pour résoudre ces problèmes, de nombreux chercheurs ont travaillé sur les plongements de mots, aussi appelés plongements lexicaux ou word embeddings.

A partir d'un grand corpus de textes, un algorithme apprend les plongements en projetant les vecteurs dans l'espace de manière à rapprocher les mots sémantiquement proches. Un mot est représenté par un vecteur numérique dense de grande

dimension. Avec ce type de représentation, deux mots apparaissant dans un contexte similaire sont représentés par deux vecteurs similaires.

Les premières solutions ainsi proposées sont les plongements de mots dits non contextualisés. Les modèles d'apprentissage de tels plongements lexicaux les plus connus sont Word2vec [Mikolov et al., 2013], GloVe (Global Vectors) [Pennington et al., 2014] et FastText [Bojanowski et al., 2017].

L'algorithme non supervisé Word2vec repose sur trois couches de neurones. Il produit, pour un corpus, un ensemble de vecteurs caractéristiques qui représentent les mots de ce corpus. L'algorithme encode chaque mot en un vecteur en construisant une fenêtre de taille n, généralement $n = 4$ ou $n = 5$, et entraîne les mots contre d'autres par itération. Il existe deux variantes de Word2vec : la première, CBOW (continuous bag of words), qui consiste à prédire un mot étant donné son contexte, et la deuxième, appelée skip-gram, qui consiste de prédire un contexte étant donné un mot. L'apprentissage d'un modèle CBOW est plus rapide, mais skip-gram produit généralement des résultats plus précis.

Le modèle GloVe est similaire à Word2vec. Chaque mot est représenté par un vecteur de grande taille et entraîné à partir d'un grand corpus sur les mots voisins dans une fenêtre contextuelle. En pratique, ces deux modèles produisent des résultats similaires pour la plupart des tâches de traitement automatique des langues. Leur différence majeure est l'algorithme d'apprentissage. GloVe construit une matrice de co-occurrences mot-mot à partir du vocabulaire du corpus et se base sur les statistiques de co-occurrences globales.

Si ces deux méthodes considèrent le mot comme étant la plus petite unité, l'approche FastText se présente comme une extension de Word2vec qui repose sur les n-grams de caractères. Un mot est alors représenté par un sac de n-grams et son vecteur correspond à la somme des vecteurs de ses n-grams. En particulier, FastText génère de meilleurs plongements pour des mots rares ou non rencontrés dans la phase d'entraînement, grâce aux n-grams partagés avec d'autres mots.

Plongements de mots contextualisés

Les plongements de mots contextualisés sont une autre méthode de plongements qui tient compte de la relation d'ordre des mots dans le document et du contexte. [Peters et al., 2018] s'appuient sur la technique context2vec de [Melamud et al., 2016] pour créer un modèle de plongements de mots contextuels profonds. ELMo (Embeddings from Language Models) produit, à travers un modèle de langue bidirectionnel profond (biLM), des vecteurs modélisant à la fois les caractéristiques complexes sémantiques et syntaxiques des mots, et les variations en fonction du contexte linguistique ; cela permet de modéliser la polysémie. Un même mot peut ainsi avoir différents vecteurs, chacun représentant un contexte différent.

BERT [Devlin et al., 2019] est un autre modèle de langage contextualisé qui représente les mots par des vecteurs numériques de dimension 768. BERT utilise une architecture de type transformer (détails en section 2.2.4) et repose non pas sur la contextualisation statistique, mais sur le modèle de langue masqué. Cette tâche consiste à remplacer un token dans la phrase de façon aléatoire par un token spécial appelé « masque ». L'apprentissage consiste alors à prédire le mot masqué.

Ainsi, il existe plusieurs approches de représentation et de projection des mots sur un espace vectoriel. L'approche sac de mots est la plus intuitive, mais d'autres

techniques plus évoluées ont permis, grâce aux plongements de mots et aux modèles de langue, de représenter les informations sémantiques, puis de prendre en compte le contexte pour modéliser les divers sens des mots.

Le choix de la technique de représentation des données a un impact sur les résultats ainsi que sur la généralisation du système d'apprentissage. Dans le cadre de ce mémoire, différentes techniques seront mises en œuvre selon l'algorithme de classification utilisé. En l'occurrence, pour les classifieurs statistiques, nous utiliserons l'approche TF-IDF, simple à implémenter et pertinente compte tenu des méthodes de classification statistique. Pour les classifieurs plus complexes de type transformer, nous utiliserons logiquement les plongements de mots appris par le modèle de langue lui-même avec la tâche de modèle de langue masqué.

2.2.3 Classifieurs statistiques

Cette section présente un inventaire des algorithmes de classification classiques utilisés dans ce mémoire. Les algorithmes présentés sont de type Naïve Bayes, Arbre de décision, K-plus-proches-voisins et SVM.

Naïve Bayes

Le classifieur bayésien naïf est un modèle génératif probabiliste. Il repose sur le théorème de Bayes et les statistiques sur les mots présents dans les documents. Appliqué à la tâche de classification, le théorème de Bayes permet de calculer la probabilité qu'un document d appartienne à la classe c : $P(c|d) = \frac{P(d|c)P(c)}{P(d)}$. L'algorithme, dit naïf car les caractéristiques ne sont pas liées entre elles, est simple à implémenter.

Il existe plusieurs variantes du classifieur bayésien. L'algorithme multinomial est généralement utilisé pour la classification de documents. En cas de jeu de données déséquilibré, [Frank and Bouckaert, 2006] ont développé une autre méthode en introduisant une étape de normalisation pour faire face à cette difficulté.

K plus proches voisins

La classification de documents avec l'approche des K plus proches voisins repose sur la proximité entre les documents. Soit un document d, l'algorithme trouve les k documents les plus proches de d parmi les documents du corpus d'apprentissage. La classe prédite est celle qui a le plus de représentants parmi ces voisins.

Arbre de décision et forêt décisionnelle

Ces techniques reposent sur la structure d'un arbre et ont été utilisées avec succès dans des tâches de classification [Safavian and Landgrebe, 1991].

Un arbre de décision peut se modéliser par un ensemble de nœuds symbolisant des propriétés, de branches et de feuilles qui représentent les classes. Le classifieur parcourt ainsi l'arbre depuis sa racine, base ses décisions sur une suite de tests évaluant des propriétés données, jusqu'à rencontrer une feuille qui déterminera la classe du document. L'organisation des nœuds dans l'arbre peut être déterminée par diverses fonctions comme la fonction de Gini ou le modèle statistique de De Mántaras [De Mántaras, 1991].

Une forêt décisionnelle [Ho, 1995] est construite sur un ensemble d'arbres de décision, entraînés sur des propriétés aléatoires.

Machine à vecteurs de support

Les machines à vecteurs de support (SVMs) sont originellement créées pour la classification binaire. L'idée est de projeter les données sur un hyperplan de sorte à les séparer en deux catégories par une frontière. L'apprentissage consiste à trouver la frontière séparatrice optimale qui maximise la distance entre la frontière et les points représentant les données. La recherche de cette frontière s'effectue grâce à des fonctions noyau, qui transforment l'hyperplan en un espace de plus grande dimension dans lequel il existe une séparation des données linéaire. Une tâche de classification en n classes peut être traitée en n classifications binaires (technique All-vs-One) ou en $n(n-1)$ classifications binaires (technique One-vs-One).

D'autres techniques statistiques existent, comme l'algorithme de Rocchio [Rocchio, 1971, Somya et al., 2016] et le boosting [Schapire, 1990, Bloehdorn and Hotho, 2004]. Plus complexes, les champs aléatoires conditionnels [Sutton and McCallum, , Chen et al., 2016] sont des modèles probabilistes qui exploitent l'aspect séquentiel du texte.

Discussion

Des comparaisons des différentes techniques ont permis de cibler les avantages comme les limites de chacune d'elles. Les classifieurs Naïve Bayes sont rapides, simples à implémenter et à entraîner, et performants sur les données textuelles, notamment sur des données hétérogènes. L'algorithme des k plus proches voisins est tout aussi performant et simple à implémenter car il ne procède pas par apprentissage mais repose sur des calculs de distances. La difficulté réside dans le paramétrage de la valeur de k et de la mesure de distance (Cosine, Euclidienne, Manhattan, etc.). L'algorithme est aussi coûteux dans la mémorisation de l'ensemble du corpus d'apprentissage. Les arbres de décision sont les plus simples à interpréter de par leur représentation intuitive sous forme d'arbre. Ce n'est en revanche plus le cas pour les forêts décisionnelles constituées de plusieurs arbres. Ces deux classifieurs sont aussi très sensibles au bruit et enclins au sur-apprentissage. Les plus gros avantages des SVM sont leur capacité à modéliser des frontières non linéaires et leur robustesse face au sur-apprentissage. Toutefois, la complexité en mémoire augmente rapidement et les résultats sont moins transparents que pour les autres algorithmes présentés.

Ainsi, selon la tâche et le corpus utilisé, il sera plus judicieux d'utiliser un algorithme plutôt qu'un autre. En recherche d'information, il s'agit de trouver ceux qui, parmi une large collection de documents, répondent à des critères particuliers. Pour la gestion d'un tel type de données, on utilise la classification de textes, principalement Naïve Bayes, les SVM, les arbres de décision et les k plus proches voisins [Dwivedi and Arya, 2016]. En analyse de sentiments, il s'agit d'associer un document à une émotion, généralement positive, négative ou neutre. Pour ce type de tâche, les approches bayésiennes et les SVM sont essentiellement utilisés [Pang et al., 2002].

Dans le cadre de ce mémoire, nous travaillons également sur une tâche de classification. Nous emploierons ainsi les méthodes bayésiennes et les SVM, généralement robustes sur ce type de tâche, mais également des algorithmes de type Arbre de décision et K plus proches voisins. Cela nous permettra de visualiser les caractéristiques propres à chaque classe apprises par les classifieurs et d'évaluer la robustesse des algorithmes sur un jeu de données hétérogène et encore inexpérimenté.

2.2.4 Classifieurs neuronaux

Les méthodes statistiques, bien que toujours populaires et ayant fait leurs preuves sur de nombreuses tâches de traitement automatique des langues, présentent certaines limites, notamment concernant les caractéristiques conçues manuellement. [Minaee et al., 2021] se concentrent ainsi davantage sur les méthodes d'apprentissage automatique basées sur des réseaux de neurones, qui contournent ce problème par l'auto-extraction des caractéristiques. Les auteurs présentent plus de 150 modèles d'apprentissage profond développés pour diverses tâches de classification de texte, regroupés en catégories selon leur architecture. Dans cette section, nous présentons succinctement les principales catégories de modèles et développons les approches envisagées dans le cadre de ce mémoire.

Réseaux de neurones à propagation avant

Les classifieurs neuronaux les plus simples sont les réseaux de neurones à propagation avant, comme le perceptron multi-couches et le Deep Average Network (DAN) [Iyyer et al., 2015]. Le texte, représenté à partir de plongements de mots, se déplace vers l'avant à travers une ou plusieurs couches cachées.

Réseaux de neurones récurrents

Les réseaux de neurones récurrents (RNN) voient le texte comme une séquence de mots, ils capturent les dépendances entre les mots et la structure textuelle. Pourtant, les RNN basiques ne donnent pas de bons résultats sur la classification. En particulier, leur mémoire est courte, ce qui les empêche de retenir des mots assez éloignés du mot courant dans la phase d'apprentissage. Les variantes alors utilisées sont les LSTM (Long Short-Term Memory) et GRU (Gated Recurrent Unit), conçus pour gérer la mémoire à court et à long terme en conservant les dépendances sur une plus grande étendue dans le texte [Tai et al., 2015, Dieng et al., 2017].

Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN) sont initialement conçus pour le traitement d'images. Ils reposent sur des mécanismes de convolution, consistant à détecter les caractéristiques visuelles en balayant l'image avec des filtres, et de pooling, pour réduire la taille des images de couche en couche tout en conservant les caractéristiques importantes. Les CNN sont également largement appliqués à la classification de texte, pour leur capacité à détecter des patterns [Kalchbrenner et al., 2014, Liu et al., 2017].

Réseaux de neurones avec mécanisme d'attention

L'architecture du réseau de neurones Transformer est d'abord proposée par [Vaswani et al., 2017] dans l'objectif de faire progresser les systèmes de traduction automatique. Elle a ensuite permis des avancées considérables dans pratiquement toutes les tâches de traitement automatique des langues.

Un Transformer est un réseau de neurones de type séquence à séquence (seq2seq) qui suit l'architecture encodeur-décodeur. Il n'utilise aucun réseau récurrent ou convolutionnel et repose uniquement sur le mécanisme d'attention. Les couches

d'auto-attention permettent de garder l'interdépendance des différents mots d'une séquence et proposent ainsi une représentation pertinente de la séquence.

Le Transformer a inspiré des modèles pré-entraînés, comme GPT (Generative Pre-trained Transformer), BERT (Bidirectional Encoder Representations from Transformers) [Devlin et al., 2019], XLNet [Yang et al., 2019] et T5 (Text-To-Text Transfer Transformers) [Raffel et al., 2020]. Les modèles sont pré-entraînés sur des millions de textes et peuvent être fine-tunés sur des tâches spécifiques. Suite au succès de tels modèles, des extensions de BERT sont imaginées dans l'objectif d'améliorer encore ses performances. Par exemple, DistilBERT [Sanh et al., 2019] exploite la technique de distillation des connaissances [Hinton et al., 2014], RoBERTa [Liu et al., 2019] améliore le modèle de base BERT grâce à un pré-entraînement plus long et un volume de données plus important. Spécifiques à la langue française, on retrouve les modèles CamemBERT [Martin et al., 2020b], entraîné sur le corpus OSCAR [1], et FlauBERT [Le et al., 2020], entraîné sur 24 corpus de types divers.

Discussion

Les réseaux de neurones profonds sont une des techniques d'intelligence artificielle les plus puissantes, ils surpassent les méthodes statistiques dans de nombreux cas. Leur architecture permet d'auto-générer les caractéristiques, de traiter des données complexes et de s'adapter à de nouveaux problèmes. Ils présentent cependant certains inconvénients : les réseaux profonds sont vus comme des « boîtes noires », et l'interprétation des phases d'apprentissage et de test est difficile. Aussi, ils requièrent une importante quantité de données d'apprentissage, contrairement aux algorithmes d'apprentissage traditionnels. Ces techniques peuvent alors difficilement être utilisées pour la classification sur un petit jeu de données. De plus, la complexité algorithmique augmente considérablement avec le grand nombre de données. Le choix de l'architecture du réseau et des paramètres de l'entraînement constitue également un grand challenge des méthodes d'apprentissage profond.

[Minaee et al., 2021] comparent les performances de modèles d'apprentissage profond sur des corpus de classification de référence. Dans l'ensemble, les modèles de type Transformer surpassent les baselines statistiques mais aussi les réseaux de neurones récurrents et convolutifs dans toutes les tâches de classification de texte. Plus précisément, BERT et XLNet présentent les plus hauts scores sur la plupart des corpus.

2.2.5 Analyse de structure de documents

Quelques travaux récents portent sur la représentation du langage dans des documents visuellement riches. L'idée est de prendre en compte le document dans son ensemble, en intégrant au modèle des caractéristiques visuelles. Ils sont utilisés par exemple pour la compréhension de formulaires, de factures, les systèmes de question-réponses sur des documents, ou encore pour la classification d'images de documents.

Certaines méthodes se basent sur les modèles de langue, comme LayoutLM [Xu et al., 2020], construit à partir de l'architecture de BERT. Dans le but d'aligner le texte avec la structure visuelle du document, sont ajoutés deux types de plongements supplémentaires. Le premier dénote la position spatiale des tokens sur le document de sorte à capturer la relation entre eux. Le deuxième est un plongement visuel qui

1. https://oscar-corpus.com/

apporte des caractéristiques telles que les types, couleurs et styles des polices de caractères utilisées dans le document. Le modèle est pré-entraîné sur une tâche de Modélisation du Langage Masqué, à partir des plongements de mots et des plongements de position, puis fine-tuné sur diverses tâches avec les trois types de plongements. Dans LayoutLMv2 [Xu et al., 2021], les plongements visuels sont traités dès le pré-entraînement du modèle, qui apprend, grâce à un alignement texte-image, l'interaction intermodale entre l'information visuelle et l'information textuelle. Dans la même direction, TILT [Powalski et al., 2021] utilise une architecture encodeur-décodeur exploitant le texte, les positions et l'image. LayoutLMv2 et TILT présentent des résultats légèrement supérieurs à LayoutLM dans une tâche de compréhension de documents, grâce à l'intégration des plongements visuels dès le pré-entraînement, mais en dépit de la complexité. De tels modèles peuvent alors être entraînés sur une tâche de classification.

D'autres modèles se basent sur des méthodes état de l'art pour la classification d'images, auxquelles sont ajoutées des techniques d'extraction et de représentation du texte. [Jain and Wigington, 2019] et [Bakkali et al., 2020] proposent des méthodes de fusion en passant les images ainsi que l'information textuelle obtenue par océrisation à travers des réseaux de neurones convolutifs. Ces méthodes multimodales dépassent les performances obtenues par les baselines unimodales sur des corpus de classification de référence.

2.3 Génération de données artificielles

La génération de données consiste à créer de nouvelles données artificielles à partir d'un ensemble de données existantes. Elle permet de compléter les données d'entraînement originales, ou de les substituer, par exemple dans un contexte où les données originales ne peuvent être employées pour des raisons de confidentialité. Dans un scénario de complément, cette technique s'est avérée efficace sur de petits jeux de données.

Les données générées sont produites à partir des données originales, par le biais de transformations. Cette transformation peut s'appliquer à l'échelle d'un caractère, d'un mot, d'une phrase ou du document, et s'appuyer sur des règles, des thésaurus, ou des méthodes neuronales. La difficulté de la génération de données dans une tâche de classification de texte réside dans le maintien de la même étiquette. Les informations de la classe doivent être conservées lors de la transformation des données. [Bayer et al., 2022] proposent ainsi une vue d'ensemble sur les différentes méthodes d'augmentation de données textuelles pour la classification.

A l'échelle du caractère, l'injection de bruit est utilisé pour que le modèle soit robuste aux perturbations [Belinkov and Bisk, 2019]. Il consiste en la substitution, l'omission ou l'insertion d'un caractère. D'autres techniques implémentent des transformations à base de règles et reproduisent des fautes d'orthographe et des fautes de frappe, modifient des entités nommées, ou emploient des abréviations et des formes contractées [Coulombe, 2018].

Plusieurs méthodes d'augmentation de données dans des tâches de traitement automatique des langues reposent sur la substitution. Elles utilisent des ressources langagières ou des plongements de mots. Par exemple, [Zhang et al., 2015] utilisent un thésaurus afin de remplacer des mots par un synonyme. La tâche

de substitution lexicale a également fait l'objet des challenges SemEval-2007 [McCarthy and Navigli, 2007] et son équivalent en langue française SemDis-TALN2014 [Fabre et al., 2014]. Parmi les systèmes candidats figure WoDis [Gábor, 2014], les candidats à la substitution sont recherchés dans une base lexicale construite à partir du WOLF (Wordnet Libre du Français [Sagot and Fišer, 2008]) et de la version française de Wikipédia. Ces systèmes comportent ensuite une étape de désambiguïsation pour sélectionner le candidat le plus adéquat compte tenu du contexte. [Kobayashi, 2018] se focalise sur les relations paradigmatiques pour tenir compte du contexte et aller au-delà de la synonymie.

D'autres approches exploitent les modèles de langue comme BERT ou GPT-2 (Generative Pre-Trained Transformers). Ceux-ci sont utilisés soit directement sur une tâche de génération automatique de texte [Kumar et al., 2020], soit pour une modification locale des données à partir de plongements contextuels [Wu et al., 2019].

A l'échelle du document, une autre technique de génération de données repose sur la traduction inversée, ou rétrotraduction [Edunov et al., 2018]. Elle consiste à traduire un document dans une autre langue puis retraduire la traduction dans la langue initiale.

[Luque, 2019] introduit la méthode crossover, inspirée des croisements génétiques. Sur un corpus de tweets, il combine deux moitiés de tweets différents pour en générer un nouveau.

Les performances de plusieurs systèmes sur une tâche de classification sont regroupées dans [Bayer et al., 2022]. Bien que les résultats dépendent en partie des modèles de classification et des types de données, ce comparatif des méthodes confirme l'intérêt de l'augmentation de données : les résultats de la classification sont globalement plus hauts avec des données artificielles. Ceci est particulièrement le cas avec les méthodes de génération, de rétro-traduction et de substitution à partir de modèles de langue, avec lesquelles l'amélioration est quasi-systématique. Les autres méthodes de substitution à l'échelle du caractère ou du mot sont moins coûteuses mais moins sûres, avec davantage de résultats en baisse sur des corpus de référence.

D'autre part, [Claveau et al., 2021] comparent l'impact de l'utilisation de données générées artificiellement dans une tâche de classification de documents. Ils utilisent des classifieurs à base de transformers et d'autres reposant sur des approches sac de mots, et observent les résultats dans des situations de substitution et de complément. Les données artificielles sont générées avec le modèle de langue GPT-2, adapté au type de données. Dans un scénario de complément des données d'apprentissage, l'apport de données supplémentaires améliore la performance des modèles dans toutes les expériences. Le gain est particulièrement important pour les approches sac de mots, plus sensibles. Les chercheurs prouvent également que la qualité des données générées joue fortement sur la performance finale.

Enfin, il est commun d'appliquer plusieurs techniques afin d'obtenir des données plus diversifiées [Hendrycks et al., 2020]. Dans cette optique de diversification des transformations, [Ratner et al., 2017] présentent une méthode de méta-learning pour la classification dans un domaine spécifique. Des réseaux adverses génératifs sont appliqués sur des opérations de transformation. Le système automatise la construction, le paramétrage et l'application des fonctions de transformation.

Conclusion

Les avancées en classification automatique de documents nous permettent un large choix de techniques. Nous partirons de quelques algorithmes traditionnels afin de construire une baseline solide, puis emploierons des méthodes état de l'art de type transformer. Les méthodes purement visuelles ne seront pas abordées dans la mesure où celles-ci se basent uniquement sur l'image au détriment du texte. Le texte est une source de donnée langagière présente dans tous nos documents que nous allons exploiter au maximum.

Pour faire face au déséquilibre entre les classes et au faible nombre d'exercices pour un tiers d'entre elles, nous souhaitons augmenter notre volume de données d'apprentissage. Celui-ci étant initialement très bas, nous pouvons difficilement employer des méthodes de génération automatique de texte. Une augmentation de données devrait toutefois permettre une amélioration des résultats, nous essayerons d'appliquer les techniques de cross-over et de rétro-traduction et proposerons une piste par règles basée sur des ressources lexicales.

