

Faculté de philosophie, arts et lettres

Annotation automatique d'erreurs dans des textes d'apprenants du Français (FLE)

**Développement d'une ressource annotée en
erreurs (approches linguistiques et techniques)**

Auteur : Angela Kasparian

Promoteur(s) : Thomas François et Rodrigo Wilkens

Année académique 2022-2023

Master en linguistique, finalité traitement automatique du langage

Remerciements :

Avant d'entrer dans le vif du sujet, il est important pour moi de remercier chaleureusement toutes les personnes (de la plus proche à la plus lointaine) qui m'ont permis de présenter ces quelques pages représentant plus d'une année de travail acharné. Tout d'abord, mon promoteur, Thomas François, pour avoir accepté de m'inclure sur le projet dans lequel mon mémoire s'inscrit. Je tiens aussi à le remercier pour la confiance et le soutien précieux qu'il m'a accordé aussi bien pendant mon stage, pendant la rédaction du mémoire que pendant les mois que j'ai passé en tant que jobiste au CENTAL. Parallèlement au soutien de mon promoteur, j'ai eu l'énorme chance de recevoir l'aide de mon Co promoteur, monsieur Rodrigo Wilkens que je remercie infiniment. Les pages que vous vous apprêtez à parcourir seraient bien moins abouties et qualitatives sans ses conseils et coups de main toujours très avisés et bienveillants. Outre le corps professoral et professionnel qui a encadré la création de ce mémoire, je me dois de remercier, évidemment mes parents pour m'avoir soutenue et poussée toutes ces années. Au même titre j'ai aussi eu la chance de compter sur mes frères et sœurs, Laura Melaet et Gérôme Melaet qui ont su, par leurs expériences, m'écouter, me comprendre et m'aider à surmonter les doutes qui surviennent tout au long d'un parcours universitaire. Enfin je remercie mon compagnon et mes amis de m'avoir supportée malgré le stress et la pression et d'avoir su me motiver quand il le fallait.

Notations :

Dans ce mémoire, les termes spécifiques au champs lexicaux du traitement automatique du langage et de la linguistique sont explicités par le biais d'un glossaire. Ce dernier constitue la dernière partie du mémoire (partie 3) et se présente comme une liste de termes complexes accompagnés d'une courte définition ou description. Les définitions sont souvent tirées de mes connaissances acquises pendant mon master (grâce aux différents cours de linguistique et de TAL). Les termes présents dans le glossaire sont indiqués par un indice numérique directement après la première apparition du terme dans le texte même. Cet indice numérique fait référence à l'indice du terme dans la liste des mots du glossaire.

Il prend systématiquement le même format : *terme complexe* [indice].

Les mots contenus dans le glossaire sont ceux qui nécessitent des explications plus longues que quelques mots, les autres mots spécifiques, moins obscurs, sont directement définis dans le corps du texte (soit entre parenthèses, soit par le biais d'une anaphore entre virgules).

Table des matières :

<u>Partie I :</u> Projet « Annotation automatique de textes d'apprenants FLE ».	p.6
Chapitre 1 : Introduction	p. 8
Chapitre 2 : Cadre théorique	p.13
2.1 L'orthographe, une norme ?	p.15
2.1.1 La place de l'orthographe française : définitions sociales et historiques. ...	p.15
2.1.2 L'erreur, rectifications de la norme et impact.	p.20
2.1.3 La norme dans le contexte éducatif, enjeux du FLE et analyse des erreurs.	p.25
2.1.4 Annotation manuelle, typologies des erreurs d'écriture.	p.27
2.2 Approches pédagogiques, le cas du traitement automatique du langage.	p.30
2.2.1 Annoter l'erreur, une ressource pédagogique nécessaire.	p.31
2.2.2 ELAO / ALAO, les apports et le besoin d'une approche automatique.	p.32
2.3 L'annotation automatique, approches techniques.	p.35
2.3.1 L'annotation d'erreur, implications et ressources.	p.35
2.3.2 Approche chronologique (et non exhaustive) des outils d'annotation automatique.	p.37
2.4 Spécificité liée à la langue et au public cibles.	p.44
2.4.1 Le problème des données.	p.44
2.4.2 Le problème des outils.	p.46
Chapitre 3 : Méthodologie	p.49
3.1 Typologie des erreurs.	p.51
3.2 Campagne d'annotation.	p.60
3.3 Choix des modèles.	p.69
3.4 Corpus, entraînement et évaluation des modèles.	p.77
Chapitre 4 : Résultats	p.82
4.1 Description des données.	p.83
4.2 Résultats de la campagne d'annotation.	p.86
4.2.1 Analyse du corpus.	p.86
4.2.2 Évaluation de l'annotation.	p.95

4.3 Résultats de la campagne de modélisation.	p.101
4.3.1 Résultats dans la littérature, « baseline » préliminaire.	p.101
4.3.2 Résultats pour le français.	p.103
4.3.3 Discussion.	p.108
4.4 Limitations et place pour la suite.	p.110
4.4.1 Première piste : la typologie.	p.110
4.4.2 Deuxième piste: la taille du corpus.	p.111
4.4.3 Troisième piste : les modèles.	p.112
Chapitre 5 : Conclusion	p.114
<u>Partie II</u> : Bibliographie et annexes	p.119
Bibliographie	p.120
Annexes	p.131
Annexe 1 (nouvelle orthographe)	p.131
Annexe 2 (Typologie d’erreurs de Nina Catach)	p.132
Annexe 3 (Typologie d’erreurs de Câslaru)	p.133
Annexe 4 (Tagset pour l’annotation de Granger)	