Par ailleurs, pour mener à bien le projet MALIN, nous pourrons également nous appuyer sur certaines techniques utilisées en simplification automatique de textes, bien que notre objectif n'est pas de simplifier, mais de transposer les documents.

Deuxième partie

Expérimentations

CHAPITRE 3

CORPUS

Sommaire

Introduction

Pour l'utilisation et la comparaison des techniques, un ensemble de jeux de données ont été construits spécialement pour la tâche de classification de textes. Par exemple, le corpus Amazon Reviews [1], créé pour l'analyse de sentiments, contient des milliers de commentaires des utilisateurs d'Amazon. AG News [2] est un corpus d'articles de journaux annotés en quatre classes et WikiQA [Yang et al., 2015] est un autre corpus de paires de questions-réponses de divers sujets. Notre tâche reposant sur un type de données particulier et encore non étudié, il n'est pas possible d'exploiter les jeux de données existants. Nous construirons entièrement notre propre corpus d'exercices de manuels scolaires, et y appliquerons les algorithmes de classification déjà reconnus par leurs performances sur les corpus de référence.

Ce chapitre, organisé en cinq sections, présente la construction du corpus. Dans la section 3.1, nous présentons les documents originaux et les traitements appliqués pour en obtenir un corpus d'exercices. Dans la section 3.2, nous présentons un descriptif de la tâche d'annotation et des étiquettes utilisées, ainsi que le jeu d'étiquettes finalement employé dans nos expériences. Dans la section 3.3, nous nous intéressons aux résultats d'analyses linguistiques et statistiques appliquées sur le corpus. La section 3.4 décrit le jeu de données finalement utilisé dans nos expériences. Enfin, dans la section 3.5, nous discutons du corpus constitué et ouvrons des perspectives.

3.1 Description et préparation du corpus

Les traitements d'extraction et d'adaptation appliqués aux manuels diffèrent selon la matière et le niveau scolaire. Par exemple, pour les matières scientifiques, il

1. https://www.kaggle.com/datasets/bittlingmayer/amazonreviews
2. https://www.kaggle.com/datasets/amanandrai/ag-news-classification-dataset

faudra prendre en compte des schémas, des formules et des poses d'opération. Nous travaillons dans un premier temps sur les manuels de français de niveau élémentaire : la majorité des traitements repose sur le texte, bien que celui-ci peut aussi être structuré dans un tableau ou accompagné d'images.

Le corpus original est ainsi constitué de manuels scolaires de français de niveau primaire au format PDF, fournis par les maisons d'édition. Nous disposons de 3 manuels extraits et annotés : 2 manuels de CE2 et 1 manuel de CE1.

Des premiers travaux d'extraction de contenu sont effectués sur les manuels scolaires en format pdf, par océrisation et traitements semi-automatiques.

Dans un premier temps, le contenu du document PDF est converti avec l'outil pdfalto en un document XML au format ALTO. En plus du contenu textuel brut représenté de manière structurée, le fichier contient des informations sur les polices et la mise en page, ainsi que les coordonnées spatiales de chaque bloc que constitue un mot, une phrase, ou un élément du manuel.

Grâce à des méthodes par règles basées sur les polices d'écriture et les espacements entre les caractères, le document XML est ensuite découpé en sections (cours, exercice, ou non classé) et en sous-sections (un exercice est divisé en cinq parties : numéro ou nom de l'exercice, consigne, énoncé, exemple, conseil). Chaque exercice dispose d'un identifiant unique constitué du numéro de page et du numéro ou nom de l'exercice. [3]

Nous disposons ainsi d'un document structuré au format XML pour chaque manuel. Comme nous travaillons sur la classification des exercices, nous extrayons de ce fichier chaque balise correspondant à un exercice et ses descendants, que nous stockons dans un nouveau fichier XML intitulé par son identifiant. Un extrait du corpus est présenté en annexe A.1.

A l'aide d'un script qui parcourt ces fichiers, sont extraits le contenu textuel brut de chaque élément de l'exercice (consigne, énoncé, exemple, conseil), ainsi que des informations de style, comme par exemple la présence d'une liste ou le nombre de polices différentes dans l'exercice. Ces données sont stockées dans un fichier tabulaire. Le tableau annexe A.1 regroupe les informations extraites pour un exercice.

D'autre part, pour appliquer des approches d'apprentissage visuelles, il s'agit de récupérer les exercices sous forme d'images. Les documents PDF des manuels sont d'abord convertis en images avec l'outil Ghostscript. Un autre script Python parcourt chaque exercice extrait au format XML, récupère sa page et ses coordonnées spatiales normalisées, puis recadre et enregistre au format PNG la portion de la page correspondant à l'exercice. Les fichiers PNG et JSON correspondant à l'exercice donné en exemple sont présentés en annexes A.2.

Enfin, selon la technique de classification utilisée, d'autres pré-traitements, présentés dans le chapitre 5, seront appliqués sur le texte.

3. Le fruit de ce travail servira également de données d'entraînement à un modèle d'apprentissage automatique pour une tâche d'extraction de contenu de manuels scolaires par classification de tokens. Ceci permettra l'extraction de chaque bloc de leçon et exercice et ainsi l'automatisation de toute la chaîne de traitement du manuel original au manuel adapté.

3.2 Annotation

Les exercices des trois manuels extraits ont été annotés manuellement par l'association *Le Cartable Fantastique*. A chaque exercice sont associées une ou plusieurs étiquettes correspondant au type d'adaptation de l'exercice.

Un certain nombre d'exercices ont une consigne multiple et sont annotés par plusieurs étiquettes. Dans un premier temps, nous ne traitons que les exercices à une seule étiquette, la classification multi-classes sera abordée dans une prochaine étude.

Le tableau 3.1 ci-dessous détaille la répartition des exercices extraits et annotés à une seule étiquette pour chaque type d'adaptation et dans chacun des manuels scolaires.

Label	**Fréquence**			
	Manuel CE1	**Manuel CE2 1**	**Manuel CE2 2**	**Total**
CM (ChoixMultiples)	106	123	148	377
RC (RemplirChamp)	75	119	138	332
CocheMot	28	76	98	202
EditPhrase	24	93	36	153
ExpressionEcrite	12	89	47	148
TransformePhrase	9	84	50	143
TransformeMot	21	79	27	127
Classe	36	44	37	117
Associe	33	38	36	107
CocheGroupeMots	50	19	35	104
RCImage	57	42	5	104
CochePhrase	24	54	23	101
Texte	16	12	71	99
RCCadre	9	30	25	64
CacheIntrus	18	29	5	52
RCDouble	2	17	23	42
ClasseCM	13	9	16	38
CocheIntrus	10	11	16	37
Echange	4	18	15	37
Phrases	13	6	15	34
EditTexte	6	0	26	32
Atypique	3	20	6	29
CocheMot*	4	0	14	18
VraiFaux	3	14	0	17
CliqueEcrire	6	0	10	16
GenreNombre	2	5	4	11
CocheLettre	3	4	3	10
Question	0	4	6	10
ClassePhrase	0	5	4	9
Trait	2	2	5	9
AssocieCoche	4	1	3	8
NonAdaptable	3	0	3	6
CMDouble	0	5	0	5
CochePonctuation	0	0	3	3
EchangeLettre	0	0	1	1

TABLE 3.1 – Fréquence des exercices extraits à 1 seule étiquette par manuel

Les 35 étiquettes, accompagnées d'un exemple d'exercice tiré d'un manuel scolaire et de son adaptation par l'association, sont listées dans le tableau annexe A.1.

Les figures 3.1 et 3.2 présentent ici un exemple d'exercice pour chacun des deux labels majoritaires : respectivement « CM » (ChoixMultiples) et « RC » (RemplirChamp).

Les deux exemples sont des exercices à trous où l'élève doit compléter un trou dans une phrase, mais l'adaptation à la dyspraxie ne sera pas la même. Dans le cas d'un exercice « CM » adapté, l'élève clique sur le trou, une nouvelle fenêtre s'ouvre dans laquelle l'élève clique sur un item parmi des choix proposés, ici « ou » ou « où ». Dans le cas d'un exercice « RC », l'élève clique sur le trou et tape sa réponse au clavier.

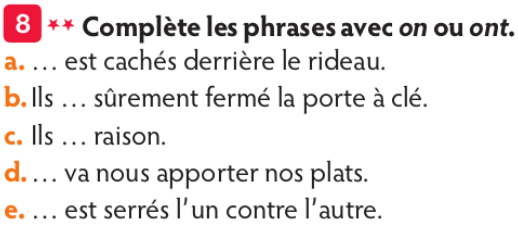

8 ** **Complète les phrases avec *on* ou *ont*.**
a. … est cachés derrière le rideau.
b. Ils … sûrement fermé la porte à clé.
c. Ils … raison.
d. … va nous apporter nos plats.
e. … est serrés l'un contre l'autre.

FIGURE 3.1 – Exemple d'exercice CM

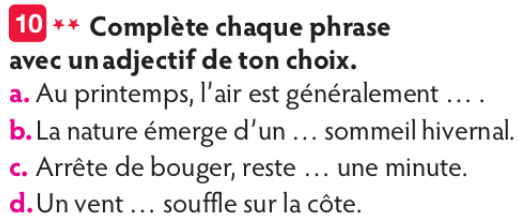

10 ** **Complète chaque phrase avec un adjectif de ton choix.**
a. Au printemps, l'air est généralement … .
b. La nature émerge d'un … sommeil hivernal.
c. Arrête de bouger, reste … une minute.
d. Un vent … souffle sur la côte.

FIGURE 3.2 – Exemple d'exercice RC

Parmi les 35 étiquettes proposées, 2 sont particulières : « NonAdaptable » et « Atypique ». Les membres de l'association considèrent que certains exercices ne doivent pas être adaptés, dans la mesure où ils ne sont pas réalisables ou ne présentent aucun intérêt pour un élève dyspraxique. D'autres exercices, qualifiés d'atypiques, sont des exercices uniques, très rares et/ou difficilement adaptables automatiquement. Un élève dyspraxique sera en mesure de les faire, ils seront alors adaptés manuellement. Dans le cadre de ce mémoire et suite à des premiers essais de classification non concluants, nous décidons d'omettre ces deux classes et de nous concentrer uniquement sur la classification des exercices adaptables automatiquement. Une classification précoce sur l'adaptabilité automatique des exercices fera l'objet de travaux futurs.

Enfin, certaines classes sont particulièrement sous-représentées. C'est principalement le cas de « EchangeLettre », pour laquelle nous ne disposons que d'un seul exemple d'exercice : elle sera exclue de la classification.

D'autre part, certaines classes sont très similaires. Par exemple, les exercices annotés « Classe » consistent à catégoriser des items dans des catégories spécifiées dans la consigne, en cliquant sur les items pour les faire apparaître en différentes couleurs. La tâche de « ClassePhrase » est identique, mais les items à catégoriser sont des phrases. Lors de l'adaptation n'est affichée qu'une phrase à la fois. L'objectif des exercices « ClasseCM » est le même, mais au-delà de 4 catégories, l'adaptation se fait comme un exercice à choix multiple : pour chaque item, l'élève clique sur un choix proposé. Un risque posé par cette classification est que le modèle ne parvienne pas à distinguer correctement ces classes entre elles. Une autre approche serait de regrouper les classes similaires en une seule et de les filtrer dans le processus d'adaptation, en fonction de la longueur des items ou du nombre de catégories.

Sur les 2602 exercices extraits annotés à une seule étiquette, 2566 constituent ainsi notre corpus. Le jeu d'étiquettes final compte 32 étiquettes, que nous pouvons répartir en 7 grandes catégories selon le type d'action à effectuer par l'élève (tableau 3.2).

Grande classe	Classes
Select	AssocieCoche Classe ClassePhrase CocheGroupeMots CocheLettre CocheMot CocheMot* CochePhrase CochePonctuation Trait
Choose	Associe ClasseCM CM CMDouble GenreNombre VraiFaux
Fill	CliqueEcrire ExpressionEcrite Phrases Question RC RCCadre RCDouble RCImage TransformeMot TransformePhrase
Edit	EditPhrase EditTexte
Swap	Echange
Show	Texte
Intrus	CacheIntrus CocheIntrus

TABLE 3.2 – Répartition des étiquettes en grandes catégories

3.3 Caractéristiques

Dans l'hypothèse où certaines classes présentent des caractéristiques spécifiques, nous proposons une analyse linguistique du corpus et une analyse statistique des informations de la mise en page. Nous sommes particulièrement attentifs à la présence de certaines classes grammaticales et entités nommées dans le texte, et à la mise en forme de l'exercice, par exemple l'utilisation d'images, de listes ou de plusieurs polices différentes.

3.3.1 Etiquetage morpho-syntaxique

Dans un premier temps, l'étiquetage morpho-syntaxique des documents permet le calcul des proportions de chaque classe grammaticale au sein des consignes et énoncés des différents types d'exercices.

Le graphique 3.3 montre la proportion de tokens étiquetés « PUNCT » dans un énoncé selon le type d'exercice. On relève significativement plus de signes de ponctuation dans les exercices annotés « RCCadre », « RC », « CliqueEcrire » et « CM », exercices dans lesquels l'élève est généralement amené à remplir un trou dans une phrase, en écrivant ou en cliquant sur un élément. Les signes de ponctuation représentent alors environ un quart de l'ensemble des tokens de l'énoncé.

De la même manière, le diagramme 3.4 montre, pour chaque classe, le pourcentage moyen de tokens étiquetés « NUM » dans un énoncé d'une part et dans une consigne d'autre part. Si la proportion reste faible par rapport aux classes grammaticales plus fréquentes, on constate que certaines classes d'exercices comptent relativement plus de tokens numériques que d'autres. Cela se confirme en observant de plus près les exercices. « Question » et « AssocieCoche » comptent plus de nombres dans l'énoncé, il s'agit d'exercices où les éléments de l'énoncé sont généra-

lement numérotés. On trouve davantage de nombres dans les consignes d'exercices de type « Phrases » , où l'on demande en effet à l'élève d'écrire un certain nombre de phrases.

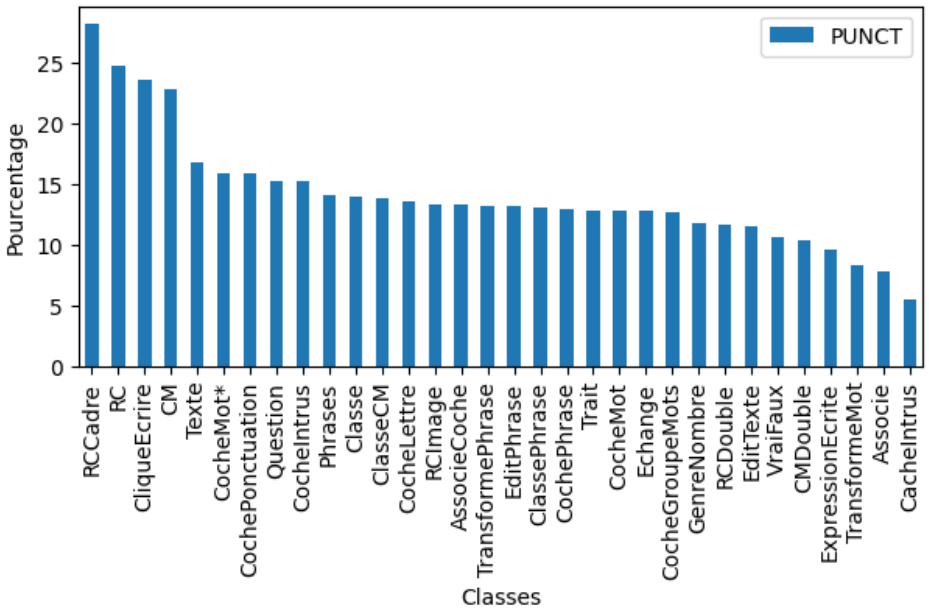


FIGURE 3.3 – Pourcentage de tokens PUNCT par énoncé par classe

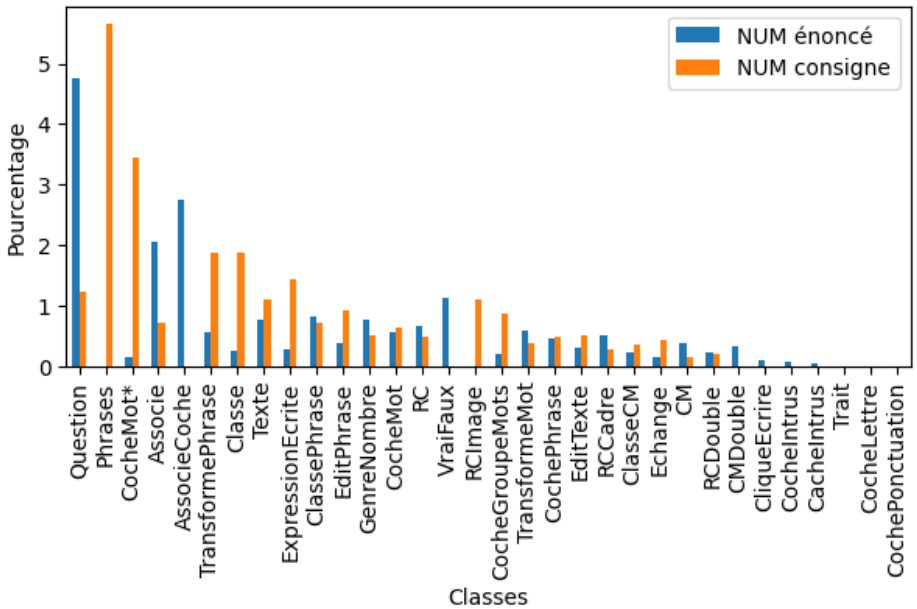


FIGURE 3.4 – Pourcentage de tokens NUM par énoncé et par consigne, par classe

Hormis les tokens numériques et les signes de ponctuation, l'analyse morpho-syntaxique ne révèle pas d'autres caractéristiques du corpus. Les proportions de tokens d'autres classes grammaticales se valent globalement dans chacune des classes et la variation des proportions observée d'un exercice à l'autre semble plutôt aléatoire.

3.3.2 Statistiques sur la mise en forme

Une autre analyse statistique que nous avons conduite concerne la mise en forme des exercices. En effet, grâce aux traitements d'extraction effectués sur les manuels, nous disposons d'informations sur les polices d'écriture, le nombre et la longueur des phrases, les coordonnées spatiales des mots sur la page, ainsi que la présence de listes ou d'images dans les exercices.

En adéquation avec nos données, la notion de phrase à été étendue. Généralement, la consigne d'un exercice est grammaticalement et sémantiquement correcte, commence par une majuscule et se termine par une ponctuation forte, comme une phrase dans sa définition linguistique la plus simple. Ce n'est pas toujours le cas dans les énoncés : on peut y trouver des mots seuls éparpillés, ou encore une suite de mots non grammaticale. Ainsi, ce que nous appelons *phrase* ici correspond à une entité de l'exercice. Le séquençage du texte en phrases appliqué lors de l'extraction du manuel tient compte des distances géométriques et d'indicateurs de début et de fin comme la ponctuation ou les puces de listes.

Le diagramme 3.5 rapporte, pour chaque classe, le nombre moyen de phrases dans un exercice. Nous constatons que certaines classes comme « RCImage », « Phrases », « ExpressionEcrite », « Question » et « Texte » comptent plus de phrases dans la consigne que dans l'énoncé. Des statistiques sur la longueur des phrases confirment également que la consigne de ces exercices est plus longue que leur énoncé. Les énoncés peuvent même être vides ou simplement contenir des images. Au contraire, certains exercices présentent un nombre de phrases dans l'énoncé bien plus élevé, allant jusqu'à 8 en moyenne pour les exercices « AssocieCoche ».

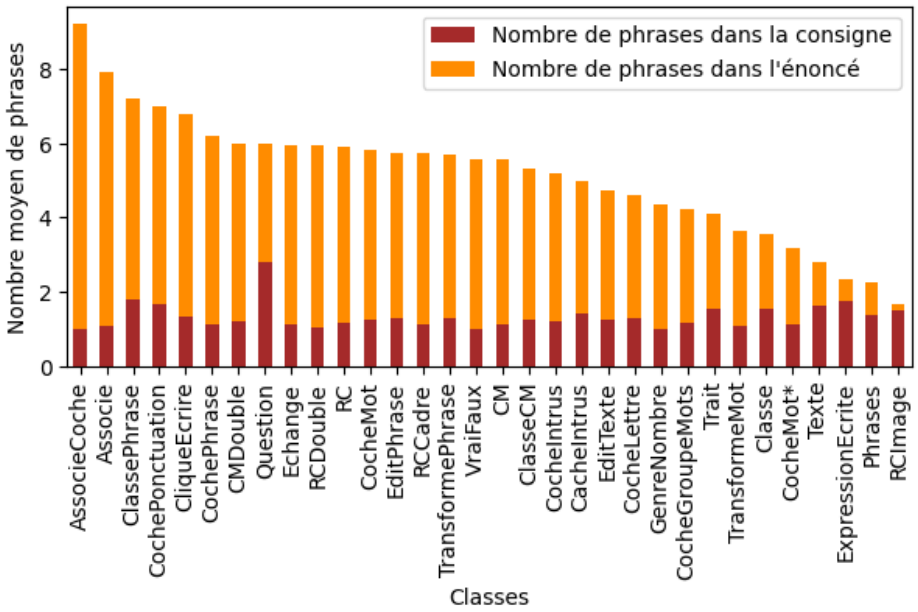


FIGURE 3.5 – Nombre moyen de phrases par exercice par classe

Le graphe 3.6 présente, pour chaque classe, le pourcentage d'exercices contenant une liste. La plupart des classes comptent moins de 10% d'exercices avec une liste. Deux classes se distinguent, pour lesquelles plus d'un exercice sur deux contient une liste. Des calculs statistiques sur les trois manuels séparément ont été réalisés pour

s'assurer que l'emploi de listes ne soit pas spécifique à un éditeur particulier : la proportion calculée dans chacun des manuels est plus ou moins similaire à celle calculée sur l'ensemble des exercices. En revanche, après observation du corpus, nous constatons que les items listés dans la consigne ou l'énoncé ne sont pas systématiquement récupérés lors de l'extraction. Nous pensons qu'une amélioration de l'étape d'extraction pour récupérer les types de listes et les items permettrait de détecter de nouvelles caractéristiques plus précises, qui pourraient ensuite être utilisées pour la classification en types d'exercices.

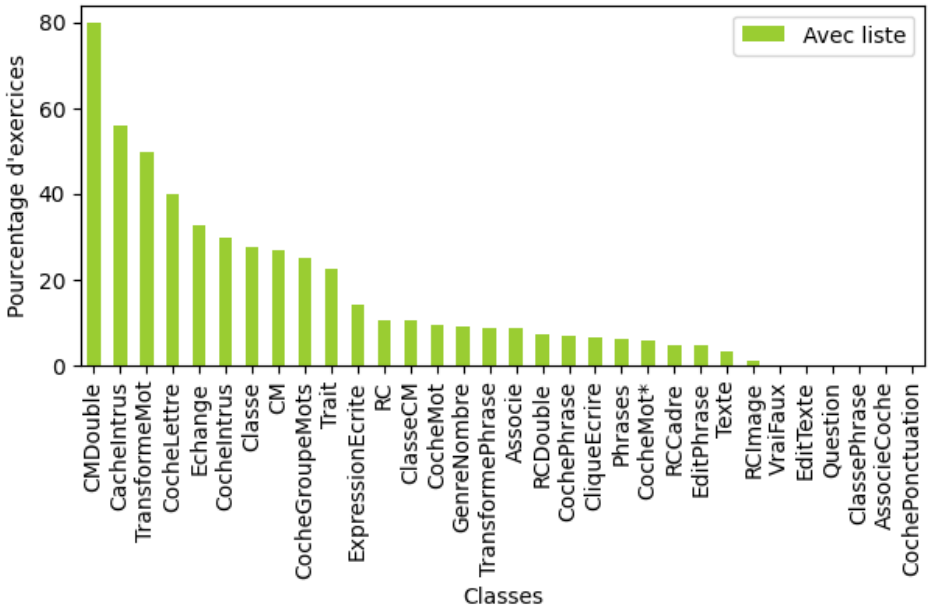


FIGURE 3.6 – Pourcentage d'exercices contenant une liste par classe

Par ailleurs, plusieurs polices d'écriture sont utilisées au sein d'un même exercice, voire au sein d'une consigne ou d'un énoncé. Une étude statistique effectuée sur les exercices des trois manuels extraits révèle que le nombre de polices différentes par exercice varie d'un éditeur à l'autre. L'éditeur du manuel 2 de niveau CE2 a tendance a employer plus de polices que l'éditeur des manuels CE1 et CE2 1 (figure 3.7).

Pour la tâche de classification, nous nous demandons si l'emploi d'un nombre de polices plus faible ou plus élevé est corrélé à certains types d'adaptation des exercices. Nous constatons d'abord que les exercices avec une consigne et un énoncé plus courts comptent, logiquement, moins de polices différentes. Au sein des consignes, les exercices de type choix multiple (par exemple « CM », « CMDouble » et « VraiFaux ») comptent plus d'1,5 polices en moyenne, contre 1 à 1,25 polices pour les autres classes. Dans l'énoncé, le nombre de polices utilisées semble plus variable, bien que trois groupes de classes se distinguent : la majorité contient environ 2 polices par énoncé, certaines classes en contiennent plus, en moyenne 2,5 à 3,5 polices différentes, et le reste ne compte qu'1 police, voire moins dans le cas d'énoncés vides de texte.

En plus de la tâche d'extraction des exercices réalisée en amont de la classification, un traitement des images a lieu afin de déterminer à quel exercice elles font référence et une classification détermine leur importance. Une image dans un exercice peut être indispensable à la réalisation de l'exercice, informative ou inutile. Par

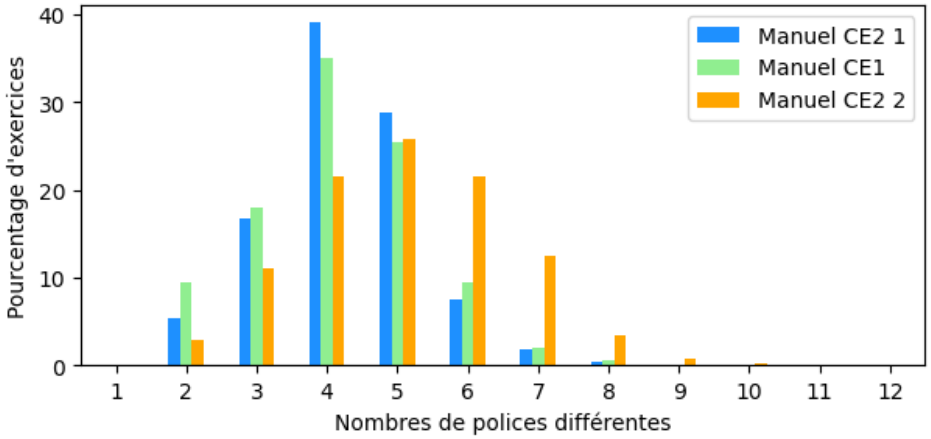


FIGURE 3.7 – Pourcentage d'exercices contenant une liste par classe

exemple dans le cas d'une adaptation à la dyspraxie, les images dites inutiles ne seront pas affichées dans le manuel adapté.

Le graphe 3.8 présente pour chaque classe le pourcentage d'exercices contenant ou non une image, et si l'éventuelle image est indispensable, informative ou inutile. Comme l'indique le nom de la classe, « RCImage » contient quasi-systématiquement une image indispensable à la réalisation de l'exercice. D'autres classes contiennent fréquemment des images, comme « ExpressionEcrite », « Question », « Phrases » et « Texte », où l'énoncé est souvent court et réfère à un thème particulier. L'image sert alors à illustrer le texte affiché à l'élève. C'est aussi le cas des images inutiles et informatives, qu'on retrouve également dans les exercices de type « EditTexte » et « CliqueEcrire ».

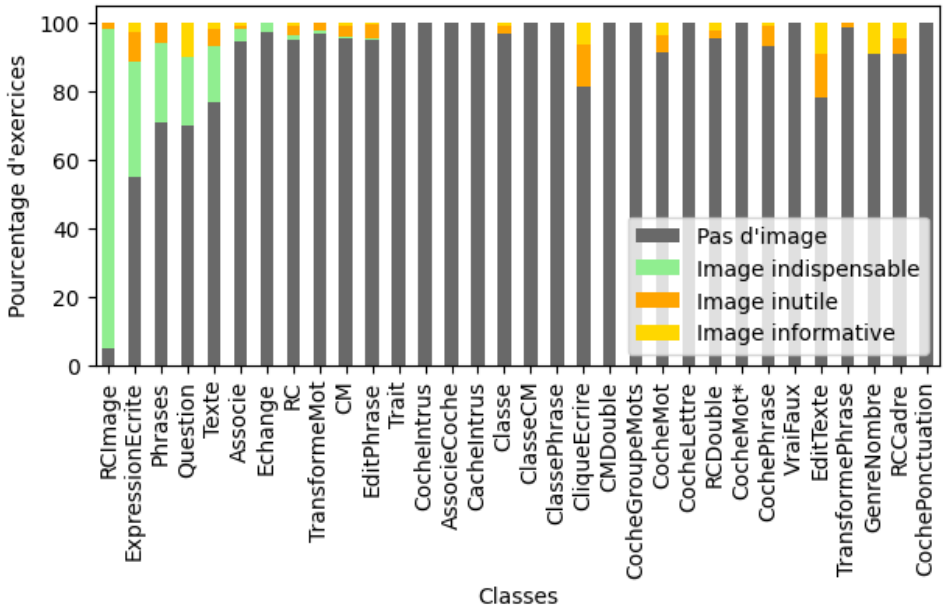


FIGURE 3.8 – Pourcentage d'exercices contenant une image par classe

Ces statistiques confirment l'importance de la mise en page des documents. Les

informations sur les phrases, les listes, les polices, et les images récupérées lors de l'extraction pourront être reprises lors des expérimentations. Tandis que les signes de ponctuation sont souvent omis dans les différentes tâches de traitement automatique de textes, ils constituent des éléments importants de notre corpus et peuvent permettre la discrimination des différentes classes. D'autres opérations ont été menées sur le corpus, comme la reconnaissance d'entités nommées, ce qui n'a pas été concluant.

3.4 Partitionnement des données

Les données sont stockées sous forme tabulaire dans des fichiers TSV, sous forme arborescente dans des fichiers JSON et sous forme d'images au format PNG. L'ensemble du corpus est divisé en trois jeux de données selon la répartition détaillée dans le tableau 3.3. Le tableau annexe A.2 détaille le nombre de données par classe et par grande classe pour chacun des jeux de données. La proportion des classes d'exercices par manuel et le ratio entre les classes sont conservés.

Jeu de données	**Apprentissage**	**Validation**	**Test**	**Total**
Pourcentage	70	10	20	100
Nombre d'exercices	1799	257	510	2566

TABLE 3.3 – Répartition des exercices dans les jeux de données

3.5 Limites

Si la tâche de classification est un objet d'étude courant en traitement automatique des langues depuis plusieurs années, notre corpus constitue un défi de part sa multimodalité mais surtout le caractère inédit de la structure des manuels scolaires. Bien que nous ayons mis en place une chaîne de traitement de qualité nous permettant d'obtenir un jeu de données de 2566 exercices, cela reste peu compte tenu de notre objectif de classification en 32 classes. En effet, dans les jeux de données de validation et de test, certaines classes comptent moins d'une dizaine d'exercices. Les exercices des classes les plus fournies auront plus de poids lors de l'apprentissage, ce qui entraînera une répercussion sur le modèle à chaque étape de validation. Les biais introduits peuvent être un sur-apprentissage ou une fluctuation des métriques d'évaluation.

Les versions numériques de manuels scolaires étant soumises à des droits d'auteurs, nos données d'entraînement reposent sur le bon vouloir des maisons d'édition. Jusqu'à présent, ce sont les éditeurs qui fournissent leurs manuels à l'association *Le Cartable Fantastique*, sous réserve d'obtenir à terme une adaptation.

Parmi les manuels obtenus, trois ont été extraits et annotés. L'annotation, si elle est plus rapide qu'une adaptation manuelle, requiert aussi du temps et surtout l'expertise d'enseignants et chercheurs spécialistes des troubles et des apprentissages.

D'autre part, la liste des sous-types d'adaptation n'est pas figée, et l'annotation peut varier : un même exercice peut être adapté de différentes manières. Compte tenu de la diversité des collections et le renouvellement des manuels, il existe une multitude d'exercices différents, dont certains sont propres à une maison d'édition. Les

exercices évoluent avec le temps, le programme pédagogique et les avancées pédagogiques. A chaque nouveau manuel annoté, les membres de l'association proposent généralement de nouvelles adaptations et ainsi un nouveau type d'exercice. De ce fait, certains types sont peu fréquents par rapport à d'autres, ce qui entraîne un important déséquilibre des classes de notre jeu de données. De plus, le nombre de classes ne cesse d'augmenter, ce qui complique inévitablement les tâches de classification et d'adaptation.

En outre, les classes d'exercices « NonAdaptable » et « Atypique » posent problème pour la tâche de classification, à la fois de par leur diversité et leur ressemblance possible avec n'importe quelle autre classe. Si elles sont exclues dans un premier temps dans le cadre ce mémoire, il est prévu d'ajouter à la chaîne de traitement une étape de classification précoce pour détecter ces exercices à adapter manuellement et ainsi alléger au maximum le travail des membres de l'association.

Enfin, les données que nous utilisons résultent d'une première étape d'extraction qui peut être source d'erreurs. Quelques exercices ne sont pas extraits, c'est par exemple le cas de certains exercices n'ayant pas de numéro, mais un nom (exemples : « Défi langue », « Autodictée »). D'autres erreurs sont observées au sein des exercices extraits : parfois, des éléments de l'énoncé se retrouvent dans la consigne, le texte n'est pas récupéré dans le bon ordre, et les exercices présentés sous forme de tableau posent particulièrement problème. D'autre part, à ce stade de l'extraction, les images des manuels ne sont pas associées à leurs exercices respectifs. Il se peut donc que nous manquions d'informations textuelles ou visuelles. Il sera indispensable de compléter l'étape d'extraction pour améliorer nos données.

Conclusion

Le corpus final est constitué de 2566 exercices de français issus d'un manuel scolaire de niveau CE1 et de deux manuels de niveau CE2, et annotés à partir de 32 types d'adaptation. Les exercices sont stockés sous différentes formes (table, arborescence et image) et répartis en jeux de données d'apprentissage, de validation et de test, afin de pouvoir entraîner et comparer divers modèles dans la phase d'expérimentation.

Les étiquettes non traitées et les classes sous-représentées feront l'objet de traitements supplémentaires en amont de la classification. La chaîne de traitement d'extraction et le jeu d'étiquettes pourront également être revus de sorte à optimiser le processus d'adaptation.

Par ailleurs, des premières analyses statistiques sur le corpus ont permis de mettre en lumière quelques caractéristiques morpho-syntaxiques et stylistiques, potentiellement discriminantes pour la classification.

CHAPITRE 4

PROTOCOLE EXPÉRIMENTAL

Sommaire

Introduction

Dans ce chapitre, nous présentons le protocole expérimental mis en place pour la tâche de classification des exercices.

La figure 4.1 rappelle la chaîne de traitement de l'exercice extrait à l'exercice adapté. Notre travail porte sur la deuxième classification, dont l'objectif est de prédire le type d'adaptation de l'exercice donné en entrée. La première classification et l'automatisation de l'adaptation seront traitées dans des prochains travaux.

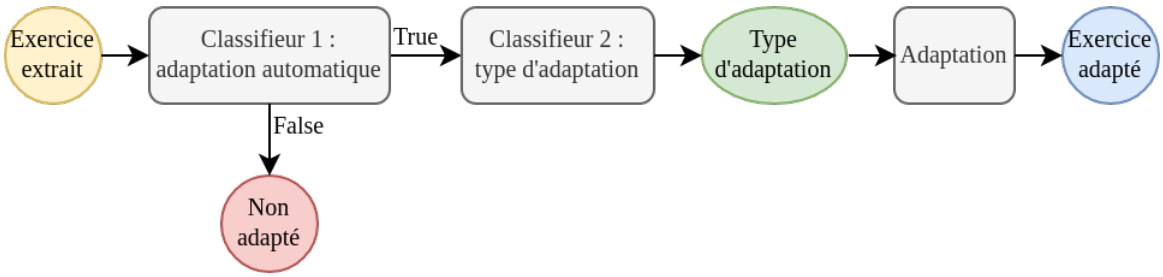


FIGURE 4.1 – Chaîne de traitement : classification et adaptation d'un exercice

D'abord, nous détaillons dans la section 4.1 la chaîne de traitement de la tâche de classification et les différentes techniques appliquées à notre corpus d'exercices. Nous mentionnons les classifieurs sélectionnés, la préparation des données d'entrée, la fusion de scores, l'enchaînement de plusieurs classifieurs en cascade, ainsi que le paramétrage des modèles pour la phase d'apprentissage.

Dans la section 4.2, nous présentons les mesures d'évaluation utilisées dans nos expérimentations.

4.1 Chaîne de traitement

Dans cet objectif de classification des exercices, nous proposons une comparaison de différentes méthodes de classification. Les classifieurs expérimentés peuvent se regrouper en trois familles :

— les classifieurs statistiques ;
— BERT ;
— LayoutLM ;

auxquelles s'ajoute une dernière méthode hybride fondée sur la fusion tardive.

Les entrées de tous les classifieurs sont les exercices. Selon le type de classifieur, différents pré-traitements sont appliqués aux données d'entrée.

4.1.1 Classifieurs

Classifieurs statistiques

Notre choix s'est porté sur six méthodes statistiques traditionnelles :

— Multinomial Naïve Bayes (MNB)
— Complement Naïve Bayes (CNB) ;
— K plus proches voisins (KNN) ;
— Arbres de décision (DT) ;
— Forêts décisionnelles (RF) ;
— Machine à vecteurs de support (SVM).

Nous utilisons les algorithmes pour la classification multi-classes, ainsi que la stratégie one-vs-rest qui consiste à résoudre un problème de classification de n classes en n classifications binaires. Notons que l'algorithme SVM, conçu pour une classification binaire, ne fonctionne qu'avec une stratégie one-vs-rest.

BERT

Le modèle CamemBERT [Martin et al., 2020b] est basé sur l'évolution de BERT RoBERTa. Nous utilisons l'architecture de base de CamemBERT : $\text{CamemBERT}_{\text{BASE}}$. Le modèle est pré-entraîné sur le corpus OSCAR [1] (Open Super-large Crawled Aggregated coRpus), avec 12 couches, 768 dimensions cachées et 12 têtes d'attention, soit 110 000 paramètres.

Dans le but d'améliorer le modèle de classification, nous expérimentons également l'intégration de caractéristiques visuelles, la variation de l'architecture du modèle et le fine-tuning du modèle de langue sur les manuels scolaires.

LayoutLM

LayoutLMv2 [Xu et al., 2021] est la deuxième version de LayoutLM, un modèle pré-entraîné sur le texte et la structure pour la compréhension d'images de documents. Nous utilisons son architecture de base : $\text{LayoutLMv2}_{\text{BASE}}$. Le modèle est pré-entraîné sur le corpus IIT-CDIP (Illinois Institute of Technology Complex Document Information Processing Test Collection) [Lewis et al., 2006], constitué de documents scannés d'archives de poursuites judiciaires contre des fabricants de tabac. L'encodeur du transformer est paramétré avec 12 couches et 12 têtes d'attention, et le modèle compte 768 dimensions cachées.

1. https://oscar-corpus.com/

LayoutLMv2 est affiné autour de tâches spécifiques. Nous utilisons le modèle LayoutLMv2ForSequenceClassification, qui comprend une tête de classification sur le token CLS de manière à prédire la classe du document.

4.1.2 Données d'entrée

Les exercices donnés en entrée des modèles sont présentés sous différentes formes en fonction du classifieur.

Classifieurs statistiques

Pour les modèles statistiques, nous utilisons uniquement le texte, pré-traité puis vectorisé sur la mesure TF-IDF.

Afin d'obtenir la suite de pré-traitements la plus adéquate à notre corpus, nous testons plusieurs configurations de pré-traitements.

Pour chaque jeu de données sont appliqués les pré-traitements de base suivants :

— normalisation des espaces et des majuscules en minuscules ;
— tokenisation ;
— suppression des stopwords.

Selon l'expérience, les pré-traitements suivants sont également appliqués :

— recherche de n-grams ;
— étiquetage morpho-syntaxique ;
— lemmatisation ;
— racinisation.

Suite à l'observation de nos données, nous savons que les signes de ponctuation et les caractères spéciaux peuvent être des caractéristiques discriminantes. Par exemple, on trouvera davantage de points de suspension dans les énoncés d'exercices à trous. Ainsi, nous ne procédons pas à la suppression des signes de ponctuation et des caractères spéciaux, bien qu'elle soit souvent appliquée dans les systèmes de traitement automatique des langues.

L'extraction de n-grams permet de relever des patterns de tokens spécifiques à certaines classes.

L'étiquetage morpho-syntaxique permet de filtrer davantage les données en cas d'homographes et d'ambiguïté syntaxique.

La lemmatisation et la racinisation semblent importantes dans la mesure où tous les manuels n'emploient pas toujours le même mode de conjugaison. Par exemple, une consigne peut être à l'infinitif, ou conjuguée à l'impératif présent à la deuxième personne du singulier ou à l'impératif présent à la deuxième personne du pluriel.

BERT et LayoutLM

Pour les méthodes neuronales, les données sont converties en vecteurs directement par le modèle. Pour BERT, ce sont les plongements de mots de BERT. Pour LayoutLM, les plongements de mots du modèle sont accompagnés des plongements d'images et des plongements de position.

Un exercice étant constitué de plusieurs parties distinctes, il semble essentiel de pouvoir séparer l'entrée en plusieurs textes distincts. Pour un transformer classique, il est tout à fait possible de fournir en une unique entrée plusieurs séquences distinctes grâce aux identifiants de type d'entrée. Dans notre cas, il s'agit d'un vecteur

de 0 et de 1. 0 correspond aux tokens de la consigne et 1 aux tokens de l'énoncé. Pour les autres architectures explorées, deux entrées distinctes passent dans le modèle.

4.1.3 Fusion des modalités

Afin de tirer profit de nos différentes sources de données, nous proposons enfin des approches multimodales basées sur la fusion tardive des modèles neuronaux présentés jusqu'alors.

Si LayoutLMv2 repose déjà sur une fusion précoce des modalités textuelles et visuelles, son modèle de langue est entraîné sur des documents en langue anglaise. Nous suggérons d'utiliser pour des travaux futurs un modèle étendu préentraîné sur un corpus multilingue, ou le fine-tuning du modèle de langue de LayoutLMv2ForSequenceClassification, difficilement réalisable à notre niveau. Dans un premier temps, nous proposons une fusion tardive des meilleurs modèles de CamemBERT et LayoutLMv2ForSequenceClassification, qui combine les sorties des deux classifieurs.

Les prédictions des deux classifieurs sont normalisées, puis fusionnées. Différentes méthodes de fusion sont expérimentées.

4.1.4 Classification directe ou en cascade

Les 32 étiquettes pouvant être regroupées en 7 grandes familles de classes, nous imaginons deux processus de classification, schématisés respectivement sur les figures 4.2 et 4.3. Le premier consiste à apprendre la classification des exercices directement sur les 32 classes retenues. Le deuxième est une classification en cascade : d'abord le grand type, puis le sous-type d'adaptation final en fonction du grand type prédit. Après élaboration d'une baseline solide avec les modèles traditionnels, chaque méthode neuronale de classification sera alors appliquée sur la classification directe et sur la classification en cascade.

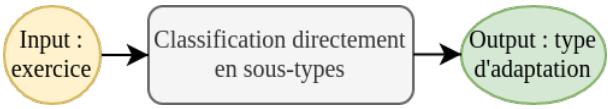


FIGURE 4.2 – Classification directe

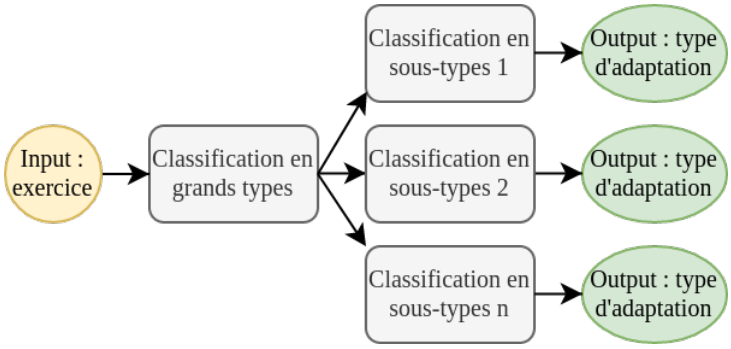


FIGURE 4.3 – Classification en cascade

4.1.5 Hyperparamètres

Afin de pouvoir comparer les différents modèles, les mêmes hyperparamètres sont appliqués à tous les classifieurs d'un même type. Ceux-ci ont été définis suite à quelques essais d'apprentissage avec diverses valeurs, par comparaison des prédictions et des courbes de précision et de perte.

Pour une classification directe, la taille du batch est fixée à 16, le taux d'apprentissage à 0.0001, et le nombre d'époques à 40.

Pour une classification en cascade, 6 classifieurs sont entraînés : 1 pour la classification en grands types, et 5 pour la classification en sous-types. Compte tenu de l'irrégularité du nombre d'étiquettes et du volume de données d'entraînement par classification, la taille du batch varie entre 8 et 16 et le nombre d'époques entre 20 et 30.

D'autre part, nous entraînons les modèles d'apprentissage profond avec un processus de rétro-propagation du gradient utilisant la fonction de perte d'entropie croisée et l'algorithme d'optimisation Adam.

4.2 Mesures d'évaluation

Une prédiction en sortie d'un classifieur est un vecteur de dimension n correspondant aux n classes à prédire. La classe finale prédite pour la classification uni-label est celle pour laquelle le score est le plus élevé.

Pour tous les modèles expérimentés, la qualité de la classification est principalement mesurée avec la métrique d'**exactitude**. Fréquemment utilisée pour la tâche de classification, l'exactitude exprime le nombre de prédictions correctes (vrais positifs + vrais négatifs) rapporté au nombre total d'éléments dans le corpus.

$$Exactitude = \frac{TP + TN}{TP + TN + FP + FN}$$

La précision, le rappel et la f-mesure sont trois autres métriques permettant de nuancer les résultats. Elles se calculent indépendamment sur chacune des classes, et sur l'ensemble des données.

La **précision** mesure le taux de prédictions exactes par rapport aux prédictions calculées. Elle se calcule par : le nombre d'éléments correctement étiquetés (vrais positifs) rapporté au nombre d'éléments prédits (vrais positifs + faux négatifs).

$$Précision = \frac{TP}{TP + FP}$$

Le **rappel** mesure le taux de prédictions correctes par rapport à l'ensemble attendu. Il se calcule par : le nombre d'éléments correctement étiquetés (vrais positifs) rapporté aux nombres d'éléments effectivement positifs (vrais positifs + faux négatifs).

$$Rappel = \frac{TP}{TP + FN}$$

La **f-mesure** est la moyenne harmonique pondérée de la précision et du rappel. Les poids de la précision et du rappel sont équilibrés.

$$F1 = \frac{2 * Précision * Rappel}{Précision + Rappel} = \frac{2 * TP}{2 * TP + FP + FN}$$

Ces métriques peuvent être calculées en micro, c'est-à-dire que le même poids est attribué à chacun des éléments mesurés, indépendamment de leur classe.

Les macro-moyennes sont calculées à partir des métriques indépendantes calculées pour chaque classe. comme la moyenne des mesure calculées de manière indépendante pour chaque classe, puis la moyenne est calculée Au contraire, les macro-mesures accordent autant d'importance à chaque classe, quel que soit son effectif. Elles permettent d'évaluer le modèle en tenant compte des performances par classe mais en restant robuste au déséquilibre du jeu de données.

Ainsi, la **macro f-mesure** est la moyenne des f-mesures de chaque classe. Soit n le nombre de classes,

$$MacroF1 = \frac{\sum_{i=1}^{n}(F1_i)}{n}$$

Les variantes pondérées des macro-moyennes permettent de tenir compte des performances par classe tout en préservant les proportions de chaque classe.

Ainsi, la **moyenne pondérée de la f-mesure** (weighted average f1) se calcule à partir des f-mesures de chaque classe, pondérées par l'effectif de la classe. Soient n le nombre de classes, et w_i le coefficient de la classe i,

$$WAvgF1 = \frac{\sum_{i=1}^{n}(w_i * F1_i)}{\sum_{i=1}^{n}(w_i)}$$

De la même manière, les **moyennes pondérées de la précision et du rappel** s'obtiennent de la façon suivante :

$$WAvgPrécision = \frac{\sum_{i=1}^{n}(w_i * Précision_i)}{\sum_{i=1}^{n}(w_i)}$$

$$WAvgRappel = \frac{\sum_{i=1}^{n}(w_i * Rappel_i)}{\sum_{i=1}^{n}(w_i)}$$

Bien que la f-mesure soit une moyenne de la précision et du rappel, le calcul des moyennes pondérées des différentes métriques peut impliquer un score de f-mesure qui n'est pas situé entre les valeurs de la précision et du rappel.

Les mesures d'évaluation présentées permettent d'évaluer, sur le corpus de test, les différents classifieurs entraînés.

Pour chacune des mesures, plus le score se rapproche de 1, plus le classifieur est performant. En l'occurrence, la f-mesure traduisant l'équilibre entre la précision et le rappel, un bon classifieur doit présenter une f-mesure élevée. Si toutes les mesures sont importantes, nous favorisons toutefois l'exactitude puis la précision par rapport au rappel, de sorte à obtenir le moins d'erreurs possible. Dans un contexte d'adaptation, il s'agit en effet de minimiser le travail manuel engendré par les erreurs.

Dans le chapitre suivant et pour chaque expérience réalisée, les tableaux de présentation des scores contiennent :

— l'exactitude, qui correspond aussi à la micro-précision, au micro-rappel et à la micro-f-mesure calculés sur l'ensemble des données de test,
— les moyennes pondérées du rappel, de la précision et de la f-mesure, pour tenir compte de l'ensemble des classes proportionnellement à leur effectif,
— puis la macro-moyenne de la f-mesure, qui accorde autant d'importance à chacune des classes.

CHAPITRE 5

EXPÉRIENCES

Sommaire

Introduction

Diverses expériences sont menées en suivant le protocole expérimental. Au sein de ce chapitre, nous reprenons en détail les approches mises en œuvre et entraînons différents modèles d'apprentissage automatique sur notre corpus. Les résultats et les critiques seront exposés et discutés.

5.1 Classification

5.1.1 Algorithmes traditionnels

Les premières expériences portent sur les algorithmes de classification statistiques traditionnels :

— Multinomial Naïve Bayes (MNB)
— Complement Naïve Bayes (CNB) ;
— K plus proches voisins (KNN) ;
— Arbres de décision (DT) ;
— Forêts décisionnelles (RF) ;
— Machine à vecteurs de support (SVM).

Entraînement et évaluation

Chacun des algorithmes est appliqué deux fois, sur le contenu textuel de tout l'exercice et sur la consigne uniquement, de sorte à observer l'impact du contenu des données en entrée. Nous pensons que la consigne porte plus d'information sur le type d'exercice, alors que le contenu de l'énoncé est plus aléatoire et peut parfois être source de bruit.

D'autre part, nous expérimentons plusieurs combinaisons de pré-traitements.

Les SVM, de par leur nature, ainsi que les algorithmes de type Naïve Bayes performent mieux avec la stratégie one-vs-rest. Les autres algorithmes présentent de meilleurs scores directement sur la classification multi-classes, ceci pourrait s'expliquer par leur nature arborescente ou spatiale et par le grand nombre de classes. En ce qui concerne l'algorithme des k plus proches voisins, suite à plusieurs essais, nous appliquons la classification multi-classes et le nombre de voisins est paramétré à 5.

Selon l'algorithme utilisé, les meilleurs résultats sont obtenus avec la configuration de base : normalisation, tokenisation, suppression des stopwords ; ou avec la suite de pré-traitements : normalisation, tokenisation, étiquetage morpho-syntaxique, suppression des stopwords, recherche de bigrammes. Les résultats des classifieurs pour ces deux combinaisons sont reportés dans les tableaux 5.1 et 5.2.

Algorithme	Mesures				
	Exactitude	W Avg Rappel	W Avg Précision	W Avg F1	Macro F1
1-vs-rest MNB	0.52	0.54	0.52	0.48	0.31
1-vs-rest CNB	0,63	0.61	0.63	0.60	0,45
KNN (k=5)	0.57	0.56	0.57	0.55	0.46
DT	0,56	0.45	0.56	0.55	0.45
RF	0,65	0,64	0.65	0,64	0,53
1-vs-rest SVM	**0,68**	**0.66**	**0.68**	**0.66**	**0,54**

5.1.a Classification sur le contenu textuel de la consigne

Algorithme	Mesures				
	Exactitude	W Avg Rappel	W Avg Précision	W Avg F1	Macro F1
1-vs-rest MNB	0.29	0.42	0.29	0.21	0.09
1-vs-rest CNB	0.53	0.52	0.53	0.50	0.35
KNN (k=5)	0.38	0.41	0.38	0.37	0.28
DT	0.50	0.51	0.50	0.49	0.37
RF	0.62	0.59	0.62	0.59	0.43
1-vs-rest SVM	0.60	0.59	0.60	0.58	0.42

5.1.b Classification sur le contenu textuel de l'exercice entier

TABLE 5.1 – Résultats de la classification avec les pré-traitements : normalisation, tokenisation, suppression des stopwords

Algorithme	Mesures				
	Exactitude	W Avg Rappel	W Avg Précision	W Avg F1	Macro F1
1-vs-rest MNB	0.55	0.62	0.55	0.52	0.36
1-vs-rest CNB	0,66	0,64	0.66	0.64	0,51
KNN (k=5)	0.56	0.54	0.56	0.53	0.48
DT	0,55	0.54	0.55	0,54	0,44
RF	0,65	0.63	0.65	0,64	0,5
1-vs-rest SVM	**0,68**	**0.66**	**0.68**	**0.67**	**0,55**

5.2.a Classification sur le contenu textuel de la consigne

Algorithme	Mesures				
	Exactitude	W Avg Rappel	W Avg Précision	W Avg F1	Macro F1
1-vs-rest MNB	0.25	0.44	0.25	0.17	0.07
1-vs-rest CNB	0.42	0.57	0.42	0.39	0.23
KNN (k=5)	0.40	0.49	0.40	0.38	0.36
DT	0.49	0.48	0.49	0.48	0.37
RF	0.60	0.61	0.60	0.58	0.41
1-vs-rest SVM	0.64	0.65	0.64	0.63	0.50

5.2.b Classification sur le contenu textuel de l'exercice entier

TABLE 5.2 – Résultats de la classification avec les pré-traitements : normalisation, tokenisation, étiquetage morpho-syntaxique, suppression des stopwords, recherche de bigrammes

Dans nos expérimentations, la lemmatisation n'a pas permis l'amélioration des résultats. Cela peut s'expliquer par le fait que les trois manuels de notre corpus partagent la même modalité : les consignes sont conjuguées à l'impératif présent. Toutefois, pour une généralisation sur d'autres manuels, la lemmatisation des données d'entraînement et d'évaluation sera nécessaire. De plus, les résultats obtenus avec une étape supplémentaire de racinisation sont similaires à ceux obtenus avec uniquement lemmatisation.

Notre hypothèse se confirme : quel que soit l'algorithme, les résultats sont nettement supérieurs avec comme entrée uniquement le texte de la consigne, plutôt que l'ensemble de l'exercice. Cet écart de score semble diminuer avec la complexité de l'algorithme : il est particulièrement élevé avec les algorithmes de type Naïve Bayes, moins avec les SVM.

La macro f-mesure confirme également l'introduction de biais causée par la diversité des énoncés d'exercices. Pour un modèle Multinomial Naïve Bayes, elle n'atteint pas 0.10. L'observation des prédictions montre que le modèle sur-apprend sur 10 classes parmi le total de 32 classes. Avec de meilleurs entrées et modèles, elle atteint au maximum 0.55, ce qui prouve la difficulté à traiter équitablement chacune des classes.

Les micro-mesures et les moyennes pondérées restent également basses pour une tâche de classification : sur le contenu textuel de la consigne, l'exactitude varie entre 0.52 et 0.68. Les moyennes pondérées de la précision et du rappel varient selon les algorithmes mais sont globalement proches, l'équilibre obtenu entre ces deux métriques est satisfaisant bien que nous recherchons un score de précision bien plus élevé. Ces expérimentations constituent toutefois une baseline solide et nous laissent une marge de progression pour la suite des expériences. L'amélioration constatée avec l'augmentation de la complexité des algorithmes nous laisse espérer une évolution avec des méthodes d'apprentissage neuronales.

Plongements et importance des caractéristiques

Les expériences présentées ci-dessus ont permis l'extraction des caractéristiques les plus importantes. En particulier, les arbres de décisions et forêts décisionnelles permettent de visualiser facilement l'importance d'une caractéristique dans la classification. Le graphique 5.1 présente les premières quarante caractéristiques les plus utilisées dans chaque arbre de la forêt décisionnelle, avec comme entrée le texte de la consigne pré-traité selon la configuration de base.

Pour les autres algorithmes, les coefficients d'importance des caractéristiques sont extraits. Plus la valeur du coefficient d'une caractéristique est élevée, plus il est probable que cette caractéristique soit propre à une classe et pèse dans la classification. Le graphique 5.2 montre, pour chaque classe, la caractéristique présentant le plus haut coefficient d'importance dans la classification avec l'algorithme SVM.

Dans les deux cas, les éléments discriminants qui reviennent principalement sont des verbes de consigne, comme « *complète* », « *recopie* » ou « *associe* », ou des noms comme « *intrus* », « *exemple* », « *ordre* » ou « *liste* ». Avec d'autres configurations de pré-traitements, les caractéristiques sont surtout des verbes, seuls ou dans des bigrammes accompagnés d'un objet ou d'un adverbe. Le vocabulaire des consignes est très présent, et certains verbes semblent être spécifiques à un type d'exercice particulier.

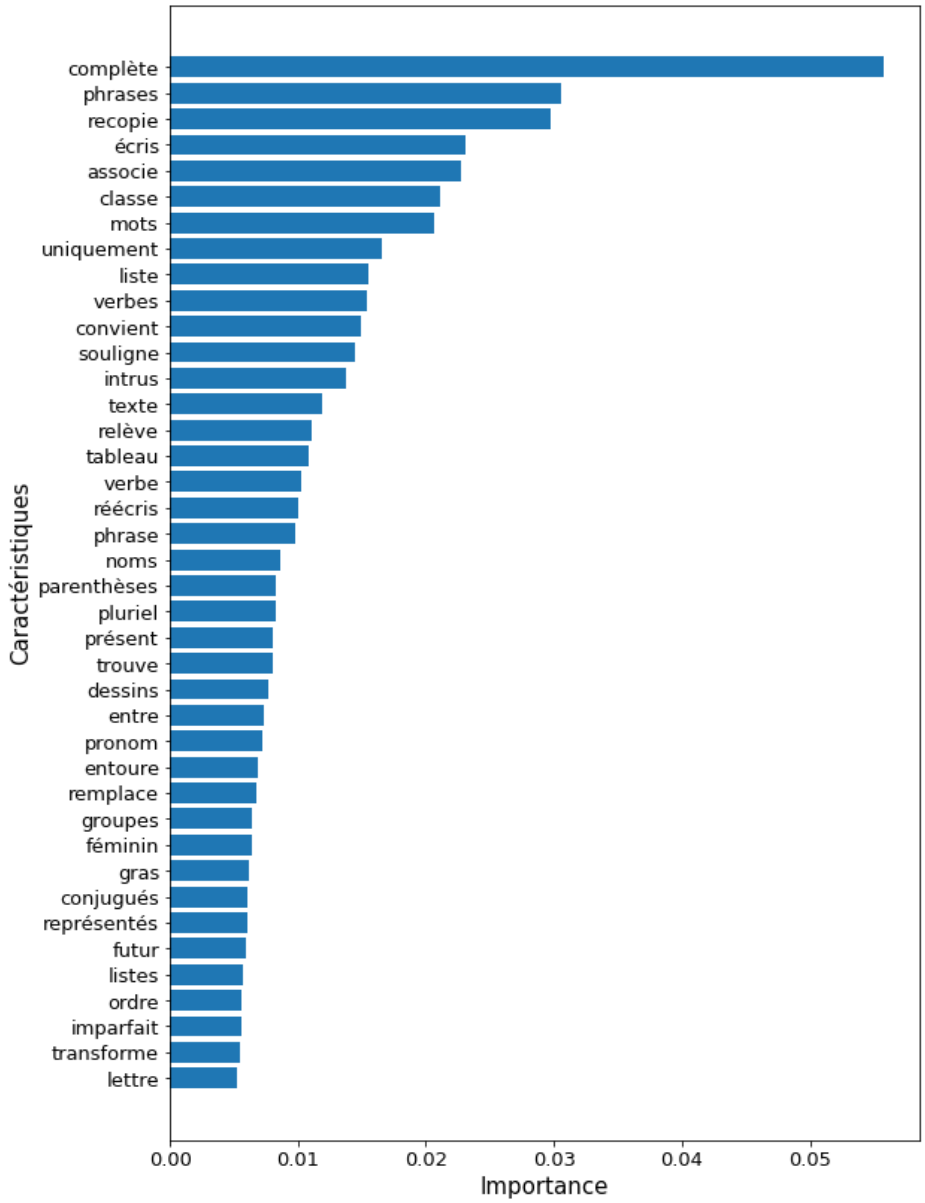


FIGURE 5.1 – Caractéristiques les plus importantes utilisées par l'algorithme des forêts décisionnelles

D'autre part, des techniques de réduction de la dimensionnalité pour la visualisation des données sont appliquées aux plongements de mots. L'objectif est de projeter les données sur un espace en deux dimensions, de sorte à grouper les éléments d'une même classe et éloigner les éléments dissemblables. Plusieurs techniques et paramètres sont expérimentés.

L'Analyse en Composantes Principales (ACP) est une méthode linéaire. Il s'agit d'identifier les directions, aussi appelées composantes principales, le long desquelles la variation des données est maximale.

La Décomposition en Valeurs Singulières (SVD) [Zhang et al., 2007] est une autre méthode linéaire de réduction dimensionnelle. Elle cherche également à identifier les composantes principales, en décomposant une matrice en un produit de trois matrices.

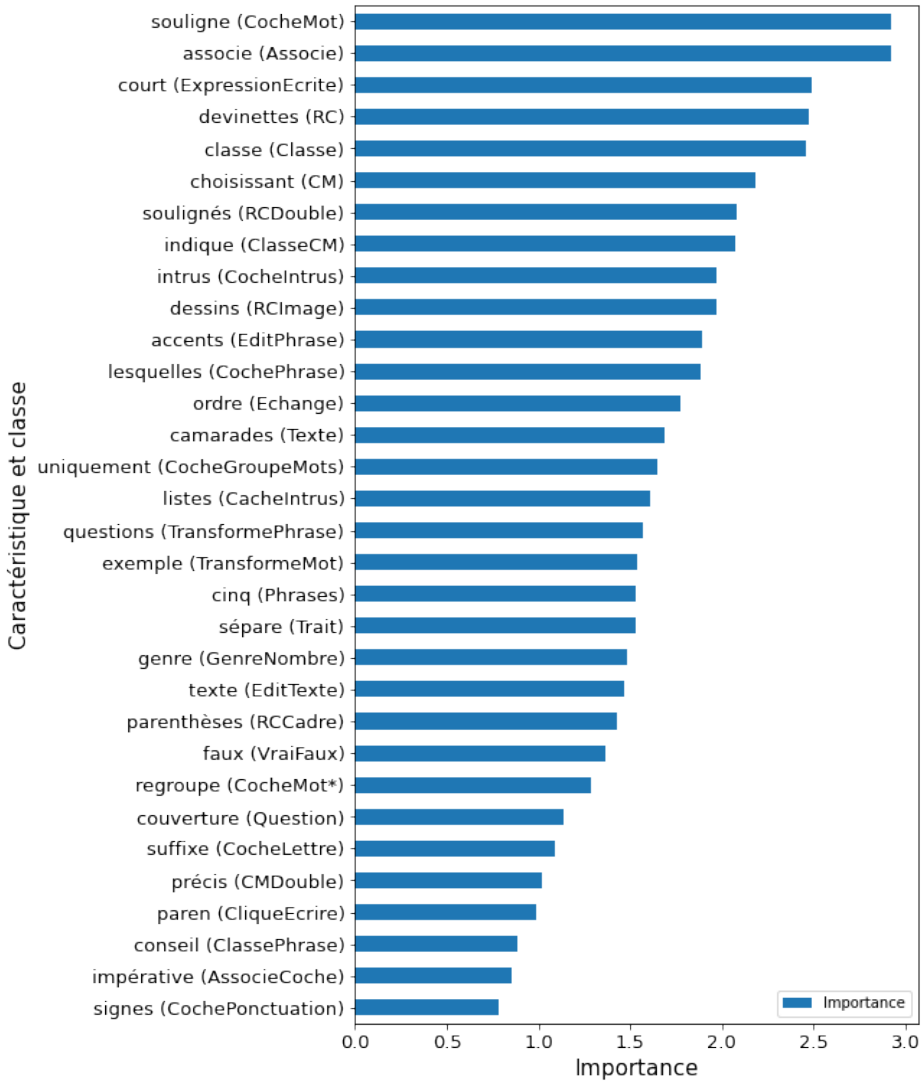


FIGURE 5.2 – Caractéristique la plus importante pour chaque classe utilisée par l'algorithme SVM

L'algorithme t-Distributed Stochastic Neighbor Embedding (t-SNE) [van der Maaten and Hinton, 2008] est une technique probabilistique non linéaire. Il crée une nouvelle représentation des données à partir des similitudes entre les éléments. La similarité de chaque paire repose sur des distributions de probabilité. La perplexité est un paramètre variable du t-SNE, plus elle est élevée, plus on considère les voisins éloignés.

Dans le cas de données à dimension très élevée, il peut aussi être pertinent de combiner plusieurs techniques de réduction dimensionnelle. Nous expérimentons en l'occurrence la succession des algorithmes SVD et t-SNE.

Aucune de ces techniques n'amène à une séparation nette des données. Le graphe 5.3 présente la projection des plongements de mots en deux dimensions avec la combinaison des techniques SVD et t-SNE avec une perplexité paramétrée à 100. Nous n'observons effectivement aucun regroupement significatif, mais un entremêlement de tous les points, quelle que soit leur classe.

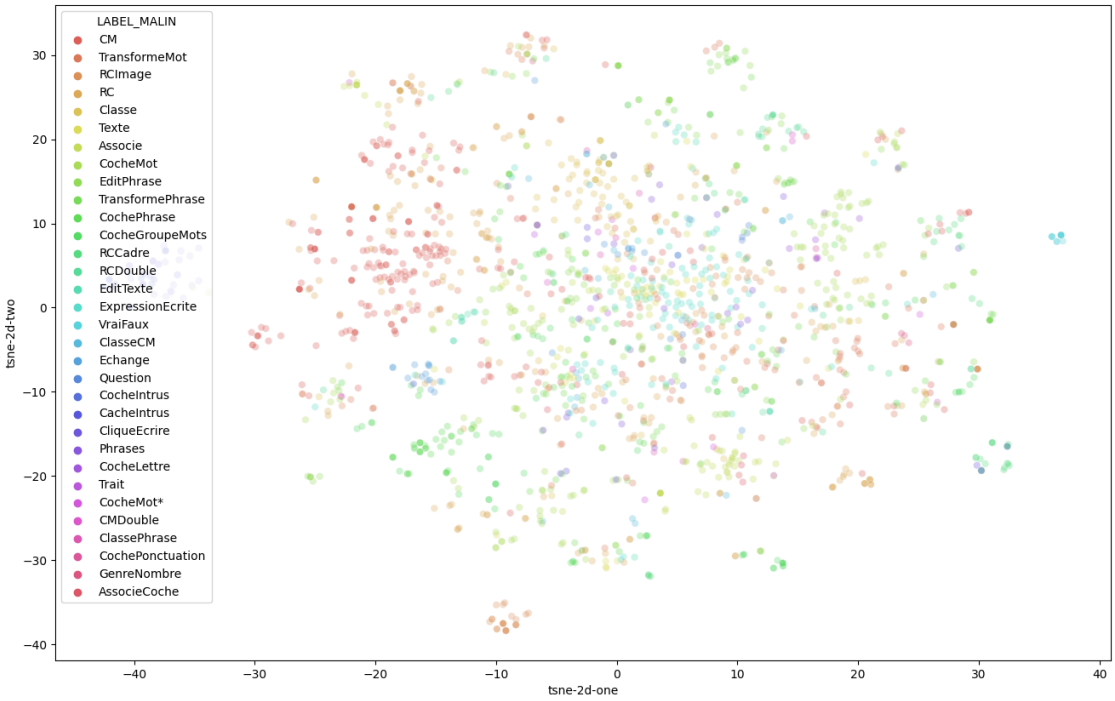


FIGURE 5.3 – Réduction de la dimensionnalité avec les méthodes SVD et t-SNE

5.1.2 Modèles de langage pré-entraînés

Dans la continuité des premières expériences, il s'agit d'améliorer non seulement le modèle en utilisant des méthodes neuronales état de l'art, mais aussi les données en entrée. Pour cela, nous commençons par entraîner des modèles à partir de CamemBERT$_{\text{BASE}}$. .

CamemBERT

Nous entraînons un modèle de langue CamemBERT sur une tâche de classification à partir de nos corpus d'apprentissage et de validation. Pour chaque exercice, les données sont le texte, de la consigne d'une part et de l'énoncé d'autre part, normalisé en minuscule et donné au modèle de manière séparée. La figure 5.4 schématise le modèle.

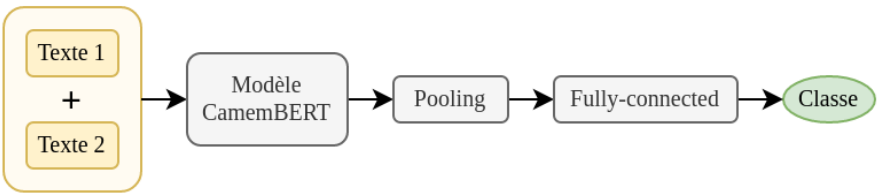


FIGURE 5.4 – Architecture du modèle simple

Les résultats de la classification directe et en cascade avec le modèle CamemBERT sont reportés dans le tableau 5.3.

Expérience	Mesures				
	Exactitude	W Avg Rappel	W Avg Précision	W Avg F1	Macro F1
Classification directe	0.73	0.73	0.73	0.73	0.61
Classification en cascade	**0.74**	**0.74**	**0.74**	0.73	0.61

TABLE 5.3 – Résultats de la classification avec CamemBERT

Par rapport aux systèmes baseline, l'utilisation d'un modèle de langage montre une nette augmentation des scores. L'exactitude atteint 0.73, contre 0.68 avec le SVM, modèle statistique le plus performant. D'autre part, la succession des classifieurs en cascade n'apporte pas d'amélioration par rapport à la classification directe sur les 32 classes.

CamemBERT et enrichissement des plongements

Dans l'objectif d'améliorer la classification, nous expérimentons un enrichissement des données à partir d'informations visuelles et statistiques obtenues lors de l'extraction des exercices depuis le document original.

En effet, les manuels scolaires se caractérisent par une mise en page particulière et la structure même d'un exercice varie d'un exercice à l'autre. D'après les statistiques sur le corpus reportées en section 3.3, les caractéristiques suivantes extraites paraissent pertinentes pour la classification :

— le nombre de phrases dans la consigne et l'énoncé ;
— le nombre de mots dans la consigne et l'énoncé ;
— le nombre de polices différentes utilisées dans l'exercice entier, la consigne et l'énoncé ;
— si la consigne ou l'énoncé contiennent une liste ;
— si une image est associée à l'exercice et l'importance de cette image : indispensable, informative, inutile, pas d'image.

Nous proposons deux méthodes pour intégrer ces caractéristiques aux entrées des modèles.

La première consiste à extraire les plongements de CamemBERT et y ajouter à la suite les nouvelles caractéristiques. Pour un exercice, le vecteur de CamemBERT de dimension $d = 768$ est obtenu en calculant la moyenne des plongements de tous les tokens de la dernière couche cachée du modèle. D'autre part, les informations statistiques sont représentées sous forme de vecteur one-hot de dimension $d = 12$. Le vecteur final de dimension $d = 780$ est obtenu par concaténation des deux vecteurs. Ce vecteur passe ensuite à travers un perceptron multicouche pour la classification.

La deuxième méthode consiste à intégrer les caractéristiques visuelles et statistiques sous forme de texte directement à la suite de l'exercice, séparées par le token spécial $[SEP]$. La chaîne de traitement est ensuite la même que pour les premières expériences : le nouveau texte passe dans le modèle CamemBERT pour vectorisation et entraînement de la classification.

La première méthode s'avère inefficace. Les résultats, reportés dans la table 5.4 ne sont pas comparables à nos systèmes baseline. Cela peut être dû au déséquilibre entre les classes, le modèle semble négliger les classes les moins représentées du jeu de données et sur-apprendre sur les classes les plus fournies. De plus, les plongements

de CamemBERT sont un type de plongements contextualisés, tandis que les vecteurs enrichis à partir de ces plongements contextualisés passent dans un modèle construit pour des plongements non-contextuels.

Expérience	Mesures				
	Exactitude	W Avg Rappel	W Avg Précision	W Avg F1	Macro F1
Classification directe	0.37	0.33	0.37	0.34	0.20
Classification en cascade	0.24	0.23	0.24	0.21	0.11

TABLE 5.4 – Résultats de la classification avec les plongements de CamemBERT enrichis

Les résultats obtenus avec la deuxième méthode sont reportés dans le tableau 5.5. Globalement, ils sont similaires à ceux des premières expériences avec les données textuelles de l'exercice non enrichies. Les scores atteints avec la succession de classifieurs en cascade sont légèrement supérieurs aux autres, avec une exactitude et des moyennes pondérées de la précision et du rappel à 0.75.

Expérience	Mesures				
	Exactitude	W Avg Rappel	W Avg Précision	W Avg F1	Macro F1
Classification directe	0.71	0.72	0.71	0.72	0.57
Classification en cascade	**0.75**	**0.75**	**0.75**	**0.75**	**0.60**

TABLE 5.5 – Résultats de la classification avec CamemBERT et les entrées enrichies avec le token [SEP]

Cette tentative d'exploiter les caractéristiques visuelles n'est pas probante mais les informations sur la structure des exercices ne sont pas à négliger. Cela nous laisse espérer des progrès avec des modèles multimodaux pré-entraînés sur la structure.

CamemBERT et variation de l'architecture

Les documents de notre corpus étant constitués de plusieurs textes, une consigne et un énoncé, nous proposons de faire varier l'architecture du modèle CamemBERT. Le modèle peut être simple, double, ou siamois. Les modèles simples ayant produit des résultats supérieurs à ceux obtenus avec des classifieurs statistiques, une architecture plus complexe pourrait encore les améliorer.

Le modèle double, schématisé en figure 5.5, consiste en deux modèles séparés, l'un apprend sur le premier texte (la consigne), l'autre apprend sur le deuxième (l'énoncé). Les sorties des deux modèles sont ensuite fusionnées de sorte à obtenir une unique classe en sortie.

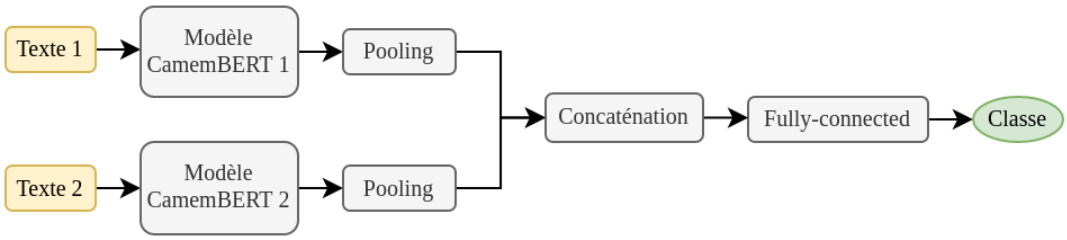


FIGURE 5.5 – Architecture du modèle double

Le modèle siamois, schématisé en figure 5.6, prend également deux textes en entrée. Dans notre cas, la consigne d'une part et l'énoncé de l'exercice d'autre part. Il s'agit de deux entrées distinctes qui passent simultanément dans le même modèle CamemBERT.

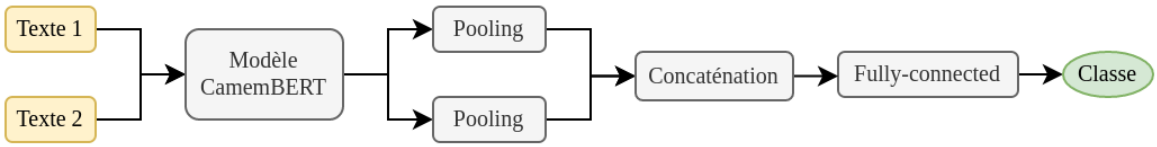


FIGURE 5.6 – Architecture du modèle siamois

Les résultats obtenus sur le corpus de texte par chacun des classifieurs sont reportés dans le tableau 5.6.

Expérience	Mesures				
	Exactitude	W Avg Rappel	W Avg Précision	W Avg F1	Macro F1
Classification directe					
Modèle simple	0.73	0.73	0.73	0.73	0.61
Modèle double	0.74	0.75	0.74	0.74	0.60
Modèle siamois	0.74	0.74	0.74	0.73	0.58
Classification en cascade					
Modèle simple	0.74	0.74	0.74	0.73	0.61
Modèle double	0.74	0.75	0.74	0.74	0.62
Modèle siamois	**0.75**	**0.75**	**0.75**	**0.74**	**0.63**

TABLE 5.6 – Résultats de la classification avec CamemBERT

Dans l'ensemble, les scores obtenus par les classifieurs sont similaires quelle que soit l'architecture du réseau. On constate une légère amélioration au niveau de l'exactitude et de la moyenne pondérée de la f-mesure avec les modèles doubles et siamois, mais la différence n'est pas significative.

Fine-tuning du modèle de langue sur des manuels scolaires

Le pré-entraînement des modèles BERT consiste en deux tâches : un modèle de langue masqué et une prédiction de la phrase suivante. Dans l'objectif d'adapter le modèle à la spécificité de notre corpus, nous envisageons un fine-tuning du modèle de langue masqué de $\text{CamemBERT}_{\text{BASE}}$. A partir de phrases extraites de manuels scolaires, le modèle apprend à prédire des tokens masqués de façon aléatoire.

Pour procéder au fine-tuning du modèle de langue masqué de $\text{CamemBERT}_{\text{BASE}}$, nous utilisons les phrases extraites des ensembles suivants :

— les leçons et exercices des trois manuels annotés, hormis les exercices présents dans les corpus de développement ou de test ;
— le contenu d'un quatrième manuel de français, de niveau CE2 ;
— les *Fantastiques Exercices*, ensemble d'exercices réalisés par *Le Cartable Fantastique* ;
— le corpus *Alector* composé de 79 textes de niveau CE1-CE2-CM1, construit dans le cadre d'un projet ANR d'aide à la lecture pour enfants dyslexiques et faibles lecteurs [Gala et al., 2020].

Les phrases extraites ont été filtrées de sorte à omettre les numéros d'exercices, d'éventuelles erreurs d'extraction, ainsi que le contenu des pages de garde, de sommaire et de références.

Ce modèle CamemBERT fine-tuné est entraîné sur une tâche de classification. Nous entraînons plusieurs modèles sur notre corpus, avec chacune des trois architectures (simple, double, siamoise), en classification directe et avec un enchaînement cascadé de classifieurs. Le tableau 5.7 reporte les résultats obtenus de chaque classification sur le corpus de test.

Expérience	Mesures				
	Exactitude	W Avg Rappel	W Avg Précision	W Avg F1	Macro F1
Classification directe					
Modèle simple	**0.77**	0.76	**0.77**	**0.76**	0.63
Modèle double	0.75	0.75	0.75	0.75	0.62
Modèle siamois	0.75	0.75	0.75	0.74	0.59
Classification en cascade					
Modèle simple	0.76	**0.77**	0.76	**0.76**	**0.68**
Modèle double	0.75	0.74	0.75	0.74	0.60
Modèle siamois	0.74	0.75	0.74	0.74	0.60

TABLE 5.7 – Résultats de la classification avec CamemBERT fine-tuné sur les manuels scolaires

D'après nos expérimentations, le fine-tuning du modèle de langue apporte une augmentation allant jusqu'à 0.04 sur toutes les métriques et pour la quasi-totalité des modèles.

En outre, cette expérience confirme l'inefficacité de la variation de l'architecture de BERT. Les modèles simples, où plusieurs textes distincts sont fournis en une même entrée, sont tout autant voire plus performants que les modèles doubles ou siamois.

5.1.3 Modèles pré-entraînés sur la structure des documents

Le modèle LayoutLMv2ForSequenceClassification est une extension de LayoutLMv2 pré-entraînée sur une tâche de classification. Elle peut traiter des données constituées d'images uniquement, en appliquant une océrisation. Nous ignorons cette étape et fournissons directement les images et le contenu textuel accompagné des coordonnées spatiales de chaque token. Les entrées sont préparées et fournies au modèle pour entraînement, puis évaluation. Les résultats sont reportés dans le tableau 5.8.

Expérience	Mesures				
	Exactitude	W Avg Rappel	W Avg Précision	W Avg F1	Macro F1
Classification directe	**0.77**	**0.76**	**0.77**	**0.76**	**0.59**
Classification en cascade	0.75	0.75	0.75	0.75	0.54

TABLE 5.8 – Résultats de la classification avec LayoutLMv2

Les résultats sont comparables à ceux obtenus avec des transformers CamemBERT. Les micro scores et les moyennes pondérées sont globalement similaires, tandis que la macro f-mesure est légèrement inférieure avec LayoutLM. Ces résultats sont toutefois très satisfaisants, LayoutLM n'étant adapté ni au français, ni au contenu de manuels scolaires. La classification multimodale est une piste à explorer pour exploiter au mieux les caractéristiques visuelles et structurelles de notre corpus.

5.1.4 Classification multimodale : fusion tardive

Nous proposons enfin une dernière méthode reposant sur la fusion tardive, orientée sur les prédictions. Les scores des deux classifieurs les plus performants sont fusionnés : ceux entraînés sur CamemBERT et LayoutLMv2. Pour CamemBERT, il s'agit en l'occurrence des modèles fine-tunés sur le texte des manuels scolaires et avec une architecture simple.

Nous expérimentons, à l'aide de l'outil *ranx* [Bassani, 2022], les algorithmes de fusion CombSUM et Weighted Sum [Fox and Shaw, 1993], des combinaisons linéaires basées sur les scores.

Avant la fusion, les scores de prédiction des classifieurs sont normalisés. La normalisation Min-Max, proposée par [Lee, 1997], ramène les scores s entre 0 et 1 en conservant les distances.

$$\text{Min-Max}(s) = \frac{s - s_{\text{min}}}{s_{\text{max}} - s_{\text{min}}}$$

Soient s_1 et s_2 les scores prédits par les deux classifieurs pour un passage donné puis normalisés, et s_{mixed} le score obtenu par fusion.

Weighted Sum Fusion calcule la somme pondérée des scores s_1 et s_2.

$$s_{mixed} = \alpha \times s_1 + \beta \times s_2$$

où α, β sont les poids et généralement $\beta = 1 - \alpha$. Les poids sont calculés sur le corpus de validation.

CombSUM est un cas particulier de la méthode Weighted Sum Fusion où $\alpha = \beta = 0.5$. Cela revient à additionner les scores de chacune des modalités.

$$s_{mixed} = s_1 + s_2$$

Dans les deux cas, la prédiction finale est la classe pour laquelle le score fusionné est le plus haut.

Le tableau 5.9 reprend les scores résultant de la classification avec CamemBERT et LayoutLMv2, et présente ceux obtenus après fusion des prédictions.

Expérience	Mesures				
	Exactitude	W Avg Rappel	W Avg Précision	W Avg F1	Macro F1
Classification directe					
CamemBERT seul	0.77	0.76	0.77	0.76	0.63
LayoutLMv2 seul	0.77	0.76	0.77	0.76	0.59
Fusion CombSUM	**0.80**	**0.79**	**0.80**	**0.79**	0.63
Fusion Weighted SUM	0.77	0.76	0.77	0.76	0.63
Classification en cascade					
CamemBERT seul	0.76	0.77	0.76	0.76	0.68
LayoutLMv2 seul	0.75	0.75	0.75	0.75	0.54
Fusion CombSUM	**0.78**	**0.79**	**0.78**	**0.78**	0.66
Fusion Weighted SUM	0.76	0.77	0.76	0.76	0.68

TABLE 5.9 – Résultats de la classification avec fusion tardive de CamemBERT et LayoutLMv2

L'algorithme de fusion CombSUM a permis d'atteindre 0.80 d'exactitude et de précision, contre 0.77 sans fusion. Étonnamment, la fusion Weighted SUM produit des résultats inférieurs à la fusion CombSUM. Pourtant, les poids attribués à chaque classifieur sont optimisés à l'aide du corpus de validation. Cela témoigne à nouveau de la diversité des données, même au sein d'une même classe. Finalement, la fusion tardive a permis une augmentation des scores significative.

Dans une perspective d'amélioration, nous pouvons explorer d'autres méthodes de fusion tardive, mais aussi de fusion précoce.

La matrice de confusion obtenue avec la meilleure méthode expérimentée, la fusion CombSUM des scores des classifieurs CamemBERT fine-tuné et LayoutLMv2, est visualisable en figure 5.7. Elle montre que la plupart des erreurs concernent encore les classes sous-représentées, pour lesquelles on ne dispose que de quelques exemples d'exercices. On constate toutefois que l'amélioration apportée par le finetuning et la fusion des classifieurs est valable pour l'ensemble des classes.

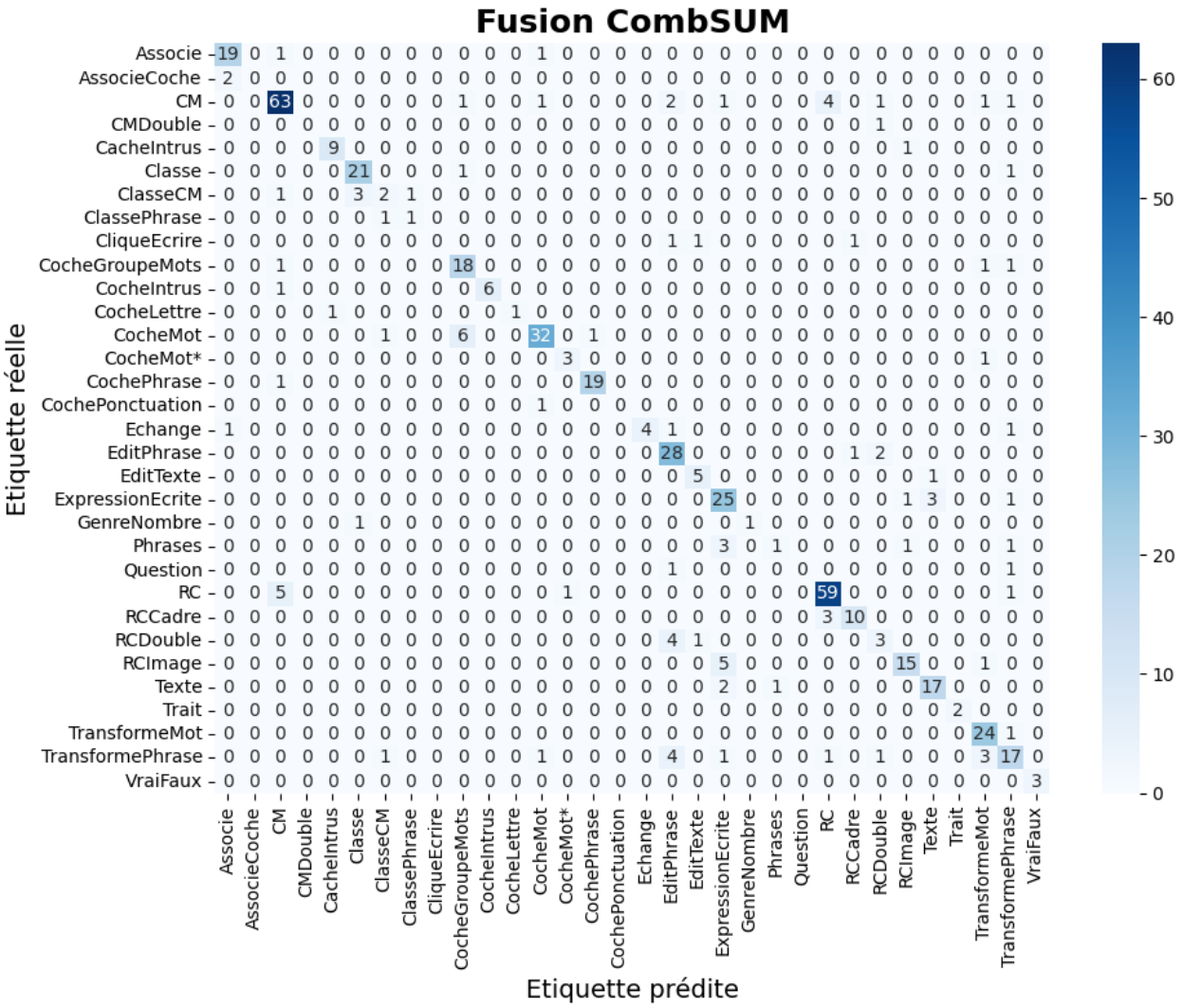


FIGURE 5.7 – Matrice de confusion

5.2 Augmentation de données

Lors de l'évaluation des modèles et quelle que soit la méthode utilisée, nous observons des scores bas voire nuls pour les classes sous-représentées. Pour faire face au déséquilibre du jeu de données et particulièrement au problème que posent ces classes, nous proposons de générer des données artificielles en complément des données d'apprentissage. Toutefois, certaines classes comptent moins de dix exercices, ce qui est insuffisant pour générer automatiquement des données artificielles de qualité. Nous extrayons alors manuellement des exercices provenant d'autres manuels scolaires afin d'avoir pour chaque classe un échantillon d'un minimum de dix exercices.

Compte tenu de la diversité des exercices au sein d'une même classe, le volume d'exercices par classe semble encore insuffisant pour fine-tuner des modèles de langues sur une tâche de génération de textes. Avec davantage de données annotées, de telles approches neuronales pourront être envisagées.

En l'état actuel, nous testons dans cette section différentes autres approches d'augmentation de données.

Le cross-over consiste à scinder les données en deux moitiés et à échanger des moitiés de données ayant le même label. Cette méthode peut s'appliquer entre deux consignes, entre deux énoncés, ou entre deux exercices, en regroupant la consigne de l'un avec l'énoncé de l'autre. En effet, les consignes sont pour la plupart constituées d'une seule phrase, parfois courte. Même au sein d'une même classe, la structure et le nombre d'éléments d'un énoncé sont aussi variables, il peut par exemple s'agir d'un seul texte comme de listes de mots. L'association d'une consigne d'un exercice avec l'énoncé d'un autre exercice semble l'option la plus adéquate, mais la difficulté réside dans le maintien de la correspondance entre une consigne et son énoncé. [Luque, 2019] utilise le cross-over sur des tweets pour l'analyse de sentiment et soutient l'hypothèse que le tweet résultant préservera le même sentiment, même s'il est agrammatical ou asémantique. L'association d'une consigne à un énoncé qui ne lui correspond pas devrait alors quand même préserver les spécificités de la classe. En revanche, nos données étant pour la plupart plus longues qu'un tweet, cette technique pourrait produire des répétitions d'énoncés qui empêcheront la généralisation des modèles.

Nous essayons également de générer des exercices par rétro-traduction. Chaque phrase est traduite du français vers l'anglais, puis la traduction anglaise est retraduite vers le français. L'implémentation se fait avec Marian et les modèles Opus-MT de Helsinki-NLP, entraînés sur OPUS, un ensemble de corpus parallèles.

Enfin, la génération de données par substitution lexicale consiste à remplacer un mot par un autre mot sémantiquement proche, par exemple un synonyme ou un hyperonyme. Pour cela, il existe des bases lexicales. La plupart est inspirée de WordNet, une base de données lexicale en langue anglaise développée depuis 1985 par [Miller et al., 1990, Miller, 1995], ayant pour but de répertorier le contenu sémantique et lexical de la langue. C'est un réseau sémantique de plus de 200 000 mots qui repose sur des synsets, des groupes de mots reliés par des relations sémantiques. Pour le français, WoNef [Pradet et al., 2014] est la traduction française de WordNet. WOLF (WordNet Libre du Français) est construit sur le modèle de WordNet par [Sagot and Fišer, 2008]. Il existe également ReSyf [Billami et al., 2018], une

ressource pédagogique où les synonymes sont classés par difficulté. Cette dernière ressource, utilisée pour la simplification de textes dans un contexte d'accessibilité de la lecture aux dyslexiques, semble plus adaptée à nos données.

La base de données lexicale libre Manulex [Lété et al., 2004, Lété, 2004] s'inscrit particulièrement dans notre sujet de recherche. Elle fournit les fréquences d'occurrences de 48 900 formes orthographiques dans 54 manuels scolaires en usage dans l'école élémentaire. Les fréquences d'occurrences sont calculées en fonction du niveau : le CP, le CE1 et le cycle 3 (CE2-CM2). Pour chaque entrée, quatre indices sont donnés :

— F, la fréquence brute du mot dans le corpus ;
— D, l'indice de dispersion du mot parmi les manuels ;
— U, la fréquence par million estimée à partir de D ;
— IFC, un indice de fréquence courant calculé à partir de U par transformation logarithmique.

Nous proposons de mettre en œuvre la substitution lexicale sur les exercices de notre corpus en recherchant des synonymes des noms, verbes et adjectifs dans la base ReSyf, puis en filtrant les candidats avec Manulex, à partir de l'indice de fréquence courant du lemme dans l'ensemble des niveaux scolaires.

Si la recherche de candidats synonymes est simple à mettre en œuvre, une étape de désambiguïsation pourrait permettre de sélectionner le candidat le plus pertinent compte tenu du contexte. Puis, après substitution des mots par leurs synonymes dans le texte original, une étape de nettoyage est nécessaire : les déterminants et adjectifs doivent être accordés au nom qu'ils complètent et les verbes conjugués au bon temps et à la bonne personne. La désambiguïsation des candidats n'est pas implémentée dans notre système mais reste une piste à ne pas écarter. Des traitements de base de re-grammaticalisation de la phrase sont appliqués, comme la conjugaison des verbes, accord des déterminants et des noms. Ces traitements reposent sur des règles et sur l'analyse des phrases originales par des librairies classiques de TAL. Des erreurs persistent, un nettoyage plus approfondi est envisagé.

Des exemples d'exercices artificiels générés avec les trois méthodes sont présentés dans le tableau annexe A.3.

Les résultats de la génération sur les énoncés sont acceptables, mais les consignes sont beaucoup trop éloignées des consignes originales. En effet, le lexique des énoncés est varié, mais la terminologie des consignes est très spécifique et les termes peuvent difficilement être substitués. Les noms qui renvoient à l'énoncé sont : « *lettres* », « *syllabes* », « *mots* », « *phrases* », « *texte* », « *tableau* », « *image* ». Les verbes d'action peuvent être, selon la classe : « *recopier* », « *entourer* », « *conjuguer* », « *tracer* », « *associer* ». De plus, des termes qui se rapportent à la grammaire, l'orthographe ou la conjugaison sont irremplaçables, comme « *genre* », « *nombre* », « *masculin* », « *féminin* », « *singulier* », « *pluriel* », « *infinitif* », « *présent* », « *imparfait* », « *radical* », « *terminaison* », « *préfixe* », « *suffixe* »...etc. Une approche par règles à partir d'une terminologie des consignes peut être envisageable. La plus grande difficulté est le maintien de la correspondance entre la consigne et son énoncé et surtout entre l'exercice et sa classe. De plus, l'évaluation de la qualité des données générées requiert une expertise du domaine.

Conclusion

Les expérimentations conduites sur notre corpus d'exercices ont permis la comparaison de divers modèles de classification et de génération de données.

Globalement, sur une tâche de classification, les méthodes neuronales se valent, mais sont plus efficaces que les algorithmes traditionnels. En outre, une comparaison de différentes variantes de CamemBERT montre que la variation de l'architecture du réseau et l'enchaînement cascadé de plusieurs classifieurs n'améliorent pas significativement les performances, tandis que le fine-tuning du modèle de langue apporte une augmentation des scores pour tous les modèles. Les scores les plus hauts sont ainsi obtenus avec un modèle fine-tuné sur le corpus de manuels scolaires et un unique classifieur d'architecture simple. Aussi, les résultats des classifieurs utilisant des modèles pré-entraînés sur des données multimodales confirment l'importance d'optimiser l'apprentissage en exploitant les modalités textuelles et visuelles des documents. Une étude plus approfondie des différentes techniques de préparation des données, des paramètres des classifieurs, et des méthodes de fusion promet également une amélioration des résultats.

D'autre part, l'ajout de documents artificiels au corpus d'apprentissage devrait contrer le déséquilibre entre les classes et permettre une amélioration des classifieurs. Cependant, les trois méthodes de génération de données mises en œuvre ne sont pas à la hauteur de nos attentes. Le langage des exercices s'avère être très spécifique et la principale difficulté concerne les consignes. Des expérimentations plus poussées et ajustées au langage des manuels scolaires devraient répondre à nos besoins.

CHAPITRE 6

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Dans ce mémoire, nous avons traité de la problématique de la classification d'exercices de manuels scolaires selon leur type d'adaptation pour les élèves dyspraxiques.

Ce sujet s'inscrit dans un projet concret au profit d'une cause encore trop souvent négligée. Bien qu'il existe des travaux sur l'inclusion scolaire et l'introduction d'outils numériques en classe, peu d'actions sont mises en place. En l'occurrence, en traitement automatique des langues, peu de recherches s'intéressent au contenu de manuels scolaires. L'automatisation de l'adaptation des manuels réalisée dans le cadre du projet MALIN, constitue ainsi une première avancée dans ce domaine, et surtout un sujet qui peut encore être développé.

6.1 Contributions

Après avoir dressé un état de l'art des méthodes de classification et de génération de données, nous avons défini une chaîne de traitements, construit un corpus d'exercices et réalisé un ensemble d'expériences.

Les exercices étant essentiellement constitués de texte, nous avons en premier lieu sélectionné plusieurs méthodes de classification de texte, statistiques et neuronales, jugées pertinentes compte tenu de notre jeu de données. Nous avons par la suite expérimenté une méthode multimodale reposant non pas uniquement sur le contenu textuel des documents, mais aussi sur leur structure. Si les méthodes statistiques ont permis de fournir une baseline solide, le passage aux transformers montre une nette évolution des résultats. D'après nos expérimentations, les classifieurs neuronaux de type Transformers sont les plus efficaces, qu'ils traitent davantage la modalité textuelle ou visuelle. Le score d'exactitude des meilleurs modèles oscille entre 0.74 et 0.77. L'implémentation d'une fusion tardive nous a permis d'atteindre 0.80 d'exactitude. Compte tenu de la spécificité et de l'hétérogénéité du corpus, ces résultats sont très satisfaisants.

Certaines classes de notre jeu de données sont particulièrement sous-représentées, nous nous sommes donc intéressés à la génération de données artificielles afin d'augmenter le corpus d'apprentissage. Trois techniques de génération de données ont ainsi été appliquées aux classes les moins fournies du corpus. La qualité des exercices artificiels générés est acceptable pour les énoncés, mais moindre pour les consignes en raison de la spécificité de leur lexique ; il reste beaucoup de progrès à faire, ce qui témoigne une fois de plus de la particularité linguistique de notre corpus.

6.2 Perspectives

La mise en œuvre de différents modèles d'apprentissage automatique a permis une comparaison de techniques et montre une progression des résultats encourageante. Toutefois, les pistes de recherches explorées dans ce mémoire devront être approfondies pour permettre une adaptation de qualité pour le projet MALIN. En effet, les manuels adaptés ne doivent comporter aucune erreur. Si une étape de relecture par des experts sera nécessaire, il s'agit de mettre en place le meilleur système possible pour alléger leur tâche.

6.2.1 Classification

Les performances des classifieurs proposés sont satisfaisants. Cependant, les résultats semblent stagner et n'atteignent pas plus de 0.77 d'exactitude sans fusion.

Dans une perspective d'amélioration, les modèles multimodaux comme LayoutLM, entraînés à la fois sur la structure et le texte, sont une piste à approfondir. En l'occurrence, la fusion tardive que nous avons expérimentée a déjà permis de meilleures performances. Nous envisageons alors d'implémenter une méthode multimodale reposant sur une fusion précoce, exploitant le texte, l'image, la position, ainsi que les informations de mise en page.

6.2.2 Jeu de données

Nous faisons face à un ensemble de données très varié, mais surtout déséquilibré. En l'état, cette problématique de manque de données ralentit le processus d'automatisation de l'adaptation et nous contraint à maintenir un travail de vérification manuel conséquent.

Le jeu d'étiquettes mis au point n'étant pas figé, nous pourrions modifier ou fusionner certaines classes, et selon le cas ajouter quelques traitements supplémentaires à la phase d'adaptation.

En revanche, cela restera probablement insuffisant. L'augmentation de données artificielles semble alors indispensable, mais constitue un réel challenge particulièrement pour la génération de consignes. A ce stade, il est nécessaire de concevoir des approches par règle spécialement pour de telles données. Avec davantage de données annotées, nous envisageons d'explorer les méthodes neuronales de génération automatique de texte, la fusion de plusieurs méthodes de génération et le méta-learning, mais aussi l'auto-apprentissage. Cette solution utilisant un ensemble de données étiquetées et non étiquetées, augmente le corpus d'entraînement à partir des prédictions les plus fiables faites sur les documents non étiquetés.

6.2.3 Généralisation

Enfin, nos expériences ne ciblent qu'une partie de tout un ensemble d'exercices de manuels scolaires.

D'une part, nous avons d'emblée omis les exercices considérés atypiques ou non adaptables par les annotateurs. Ceux-ci, non adaptés aux enfants dyspraxiques ou trop particuliers pour être adaptés automatiquement, devront nécessairement être traités manuellement. Afin de réduire au maximum le travail manuel, une classification en amont est envisagée. L'objectif est de séparer ces exercices spécifiques des autres exercices à adapter de manière automatique.

D'autre part, nombreux sont les exercices qui demandent la réalisation de plusieurs tâches. Ces exercices, annotés en deux, trois, voire quatre classes, constituent un obstacle dans la tâche de classification. La classification multi-labels n'a pas été abordée dans ce travail mais s'avère indispensable pour la suite de l'étude.

Enfin, notre travail porte uniquement sur les exercices de français de niveau élémentaire et sur l'adaptation pour les élèves dyspraxiques. Il n'est actuellement pas transposable à d'autres matières ou d'autres handicaps. A plus long terme, il s'agira de proposer un système d'adaptation applicable à toutes les matières enseignées, tous les niveaux, et d'autres handicaps.

BIBLIOGRAPHIE

[Alva-Manchego et al., 2020] Alva-Manchego, F., Scarton, C., and Specia, L. (2020). Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1). – Cité page 17.

[AUDIAU, 2009] AUDIAU, A. (2009). L'information pour tous. règles européennes pour une information facile à lire et à comprendre. rapport interne. *Nous aussi, UNAPEI*. – Cité page 18.

[Bakkali et al., 2020] Bakkali, S., Ming, Z., Coustaty, M., and Rusiñol, M. (2020). Visual and textual deep feature fusion for document image classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. – Cité page 26.

[Bassani, 2022] Bassani, E. (2022). ranx: A blazing-fast python library for ranking evaluation and comparison. In *Proceedings of the 44th European Conference on Information Retrieval Research (ECIR)*, volume 13186 of *Lecture Notes in Computer Science*. – Cité page 59.

[Bayer et al., 2022] Bayer, M., Kaufhold, M., and Reuter, C. (2022). A survey on data augmentation for text classification. *ACM Computing Surveys*. – Cité pages 26 et 27.

[Belinkov and Bisk, 2019] Belinkov, Y. and Bisk, Y. (2019). Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*. – Cité page 26.

[Billami et al., 2018] Billami, M. B., François, T., and Gala, N. (2018). ReSyf: a French lexicon with ranked synonyms. In *Proceedings of the 27th International Conference on Computational Linguistics (ICCL)*. – Cité page 61.

[Bloehdorn and Hotho, 2004] Bloehdorn, S. and Hotho, A. (2004). Boosting for text classification with semantic features. volume 3932. – Cité page 23.

[Bojanowski et al., 2017] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics (TACL)*, 5. – Cité page 21.

[Brouwers et al., 2014] Brouwers, L., Bernhard, D., Ligozat, A.-L., and François, T. (2014). Syntactic sentence simplification for french. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*. – Cité page 17.

[Canning and Tait, 1999] Canning, Y. and Tait, J. (1999). Syntactic simplification of newspaper text for aphasic readers. In *ACM SIGIR'99 Workshop on Customised Information Delivery*. – Cité page 18.

[Cardon and Grabar, 2020] Cardon, R. and Grabar, N. (2020). French biomedical text simplification: When small and precise helps. In *Proceedings of the 28th International Conference on Computational Linguistics (ICCL)*. – Cité page 17.

[Carroll et al., 1999] Carroll, J. A., Minnen, G., Pearce, D., Canning, Y., Devlin, S., and Tait, J. (1999). Simplifying text for language-impaired readers. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (ACL)*. – Cité page 18.

[Cemri et al., 2022] Cemri, M., Çukur, T., and Koç, A. (2022). Unsupervised simplification of legal texts. *CoRR*. – Cité page 17.

[Chen et al., 2016] Chen, T., Xu, R., and Wang, X. (2016). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications*. – Cité page 23.

[Claveau et al., 2021] Claveau, V., Chaffin, A., and Kijak, E. (2021). La génération de textes artificiels en substitution ou en complément de données d'apprentissage. In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles (TALN). Volume 1 : conférence principale*. – Cité page 27.

[Coulombe, 2018] Coulombe, C. (2018). Text data augmentation made simple by leveraging NLP cloud apis. *CoRR*, abs/1812.04718. – Cité page 26.

[De Mántaras, 1991] De Mántaras, R. (1991). A distance-based attribute selection measure for decision tree induction. *Machine Learning*, 6. – Cité page 22.

[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 (NAACL-HLT). Volume 1 : Long and Short Papers*. – Cité pages 21 et 25.

[Dieng et al., 2017] Dieng, A. B., Wang, C., Gao, J., and Paisley, J. (2017). TopicRNN: A recurrent neural network with long-range semantic dependency. In *International Conference on Learning Representations (ICLR)*. – Cité page 24.

[Dwivedi and Arya, 2016] Dwivedi, S. K. and Arya, C. (2016). Automatic text classification in information retrieval: A survey. *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies (ICTCS)*. – Cité page 23.

[Edunov et al., 2018] Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. – Cité page 27.

[Fabre et al., 2014] Fabre, C., Hathout, N., Ho-Dac, L.-M., Morlane-Hondère, F., Muller, P., Sajous, F., Tanguy, L., and van de Cruys, T. (2014). Présentation de l'atelier SemDis 2014 : sémantique distributionnelle pour la substitution lexicale et l'exploration de corpus spécialisés. In *Actes de la 21e Conférence sur le Traitement Automatique des Langues Naturelles (TALN)*. – Cité page 27.

[Fox and Shaw, 1993] Fox, E. A. and Shaw, J. A. (1993). Combination of multiple searches. In *Proceedings of the Second Text REtrieval Conference (TREC)*. – Cité page 59.

[Frank and Bouckaert, 2006] Frank, E. and Bouckaert, R. (2006). Naive bayes for text classification with unbalanced classes. In *Proceedings of the 10th European conference on Principle and Practice of Knowledge Discovery in Databases (ECMLPKDD)*. – Cité page 22.

[Gábor, 2014] Gábor, K. (2014). The WoDiS system - WOlf and DIStributions for lexical substitution (le système WoDiS - WOLF et DIStributions pour la substitution lexicale) [in French]. In *TALN-RECITAL 2014 Workshop SemDis 2014 : Enjeux actuels de la sémantique distributionnelle*. – Cité page 27.

[Gala et al., 2020] Gala, N., Tack, A., Javourey-Drevet, L., François, T., and Ziegler, J. C. (2020). Alector: A Parallel Corpus of Simplified French Texts with Alignments of Misreadings by Poor and Dyslexic Readers. In *12th Language Resources and Evaluation for Language Technologies (LREC)*. – Cité pages 14 et 57.

[Gasparetto et al., 2022] Gasparetto, A., Marcuzzo, M., Zangari, A., and Albarelli, A. (2022). A survey on text classification algorithms: From text to predictions. *Information*, 13(2). – Cité page 18.

[Hendrycks et al., 2020] Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. (2020). Augmix: A simple data processing method to improve robustness and uncertainty. In *8th International Conference on Learning Representations, (ICLR)*. – Cité page 27.

[Hinton et al., 2014] Hinton, G., Dean, J., and Vinyals, O. (2014). Distilling the knowledge in a neural network. – Cité page 25.

[Ho, 1995] Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition (ICDAR)*, volume 1. – Cité page 22.

[Huron, 2011] Huron, C. (2011). *L'enfant dyspraxique*. 3e edition. – Cité page 14.

[Huron, 2017] Huron, C. (2017). De la souffrance à l'autonomie. *L'école des parents*, 624(3). – Cité page 14.

[Iyyer et al., 2015] Iyyer, M., Manjunatha, V., Boyd-Graber, J., and Daumé III, H. (2015). Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP). Volume 1 : Long Papers*. – Cité page 24.

[Jain and Wigington, 2019] Jain, R. and Wigington, C. (2019). Multimodal document image classification. In *Proceedings of 15th International Conference on Document Analysis and Recognition (ICDAR)*. – Cité page 26.

[Jones, 2004] Jones, K. S. (2004). Idf term weighting and ir research lessons. *Journal of Documentation*, 60. – Cité page 20.

[Kalchbrenner et al., 2014] Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL). Volume 1 : Long Papers*. – Cité page 24.

[Kobayashi, 2018] Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). Volume 2 : Short Papers*. – Cité page 27.

[Kowsari et al., 2019] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D., Id, L., and Barnes (2019). Text classification algorithms: A survey. *Information (Switzerland)*, 10. – Cité pages 18 et 19.

[Kumar et al., 2020] Kumar, V., Choudhary, A., and Cho, E. (2020). Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*. – Cité page 27.

[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 *6e conférence conjointe Journées d'Études sur la Parole (JEP), Traitement Automatique des Langues Naturelles (TALN), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL). Volume 2 : Traitement Automatique des Langues Naturelles*. – Cité page 25.

[Lee, 1997] Lee, J. (1997). Analyses of multiple evidence combination. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. – Cité page 59.

[Lété, 2004] Lété, B. (2004). Chapitre 17 : Manulex : une base de données du lexique écrit adressé aux élèves. *Didactique du lexique. Contextes, démarches, supports.* – Cité page 62.

[Lété et al., 2004] Lété, B., Sprenger-Charolles, L., and Colé, P. (2004). MANULEX: a grade-level lexical database from French elementary school readers. *Behavior Research Methods Instruments and Computers*, 36(1). – Cité page 62.

[Lewis et al., 2006] Lewis, D., Agam, G., Argamon, S., Frieder, O., Grossman, D., and Heard, J. (2006). Building a test collection for complex document information processing. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. – Cité page 44.

[Liu et al., 2017] Liu, J., Chang, W.-C., Wu, Y., and Yang, Y. (2017). Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. – Cité page 24.

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. – Cité page 25.

[Loty and Mazeau, 2020] Loty, G. and Mazeau, M. (2020). *Dys : outils et adaptations dans ma classe - Cycles 2 et 3*. – Cité page 14.

[Luque, 2019] Luque, F. M. (2019). Atalaya at tass 2019: Data augmentation and robust embeddings for sentiment analysis. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing (IberLEF@SEPLN)*. – Cité pages 27 et 61.

[Martin et al., 2020a] Martin, L., de la Clergerie, É. V., Sagot, B., and Bordes, A. (2020a). Controllable sentence simplification. In *12th Language Resources and Evaluation Conference (LREC)*. – Cité page 18.

[Martin et al., 2020b] Martin, L., Müller, B., Suárez, P. J. O., Dupont, Y., Romary, L., de la Clergerie, É. V., Seddah, D., and Sagot, B. (2020b). CamemBERT: a

tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. – Cité pages 25 et 44.

[McCarthy and Navigli, 2007] McCarthy, D. and Navigli, R. (2007). SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. – Cité page 27.

[Melamud et al., 2016] Melamud, O., Goldberger, J., and Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of the 20th Special Interest Group on Natural Language Learning (SIGNLL) Conference on Computational Natural Language Learning (CoNLL)*. – Cité page 21.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NeurIPS)*, volume 2. – Cité page 21.

[Miller, 1995] Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, 38(11). – Cité page 61.

[Miller et al., 1990] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to WordNet: An On-line Lexical Database*. *International Journal of Lexicography*, 3(4). – Cité page 61.

[Minaee et al., 2021] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad Khasmakhi, N., Asgari-Chenaghlu, M., and Gao, J. (2021). Deep learning–based text classification: A comprehensive review. *ACM Computing Surveys*, 54. – Cité pages 18, 24 et 25.

[Pang et al., 2002] Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. – Cité page 23.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. – Cité page 21.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). Volume 1 : Long Papers*. – Cité page 21.

[Powalski et al., 2021] Powalski, R., Borchmann, L., Jurkiewicz, D., Dwojak, T., Pietruszka, M., and Palka, G. (2021). Going full-tilt boogie on document understanding with text-image-layout transformer. In *Proceedings of 16th International Conference on Document Analysis and Recognition (ICDAR)*. – Cité page 26.

[Pradet et al., 2014] Pradet, Q., Chalendar, G. D., and Baguenier-Desormeaux, J. (2014). WoNeF, an improved, expanded and evaluated automatic French translation of WordNet. In *7th Global Wordnet Conference (GWC)*. – Cité page 61.

[Raffel et al., 2020] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning

with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21. – Cité page 25.

[Ratner et al., 2017] Ratner, A. J., Ehrenberg, H. R., Hussain, Z., Dunnmon, J., and Ré, C. (2017). Learning to compose domain-specific transformations for data augmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 30. – Cité page 27.

[Rocchio, 1971] Rocchio, J. J. (1971). Relevance feedback in information retrieval. In *The Smart retrieval system - experiments in automatic document processing*. – Cité page 23.

[Safavian and Landgrebe, 1991] Safavian, S. and Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3). – Cité page 22.

[Sagot and Fišer, 2008] Sagot, B. and Fišer, D. (2008). Construction d'un wordnet libre du français à partir de ressources multilingues. In *Actes de la 15ème conférence sur le Traitement Automatique des Langues Naturelles (TALN). Articles longs*. – Cité pages 27 et 61.

[Sanh et al., 2019] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*. – Cité page 25.

[Scarton and Specia, 2018] Scarton, C. and Specia, L. (2018). Learning simplifications for specific target audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL). Volume 2 : Short Papers*. – Cité page 18.

[Schapire, 1990] Schapire, R. (1990). The strength of weak learnability. *Machine Learning*, 5(2). – Cité page 23.

[Siddharthan, 2014] Siddharthan, A. (2014). A survey of research on text simplification. *ITL-International Journal of Applied Linguistics (InJAL)*, 165(2). – Cité page 17.

[Somya et al., 2016] Somya, B. J., Chetan, and Srinivasa, K. (2016). Large scale multi-label text classification of a hierarchical dataset using rocchio algorithm. *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. – Cité page 23.

[Sutton and McCallum,] Sutton, C. and McCallum, A. – Cité page 23.

[Tai et al., 2015] Tai, K., Socher, R., and Manning, C. (2015). Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP). Volume 1 : Long Papers*. – Cité page 24.

[Todirascu et al., 2022] Todirascu, A., Wilkens, R., Rolin, E., François, T., Bernhard, D., and Gala, N. (2022). HECTOR: A Hybrid tExt simplifiCation TOol for Raw texts in french. In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*. – Cité page 18.

[van der Maaten and Hinton, 2008] van der Maaten, L. and Hinton, G. (2008). Viualizing data using t-sne. *Journal of Machine Learning Research*, 9. – Cité page 53.

[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 *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30. – Cité page 24.

[Wu et al., 2019] Wu, X., Lv, S., Zang, L., Han, J., and Hu, S. (2019). Conditional BERT contextual augmentation. In *Proceedings of the 19th International Conference on Computational Science (ICCS), Part IV*. – Cité page 27.

[Wubben et al., 2012] Wubben, S., Van Den Bosch, A., and Krahmer, E. (2012). Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL). Volume 1 : Long Papers*. – Cité page 17.

[Xu et al., 2020] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., and Zhou, M. (2020). LayoutLM: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (KDD)*. – Cité page 25.

[Xu et al., 2021] Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., and Zhou, L. (2021). LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL) and the 11th International Joint Conference on Natural Language Processing (IJCNLP). Volume 1 : Long Papers*. – Cité pages 26 et 44.

[Yang et al., 2015] Yang, Y., Yih, W.-t., and Meek, C. (2015). Wikiqa: A challenge dataset for open-domain question answering. – Cité page 31.

[Yang et al., 2019] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*. – Cité page 25.

[Zhang et al., 2007] Zhang, L., Marron, J., Shen, H., and Zhu, Z. (2007). Singular value decomposition and its visualization. *Journal of Computational and Graphical Statistics (JCGS)*, 16. – Cité page 52.

[Zhang et al., 2015] Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, volume 1. – Cité page 26.

ANNEXES

A.1 Extrait du corpus

Exemple d'un exercice de CE2

Listing A.1 – Extrait du corpus : exercice extrait au format XML

```
<section type="exercice" id="p173_e3" VMIN="136" HMIN="40" VMAX="243" HMAX="267">
  <Sentence id="p173_s3" CONTENT=" 3" majFont="font95" VMIN="136.726" HMIN="45.4489"
      VMAX="147.6087" HMAX="51.941100000000006" isBeginList="false" isList="false"
      categorie="numExo" section="exercice">
    <String id="p175_w3" CONTENT="3" HPOS="45.4489" VPOS="136.726" WIDTH="6.4922"
        HEIGHT="10.8827" STYLEREFS="font95" spNextWord="0"/>
  </Sentence>
  <Sentence id="p173_s4" CONTENT=" Écris chaque liste sans intrus." majFont="font356"
      VMIN="137.614" HMIN="75.368" VMAX="149.489" HMAX="231.4778" isBeginList="false"
      isList="false" section="exercice" categorie="consigne">
    <String id="p175_w4" CONTENT="Écris" HPOS="75.3680" VPOS="137.614" WIDTH="25.1500"
        HEIGHT="11.8750" STYLEREFS="font356" spNextWord="2.5750"/>
    <String id="p175_w5" CONTENT="chaque" HPOS="103.093" VPOS="137.614" WIDTH="39.5112"
        HEIGHT="11.8750" STYLEREFS="font356" spNextWord="2.5750"/>
    <String id="p175_w6" CONTENT="liste" HPOS="145.179" VPOS="137.614" WIDTH="22.5875"
        HEIGHT="11.8750" STYLEREFS="font356" spNextWord="2.5750"/>
    <String id="p175_w7" CONTENT="sans" HPOS="170.341" VPOS="137.614" WIDTH="23.2375"
        HEIGHT="11.8750" STYLEREFS="font356" spNextWord="2.5750"/>
    <String id="p175_w8" CONTENT="intrus." HPOS="196.154" VPOS="137.614"
        WIDTH="35.3238" HEIGHT="11.8750" STYLEREFS="font356" spNextWord="0"/>
  </Sentence>
  <Sentence id="p173_s9" CONTENT=" a. drôle ♦ amusant ♦ nul ♦ hilarant ♦ divertissant"
      majFont="font81" VMIN="152.614" HMIN="40.043" VMAX="179.514" HMAX="213.841"
      isBeginList="true" isList="true" section="exercice" categorie="contenuExercice">
    <String id="p175_w9" CONTENT="a." HPOS="40.0430" VPOS="152.614" WIDTH="9.3750"
        HEIGHT="11.8750" STYLEREFS="font286" spNextWord="2.6250"/>
    <String id="p175_w10" CONTENT="drôle" HPOS="52.0430" VPOS="152.614" WIDTH="26.8750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
    <String id="p175_w11" CONTENT="♦" HPOS="81.6305" VPOS="152.614" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.5875"/>
    <String id="p175_w12" CONTENT="amusant" HPOS="89.8930" VPOS="152.614"
        WIDTH="41.3250" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
    <String id="p175_w13" CONTENT="♦" HPOS="133.930" VPOS="152.614" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.5875"/>
    <String id="p175_w14" CONTENT="nul" HPOS="142.193" VPOS="152.614" WIDTH="15.3000"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
    <String id="p175_w15" CONTENT="♦" HPOS="160.205" VPOS="152.614" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
    <String id="p175_w16" CONTENT="hilarant" HPOS="168.593" VPOS="152.614"
        WIDTH="36.8612" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
    <String id="p175_w17" CONTENT="♦" HPOS="208.166" VPOS="152.614" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="0"/>
    <String id="p175_w18" CONTENT="divertissant" HPOS="40.0430" VPOS="167.614"
        WIDTH="57.0613" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="0"/>
  </Sentence>
  <Sentence id="p173_s19" CONTENT=" b. casse-pied ♦ ennuyeux ♦ fâcheux ♦ désobligeant
```

```
      ♦ marrant" majFont="font81" VMIN="184.026" HMIN="40.043"
      VMAX="210.92600000000002" HMAX="221.131" isBeginList="true" isList="true"
      section="exercice" categorie="contenuExercice">
25    <String id="p175_w19" CONTENT="b." HPOS="40.0430" VPOS="184.026" WIDTH="10.5875"
        HEIGHT="11.8750" STYLEREFS="font286" spNextWord="1.4125"/>
26    <String id="p175_w20" CONTENT="casse-pied" HPOS="52.0430" VPOS="184.026"
        WIDTH="52.7875" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
27    <String id="p175_w21" CONTENT="♦" HPOS="107.543" VPOS="184.026" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.5875"/>
28    <String id="p175_w22" CONTENT="ennuyeux" HPOS="115.805" VPOS="184.026"
        WIDTH="48.2000" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
29    <String id="p175_w23" CONTENT="♦" HPOS="166.718" VPOS="184.026" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.5875"/>
30    <String id="p175_w24" CONTENT="fâcheux" HPOS="174.980" VPOS="184.026"
        WIDTH="37.7637" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
31    <String id="p175_w25" CONTENT="♦" HPOS="215.456" VPOS="184.026" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="0"/>
32    <String id="p175_w26" CONTENT="désobligeant" HPOS="40.0430" VPOS="199.026"
        WIDTH="64.7113" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
33    <String id="p175_w27" CONTENT="♦" HPOS="107.466" VPOS="199.026" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.5875"/>
34    <String id="p175_w28" CONTENT="marrant" HPOS="115.729" VPOS="199.026"
        WIDTH="38.7000" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="0"/>
35  </Sentence>
36  <Sentence id="p173_s29" CONTENT=" c. agréable ♦ aimable ♦ attentionné ♦ sympa ♦
      gracieux" majFont="font81" VMIN="215.439" HMIN="40.043" VMAX="242.339"
      HMAX="266.528" isBeginList="true" isList="true" section="exercice"
      categorie="contenuExercice">
37    <String id="p175_w29" CONTENT="c." HPOS="40.0430" VPOS="215.439" WIDTH="8.8625"
        HEIGHT="11.8750" STYLEREFS="font286" spNextWord="3.1375"/>
38    <String id="p175_w30" CONTENT="agréable" HPOS="52.0430" VPOS="215.439"
        WIDTH="43.4500" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
39    <String id="p175_w31" CONTENT="♦" HPOS="98.2055" VPOS="215.439" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
40    <String id="p175_w32" CONTENT="aimable" HPOS="106.593" VPOS="215.439"
        WIDTH="39.3000" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
41    <String id="p175_w33" CONTENT="♦" HPOS="148.605" VPOS="215.439" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
42    <String id="p175_w34" CONTENT="attentionné" HPOS="156.993" VPOS="215.439"
        WIDTH="57.5987" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
43    <String id="p175_w35" CONTENT="♦" HPOS="217.304" VPOS="215.439" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
44    <String id="p175_w36" CONTENT="sympa" HPOS="225.691" VPOS="215.439" WIDTH="32.4488"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="2.7125"/>
45    <String id="p175_w37" CONTENT="♦" HPOS="260.853" VPOS="215.439" WIDTH="5.6750"
        HEIGHT="11.9000" STYLEREFS="font81" spNextWord="0"/>
46    <String id="p175_w38" CONTENT="gracieux" HPOS="40.0430" VPOS="230.439"
        WIDTH="41.1363" HEIGHT="11.9000" STYLEREFS="font81" spNextWord="0"/>
47  </Sentence>
48 </section>
```

manuel	id	fullex_rawtext	consigne_rawtext	conseil_rawtext	exemple_rawtext	enonce_rawtext	consigne_liste	enonce_liste	consigne_nbsents	enonce_nbsents	consigne_nbmots	enonce_nbmots	fullex_nbfonts	consigne_nbfonts	enonce_nbfonts	image	label
magnardCE2	173_3	écris chaque liste sans intrus. a. drôle ♦ amusant ♦ nul ♦ hilarant ♦ divertissant b. casse-pied ♦ ennuyeux ♦ fâcheux ♦ désobligeant ♦ marrant c. agréable ♦ aimable ♦ attentionné ♦ sympa ♦ gracieux	écris chaque liste sans intrus.			a. drôle ♦ amusant ♦ nul ♦ hilarant ♦ divertissant b. casse-pied ♦ ennuyeux ♦ fâcheux ♦ désobligeant ♦ marrant c. agréable ♦ aimable ♦ attentionné ♦ sympa ♦ gracieux	False	True	1	3	5	10	4	1	2		CacheIntrus

FIGURE A.1 – Extrait du corpus : exercice extrait au format tabulaire

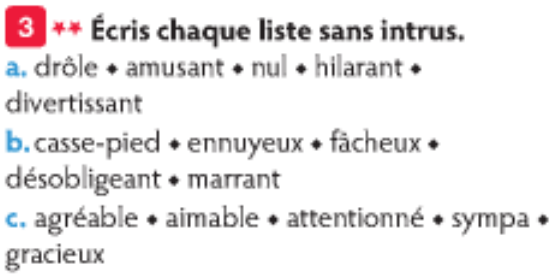


FIGURE A.2 – Extrait du corpus : exercice au format PNG pour LayoutLM

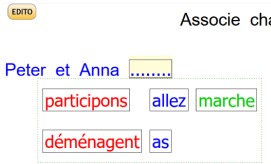
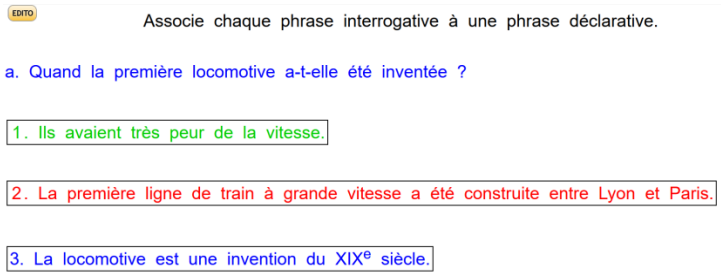
Listing A.2 – Extrait du corpus : exercice au format JSON pour LayoutLM


```
{
  "exercise": {
    "id": "173_3",
    "manual": "magnardCE2",
    "box": [8.0, 8.0, 235.0, 115.0],
    "label": "CacheIntrus",
    "sentences": [
      {
        "id": "p173_s3",
        "text": "3",
        "categorie": "numExo",
        "box": [13.448900000000002, 8.725999999999999, 19.941100000000006, 19.6087],
        "words": [
          {
            "text": "3",
            "box": [13.448900000000002, 8.725999999999999, 19.941100000000006, 19.6087]
          }
        ]
      },
      {
        "id": "p173_s4",
        "text": " Écris chaque liste sans intrus.",
        "categorie": "consigne",
        "box": [43.367999999999995, 9.614000000000004, 199.4778, 21.489000000000004],
        "words": [
          {
            "text": "Écris",
            "box": [43.367999999999995, 9.614000000000004, 68.518, 21.489000000000004]
          },
          {
            "text": "chaque",
            "box": [71.093, 9.614000000000004, 110.60419999999999, 21.489000000000004]
          },
          {
            "text": "liste",
            "box": [113.179, 9.614000000000004, 135.7665, 21.489000000000004]
          },
          {
            "text": "sans",
            "box": [138.341, 9.614000000000004, 161.57850000000002, 21.489000000000004]
          },
          {
            "text": "intrus.",
            "box": [164.154, 9.614000000000004, 199.4778, 21.489000000000004]
          }
        ]
      },
      {
        "id": "p173_s9",
        "text": " a. drôle ◆ amusant ◆ nul ◆ hilarant ◆ divertissant",
        "categorie": "contenuExercice",
        "box": [8.043, 24.614000000000004, 181.841, 51.51400000000001],
        "words": [
          {
            "text": "a.",
            "box": [8.043, 24.614000000000004, 17.418, 36.489000000000004]
          },
          {
            "text": "drôle",
            "box": [20.043, 24.614000000000004, 46.918000000000006, 36.51400000000001]
          },
          {
            "text": "◆",
            "box": [49.6305, 24.614000000000004, 55.305499999999995, 36.51400000000001]
          },
          {
            "text": "amusant",
            "box": [57.893, 24.614000000000004, 99.21800000000002, 36.51400000000001]
          },
```

```
          {
            "text": "◆",
            "box": [101.93, 24.614000000000004, 107.60500000000002, 36.51400000000001]
          },
          {
            "text": "nul",
            "box": [110.19300000000001, 24.614000000000004, 125.49300000000002,
                36.51400000000001]
          },
          {
            "text": "◆",
            "box": [128.205, 24.614000000000004, 133.88000000000002, 36.51400000000001]
          },
          {
            "text": "hilarant",
            "box": [136.593, 24.614000000000004, 173.4542, 36.51400000000001]
          },
          {
            "text": "◆",
            "box": [176.166, 24.614000000000004, 181.841, 36.51400000000001]
          },
          {
            "text": "divertissant",
            "box": [8.043, 39.614000000000004, 65.1043, 51.51400000000001]
          }
        ]
      },
      {
        "id": "p173_s17",
        "text": " b. casse-pied ◆ ennuyeux ◆ fâcheux ◆ désobligeant ◆ marrant",
        "categorie": "contenuExercice",
        "box": [8.043, 56.02600000000001, 189.131, 82.92600000000002],
        "words": [
          {
            "text": "b.",
            "box": [8.043, 56.02600000000001, 18.630499999999998, 67.90100000000001]
          },
          {
            "text": "casse-pied",
            "box": [20.043, 56.02600000000001, 72.8305, 67.92600000000002]
          },
          {
            "text": "◆",
            "box": [75.543, 56.02600000000001, 81.218, 67.92600000000002]
          },
          {
            "text": "ennuyeux",
            "box": [83.805, 56.02600000000001, 132.005, 67.92600000000002]
          },
          {
            "text": "◆",
            "box": [134.718, 56.02600000000001, 140.393, 67.92600000000002]
          },
          {
            "text": "fâcheux",
            "box": [142.98, 56.02600000000001, 180.7437, 67.92600000000002]
          },
          {
            "text": "◆",
            "box": [183.456, 56.02600000000001, 189.131, 67.92600000000002]
          },
          {
            "text": "désobligeant",
            "box": [8.043, 71.02600000000001, 72.7543, 82.92600000000002]
          },
          {
            "text": "◆",
            "box": [75.466, 71.02600000000001, 81.14099999999999, 82.92600000000002]
          },
          {
```

```
            "text": "marrant",
            "box": [83.729, 71.02600000000001, 122.429, 82.92600000000002]
          }
        ]
      },
      {
        "id": "p173_s26",
        "text": " c. agréable ♦ aimable ♦ attentionné ♦ sympa ♦ gracieux",
        "categorie": "contenuExercice",
        "box": [8.043, 87.439, 234.52800000000002, 114.339],
        "words": [
          {
            "text": "c.",
            "box": [8.043, 87.439, 16.905500000000004, 99.314]
          },
          {
            "text": "agréable",
            "box": [20.043, 87.439, 63.492999999999995, 99.339]
          },
          {
            "text": "♦",
            "box": [66.2055, 87.439, 71.8805, 99.339]
          },
          {
            "text": "aimable",
            "box": [74.593, 87.439, 113.893, 99.339]
          },
          {
            "text": "♦",
            "box": [116.60499999999999, 87.439, 122.28, 99.339]
          },
          {
            "text": "attentionné",
            "box": [124.993, 87.439, 182.5917, 99.339]
          },
          {
            "text": "♦",
            "box": [185.304, 87.439, 190.979, 99.339]
          },
          {
            "text": "sympa",
            "box": [193.691, 87.439, 226.13979999999998, 99.339]
          },
          {
            "text": "♦",
            "box": [228.853, 87.439, 234.52800000000002, 99.339]
          },
          {
            "text": "gracieux",
            "box": [8.043, 102.439, 49.1793, 114.339]
          }
        ]
      }
    ]
  }
}
```


A.2 Jeu d'étiquettes

Étiquette	Exercice original	Exercice adapté
Associe	**4** * **Associe chaque sujet au verbe qui convient.** Peter et Anna ○ ○ participons Je ○ ○ allez Nous ○ ○ marche Tu ○ ○ déménagent Vous ○ ○ as	
AssocieCoche	**3** ** **Associe chaque phrase interrogative à une phrase déclarative.** **a.** Quand la première locomotive a-t-elle été inventée ? **b.** Comment réagissaient les premiers voyageurs en train ? **c.** Quelles villes a reliées la première ligne de TGV ? **1.** Ils avaient très peur de la vitesse. **2.** La première ligne de train à grande vitesse a été construite entre Lyon et Paris. **3.** La locomotive est une invention du XIX^{e} siècle.	
Atypique	Pas d'adaptation type.	
CacheIntrus	**14** * **Recopie chaque liste sans l'intrus.** **a.** partout • grandes • aussi • demain **b.** tôt • longtemps • dormir • assez **c.** espadrille • pour • de • par **d.** car • donc • mais • suivant	

Suite page suivante

Étiquette	Exercice original	Exercice adapté
Classe	**3** ** **Relève dans ce texte les verbes conjugués et les verbes à l'infinitif. Classe-les dans le tableau.** verbes conjugués | verbes à l'infinitif La tortue leva ses yeux noirs et enfoncés sur le petit garçon. […] Personne ne bougeait. Puis, avec une grande dignité, l'énorme bête se retourna et se dirigea vers le bord de l'eau en se dandinant, sans se presser. Elle traversa posément la plage de sable et sa grosse carapace se balançait doucement. La foule regardait en silence. La tortue entra dans l'eau. Elle continua d'avancer. Roald Dahl, *L'enfant qui parlait aux animaux*, trad. Marie-Raymond Farré, © Roald Dahl Ltd, © Éditions Gallimard pour la traduction, 2001.	EDITO Colorie dans ce texte les verbes conjugués en jaune et les verbes à l'infinitif en rose. La tortue leva ses yeux noirs et enfoncés sur le petit garçon. […] Personne ne bougeait. Puis, avec une grande dignité, l'énorme bête se retourna et se dirigea vers le bord de l'eau en se dandinant, sans se presser.
ClasseCM	**2** ** **Écris pour chaque mot la saison correspondante : *hiver, automne, printemps, été*.** **a.** congère • verglas • estival • printanier • bruine **b.** givre • bourgeonnement • hivernal • canicule • orage	EDITO Écris pour chaque mot la saison correspondante : hiver automne printemps ou été a. congère → ..... hiver automne printemps été
ClassePhrase	MANIPULER **3** **Quelle phrase peux-tu faire commencer par « Aujourd'hui » ? par « Demain » ? par « Hier » ?** • J'ai fait un cauchemar. • Le chat de David miaule. • Fiona repeindra sa chambre.	EDITO Quelle phrase peux-tu faire commencer par Aujourd'hui ? par Demain ? par Hier ? Colorie les phrases de la bonne couleur J'ai fait un cauchemar. Le chat de David miaule. Fiona repeindra sa chambre.

Suite page suivante

Étiquette	Exercice original	Exercice adapté
CliqueEcrire	**5** * **Écris l'infinitif des verbes soulignés dans ce texte.** Dans les rues, les voitures vont et viennent dans tous les sens. Théo regarde à gauche, puis à droite. Il voit une grosse moto qui prend le virage et accélère. Théo fait un bond en arrière et ne peut presque plus bouger. « Tout va bien… dit sa maman. Tu veux rentrer à la maison ? – Oui ! » répond Théo.	Cette adaptation n'est actuellement pas réalisable sur la plate-forme du Cartable Fantastique.
CM	**8** ** **Complète les phrases avec *on* ou *ont*.** **a.** … est cachés derrière le rideau. **b.** Ils … sûrement fermé la porte à clé. **c.** Ils … raison. **d.** … va nous apporter nos plats. **e.** … est serrés l'un contre l'autre.	EDITO Complète les phrases avec on ou ont. a. .... est cachés derrière le rideau. On Ont
CMDouble	**11** * **Remplace chaque mot en gras par un antonyme de la liste.** *arriver* ◆ *enrouler* ◆ *intérieur* ◆ *long* **a.** Les ballons sont interdits à l'**extérieur** ! **b.** Dis-nous l'heure à laquelle tu penses **partir**. **c.** Le marchand de tissu choisit de **dérouler** ses tissus pour mieux les présenter. **d.** Il préfère porter un pantalon **court**.	EDITO Remplace chaque mot en jaune par un antonyme de la liste. arriver enrouler intérieur ou long a. Les ballons sont interdits à l'extérieur ! → a. Les ballons sont interdits à l'..... !
CocheGroupeMots	**2** * **Recopie uniquement les verbes conjugués à l'imparfait.** il cassait ◆ je cherche ◆ nous mangions ◆ tu imaginais ◆ nous finissions ◆ elle tremble ◆ vous passez ◆ elles sonnent ◆ nous déménageons ◆ elles refroidissaient ◆ vous ponciez ◆ nous rinçons	EDITO Recopie uniquement les verbes conjugués à l'imparfait. il cassait / je cherche / nous mangions / tu imaginais nous finissions / elle tremble / vous passez / elles sonnent nous déménageons / elles refroidissaient / vous ponciez / nous rinçons

Suite page suivante

Étiquette	Exercice original	Exercice adapté
CocheIntrus	**2** * **Recopie l'intrus de chaque liste.** **a.** père • terre • mère • lèvre • frère **b.** épuisette • idée • appétit • lièvre • opération **c.** téléphone • éléphant • légère • épuisé	EDITO Colorie l'intrus de chaque liste a. père terre mère lèvre frère
CocheLettre	**5** ** **Encadre le radical et souligne le suffixe des adjectifs suivants.** **a.** peureux • pensif • excessif • malheureux **b.** maniable • craintif • lisible • lavable	EDITO Colorie en jaune le radical et en rose le suffixe des adjectifs suivants. a. peureux pensif excessif malheureux b. maniable craintif lisible lavable
CocheMot	**3** * **Relève les verbes conjugués à l'imparfait.** Quand le ballon arrivait, l'équipe fonçait à sa poursuite. Parfois, ils coinçaient la balle entre leurs pieds. Elsa et Mathis se bousculaient fréquemment. Ils donnaient plus de coups sur les tibias de l'adversaire que sur la balle, qu'ils piétinaient à l'occasion.	EDITO Colorie les verbes conjugués à l'imparfait. Quand le ballon arrivait, l'équipe fonçait à sa poursuite. Parfois, ils coinçaient la balle entre leurs pieds. Elsa et Mathis se bousculaient fréquemment. Ils donnaient plus de coups sur les tibias de l'adversaire que sur la balle, qu'ils piétinaient à l'occasion.
CocheMot*	MANIPULER **5** **Associe le pluriel qui convient à chaque nom singulier.** un journal – un idéal – des chevaux – un festival – un animal – un carnaval – un cheval – des journaux – des idéaux – des festivals – des carnavals – des animaux	EDITO Colorie le pluriel qui convient à chaque nom singulier un journal un idéal des chevaux un festival un animal un carnaval un cheval des journaux des idéaux des festivals des carnavals des animaux

Suite page suivante

Étiquette	Exercice original	Exercice adapté
CochePhrase	**1** * **Recopie uniquement les phrases justes.** **a.** Un déterminant + un nom forment un groupe nominal. **b.** *La forêt bleue* n'est pas un groupe nominal. **c.** On peut supprimer les adjectifs dans le groupe nominal. **d.** Si le nom est au pluriel, le déterminant est au pluriel.	EDITO Colorie uniquement les phrases justes. a. Un déterminant + un nom forment un groupe nominal. b. La forêt bleue n'est pas un groupe nominal. c. On peut supprimer les adjectifs dans le groupe nominal. d. Si le nom est au pluriel, le déterminant est au pluriel.
CochePonctuation	**6** * **Entoure les signes de ponctuation.** **a.** L'aviateur est perdu. **b.** Malheureusement, son avion est en panne ! **c.** Il se retrouve seul au milieu du désert. **d.** Pendant la nuit, il a eu très froid. **e.** Comment va-t-il dessiner ce mouton ?	EDITO Colorie les signes de ponctuation a. L'aviateur est perdu . b. Malheureusement , son avion est en panne ! c. Il se retrouve seul au milieu du désert .
Echange	**3** * **Remets les groupes de mots dans l'ordre pour écrire des phrases.** **a.** au cirque. / nous allons / Demain après-midi, **b.** une glace / nous prendrons / Mon frère et moi, / à l'entracte. **c.** saluent le public / Les musiciens sous les applaudissements.	EDITO Remets les groupes de mots dans l'ordre pour écrire des phrases. a. au cirque. / nous allons / Demain après-midi,
Suite page suivante		

Étiquette	Exercice original	Exercice adapté
EchangeLettre	**À toi de jouer** **Anagrammes. Remets les lettres dans l'ordre pour retrouver huit déterminants.** NU – TORVE – AS – SON – SEC – RELU – NOM – EST	EDITO Anagrammes. Remets les lettres dans l'ordre pour retrouver huit déterminants. UN – TORVE – AS – SON – SEC – RELU – NOM – EST
EditPhrase	**4** ** **Recopie chaque phrase en rétablissant la ponctuation comme dans l'exemple.** *la nuit dans le ciel les étoiles brillent* → ***La nuit, dans le ciel, les étoiles brillent.*** **a.** souvent dans le noir j'ai peur **b.** à tour de rôle avant de jouer chaque joueur pioche une carte **c.** au loin dans la lumière du soleil se découpent les montagnes	EDITO Recopie chaque phrase en rétablissant la ponctuation comme dans l'exemple. la nuit dans le ciel les étoiles brillent → La nuit, dans le ciel, les étoiles brillent. a. souvent dans le noir j'ai peur → a. souvent dans le noir j'ai peur
EditTexte	**9** * **Recopie ce texte en ajoutant les accents aigu ou grave qui manquent sur les lettres e. Aide-toi de ton dictionnaire.** Un beau jour, les viperes donnerent un grand bal. [...] Les crapauds, entierement recouverts d'ecailles de poissons, avançaient en se dandinant comme s'ils nageaient. Les grenouilles s'etaient parfume tout le corps et marchaient sur les pattes de derriere. [...] Seuls les flamants etaient tristes parce que, comme ils ne sont pas tres intelligents, ils n'avaient su imaginer aucune toilette. Horacio Quiroga, *Contes de la forêt vierge*, © Éditions du Seuil, 1998.	EDITO Complète ce texte en ajoutant les accents aigu ou grave qui manquent sur les lettres e. Aide-toi de ton dictionnaire. Un beau jour, les viperes donnerent un grand bal. [...] Les crapauds, entierement recouverts d'ecailles de poissons, avançaient en se dandinant comme s'ils nageaient. Les grenouilles s'etaient parfume tout le corps et marchaient sur les pattes de derriere. [...] Seuls les flamants etaient tristes parce que, comme ils ne sont pas tres intelligents, ils n'avaient su OUPS
ExpressionEcrite	**16** ** **Écris trois phrases avec les négations suivantes.** *ne … plus* ◆ *ne … jamais* ◆ *ni … ni*	EDITO Écris trois phrases avec les négations suivantes. ne … plus ne … jamais ni … ni

Suite page suivante

Étiquette	Exercice original	Exercice adapté
GenreNombre	**3** * **Pour chaque nom, écris son genre et son nombre.** *flûtes → féminin, pluriel* **a.** classe • cachots • poux • mer • poules **b.** montagnes • clientes • calendrier • nièces • invitée • activité • métiers • amitiés **c.** œil • eaux • lièvre • bois • rond	EDITO Pour chaque nom, écris son genre et son nombre. flute → féminin, pluriel a. classe → ....., ... cachots → ....., ...
NonAdaptable	Exercices inadaptés pour les élèves dyspraxiques : pas d'adaptation.	
Phrases	**7** ** **Écris deux phrases en utilisant le mot *chaîne* dans deux sens différents.**	EDITO Écris deux phrases en utilisant le mot chaîne dans deux sens différents. 1. 2.
Question	**8** * **Lis cet article de dictionnaire, puis réponds aux questions.** **hérisser v.** ❶ Dresser son poil ou ses plumes. *Le chat hérisse ses poils en voyant le chien.* ❷ Mettre quelqu'un en colère. *Son attitude me hérisse.* **SYN. exaspérer, horripiler.** *Dictionnaire Larousse Junior 7-11 ans*, © Larousse, 2017. **a.** Combien de sens le mot *hérisser* a-t-il ? **b.** À quel sens correspond la phrase suivante : « Ce mauvais joueur a le don de me hérisser ! » ? **c.** Donne un synonyme du verbe *hérisser*.	EDITO Lis cet article de dictionnaire, puis réponds aux questions. **hérisser v.** 1. Dresser son poil ou ses plumes. Le chat hérisse ses poils en voyant le chien. 2. Mettre quelqu'un en colère. Son attitude me hérisse. SYN. exaspérer, horripiler. EDITO a. Combien de sens le mot hérisser a-t-il ?
Suite page suivante		

Étiquette	Exercice original	Exercice adapté
RC	**10** ** **Complète chaque phrase avec un adjectif de ton choix.** **a.** Au printemps, l'air est généralement … . **b.** La nature émerge d'un … sommeil hivernal. **c.** Arrête de bouger, reste … une minute. **d.** Un vent … souffle sur la côte.	EDITO Complète chaque phrase avec un adjectif de ton choix. a. Au printemps, l'air est généralement ☐ .
RCCadre	**8** * **Recopie les phrases avec le verbe conjugué au présent.** **a.** Il *(venir)* me voir. **b.** Nous *(venir)* tous les jours. **c.** Je *(venir)* à ta rencontre. **d.** Vous ne *(venir)* jamais. **e.** *(revenir)*-elle ce soir ?	EDITO Complète les phrases avec le verbe conjugué au présent. venir a. Il ☐ me voir.
RCDouble	**12** ** **Transforme ces phrases en remplaçant les mots en gras par leur antonyme.** **a.** Les légumes **crus** sont très bons pour la santé. **b.** Notre magasin est **ouvert** tous les dimanches. **c.** Toutes les réponses de son exercice sont **justes**. **d.** Mon ordinateur est un modèle **récent**.	EDITO Transforme ces phrases en remplaçant les mots en jaune par leur antonyme. a. Les légumes crus sont très bons pour la santé. → a. Les légumes ☐ sont très bons pour la santé.
RCImage	**7** ** **Écris les mots représentés par les dessins si le son [s] s'écrit avec la lettre *s*.** *Tu peux vérifier à l'aide du dictionnaire.* 	EDITO Écris les mots représentés par les dessins si le son [s] s'écrit avec la lettre s.  ☐  ☐
Texte	**9** À l'oral **Prononcer cette phrase distinctement.** Six-cent-six scies scieront six-cent-six saucissons.	EDITO À l'oral : Prononcer cette phrase distinctement. Six-cent-six scies scieront six-cent-six saucissons.

Suite page suivante

Étiquette	Exercice original	Exercice adapté
Trait	**2** * **Sépare par un trait le radical et le suffixe de chacun de ces mots.** **a.** payable ◆ paysagiste ◆ fleurette ◆ maladif **b.** fillette ◆ poussoir ◆ affichette ◆ géographie **c.** lisible ◆ habileté ◆ énormément ◆ dérapage ◆ éducation	EDITO Sépare par un trait le radical et le suffixe de chacun de ces mots. a. payable ◆ paysagiste ◆ fleurette ◆ maladif
TransformeMot	**7** * **Écris l'infinitif des verbes.** **a.** je choisis ◆ il prend ◆ vous salissez **b.** elle pétille ◆ elle voit ◆ il galope ◆ je reçois **c.** nous partons ◆ je dis ◆ tu regardes ◆ elle fait	EDITO Écris l'infinitif des verbes. a. je choisis → ☐ il prend → ☐ vous salissez → ☐
TransformePhrase	**7** ** **Transforme chaque couple de phrases pour en faire une seule.** *Léa souffle les bougies. Léo souffle les bougies.* *→ Léa et Léo soufflent les bougies.* **a.** Le manchot glisse. Le pingouin glisse. **b.** Kiki s'approche de moi. Loulou s'approche de moi. **c.** La grive chante dans les arbres. Le rossignol chante dans les arbres. **d.** Le cheval a des sabots. L'âne a des sabots. **e.** Le lézard a des écailles. Le crocodile a des écailles.	EDITO Transforme chaque couple de phrases pour en faire une seule. Léa souffle les bougies. Léo souffle les bougies. → Léa et Léo soufflent les bougies. a. Le manchot glisse. Le pingouin glisse. → ☐
Suite page suivante		

Étiquette	Exercice original	Exercice adapté
VraiFaux	**1** * **Réponds par *vrai* ou *faux*.** **a.** À la 2e personne du singulier, les verbes *aller* et *venir* prennent un *s*. **b.** Le verbe *aller* se conjugue comme *chanter*. **c.** *Je vais* et *nous allons* sont deux formes du même verbe. **d.** Le verbe *venir* se conjugue comme *finir*.	EDITO Réponds par vrai ou faux. a. À la 2e personne du singulier, les verbes aller et venir prennent un s. → .... vrai faux

TABLE A.1 – Jeu d'étiquettes et exemples d'exercices

A.3 Partitionnement du corpus

Grande classe	Classe	Apprentissage	Validation	Test
Select		407	57	117
	CocheMot	142	20	40
	Classe	83	11	23
	CocheGroupeMots	73	10	21
	CochePhrase	71	10	20
	ClasseCM	27	4	7
	CocheMot*	12	2	4
	CocheLettre	7	1	2
	Trait	6	1	2
	AssocieCoche	5	1	2
	CochePonctuation	2	0	1
	ClassePhrase	6	1	2
Choose		389	57	109
	CM	264	38	75
	Associe	75	11	21
	VraiFaux	12	2	3
	GenreNombre	8	1	2
	CMDouble	3	1	1
Fill		715	102	203
	RC	233	33	66
	ExpressionEcrite	103	15	30
	TransformePhrase	100	14	29
	TransformeMot	89	13	25
	RCImage	73	10	21
	RCCadre	45	6	13
	RCDouble	30	4	8
	Phrases	24	4	6
	CliqueEcrire	11	2	3
	Question	7	1	2
Edit		130	18	37
	EditPhrase	107	15	31
	EditTexte	23	3	6
Swap		26	4	7
	Echange	26	4	7
Show		69	10	20
	Texte	69	10	20
Intrus		63	9	17
	CacheIntrus	37	5	10
	CocheIntrus	26	4	7

TABLE A.2 – Répartition des classes et grandes classes dans les jeux de données

A.4 Augmentation des données

Classe	Originaux	Artificiels		
		Cross-over	**Rétro-traduction**	**Substitution lexicale**
VraiFaux	**Réponds par vrai ou faux** a. Manger, pondre, cinq sont bien orthographiés. b. Le n devient m devant m, p, b et d. c. Embonpoint s'écrit avec un m et deux n. d. Bonbon et néanmoins sont bien orthographiés. e. Cambrioleur, caméléon et camembert sont bien orthographiés.	*Pas de cross-over consigne-énoncé car toutes les consignes sont identiques.*	**Répondez pour vrai ou faux.** A. Manger, poser, cinq sont bien orthographiés. - b. Le n devient m devant m, p, b et d. C. Embonpoint est écrit avec un m et deux n. D. Candy et néanmoins sont bien orthographiés. E. Burglar, caméléon et camembert sont bien orthographiés.	**Répliquer par convenable ou inexact.** a. Croquer, rédiger, cinq sont bien écrit. b. Le n rendre m devant m, p, argent et d. c. Embonpoint se graver avec un m et deux d. Bonbon et néanmoins sont bien écrit. e. Cambrioleur, lézard et tampon sont bien écrit.
GenreNombre	**Recopie chaque groupe nominal en indiquant son genre (masculin, féminin) et son nombre (singulier, pluriel).** nos nouveaux amis – cette incroyable histoire – des livres épais – un beau dessin – leurs affaires personnelles – une fête réussie – ce gentil héros – vos jolies barques	**Recopie les sept noms communs du texte et indique pour chacun s'il est masculin ou féminin et singulier ou pluriel.** nos nouveaux amis – cette incroyable histoire – des livres épais – un beau dessin – leurs affaires personnelles – une fête réussie – ce gentil héros – vos jolies barques	**Copier chaque groupe nominal en indiquant son sexe (hommes, femmes) et son nombre (singulaire, pluriel).** Nos nouveaux amis – cette histoire incroyable – des livres épais – un beau dessin – leurs affaires personnelles – une fête réussie – ce beau héros – vos jolis bateaux	**Extraire chaque ensemble nominal en marquer sa classe (viril, femelle) et son unité (individuel, multiple).** notre vierge compagnon – cet ridicule mensonge – du volume profond – un exquis treillis – leur ennui individuel – un festival réussi – ce chic grand – votre infâme barquette
Suite page suivante				

Classe	Originaux	Artificiels		
		Cross-over	Rétro-traduction	Substitution lexicale
CochePonctuation	**Repère et nomme le signe de ponctuation à la fin de chaque phrase.** a. Avez-vous pris votre parapluie ? b. Il y a 3 stylos dans ma trousse. c. Je déteste le chou-fleur ! d. Quelle est ta couleur préférée ?	**Entoure les signes du dialogue et la ponctuation.** a. Avez-vous pris votre parapluie ? b. Il y a 3 stylos dans ma trousse. c. Je déteste le chou-fleur ! d. Quelle est ta couleur préférée ?	**- Marques et noms du signe de ponctuation à la fin de chaque phrase.** Tu as pris ton parapluie ? Il y a trois stylos dans mon kit. Je déteste le chou-fleur ! Quelle est ta couleur préférée ?	**Indice et nomme le caractéristique de tiret à la solution de chaque parole.** a. Avez-vous interprété votre protection ? b. Il y a 3 porte-plume dans mon étui. c. Je déteste le champignon ! d. Quelle est ta ton préférée ?
AssocieCoche	**Associe les sujets et les verbes.** Ma cousine marchons chaque jour. Samia et Zoé adore visiter Paris. Paul et moi préparez des crêpes. Noé et toi viennent d'arriver.	**Associe la question à la réponse qui correspond.** Ma cousine marchons chaque jour. Samia et Zoé adore visiter Paris. Paul et moi préparez des crêpes. Noé et toi viennent d'arriver.	**- Associe sujets et verbes.** - Mon cousin. Nous marchons tous les jours. Samia et Zoé Il aime visiter Paris. Paul et moi - Préparez des crêpes. Noah et toi Je viens d'arriver.	**Adapter la figurine et la parole.** Mon parent Cingler chaque traitement. Samia et Zoé Adore saluer Paris. Paul et moi Composer de la galette. Noé et toi Réussir de parvenir.
Suite page suivante				

Classe	Originaux	Artificiels		
		Cross-over	Rétro-traduction	Substitution lexicale
CliqueEcrire	**Recopie ces phrases en conjuguant les verbes entre parenthèses à l'imparfait.** L'année dernière, Louane (chanter) dans une chorale. Félix, lui, (jouer) du trombone dans un orchestre. Tous les enfants (répéter) les vendredis après l'école. Nous (écouter) leurs concerts. Après chaque spectacle, je (bavarder) avec les autres invités, pendant que vous (ranger) la salle.	**Écris l'infinitif des verbes soulignés dans ce texte.** L'année dernière, Louane (chanter) dans une chorale. Félix, lui, (jouer) du trombone dans un orchestre. Tous les enfants (répéter) les vendredis après l'école. Nous (écouter) leurs concerts. Après chaque spectacle, je (bavarder) avec les autres invités, pendant que vous (ranger) la salle.	**Copier ces phrases en conjuguant les verbes entre parenthèses à l'imperfection.** L'année dernière, Louane (chantant) dans une chorale. Félix (jouant) le trombone dans un orchestre. Tous les enfants (répétés) le vendredi après l'école. Nous (écouter) leurs concerts. Après chaque spectacle, je (bavarder) avec les autres invités, tandis que vous (ranger) la chambre.	**Extraire cette formule en grouper la parole entre crochet à l'imparfait.** La berge ultime, Louane (répéter) dans un hymne. Félix, lui, (rire) de l'attache dans un groupe. Tous le bébé (bourdonner) les vendredis après l'établissement. Nous (suivre) leurs musique. Après chaque attraction, je (jacasser) avec les restant appelé, pendant que vous (distribuer) le théâtre.
Phrases	**Écris une phrase avec chacun des mots suivants.** se coiffer un point un bijou	**Forme trois nouveaux mots contenant le son [ɔ̃] à partir des mots proposés.** se coiffer un point un bijou	**Écrire une phrase avec chacun des mots suivants.** - Coiffure Un seul point Un bijou	**Noter une parole avec chacun de l'expression postérieure.** Se couvrir Une ponctuation Une perfection

TABLE A.3 – Augmentation de données : exemples d'exercices générés avec les méthodes de cross-over, rétro-traduction et substitution lexicale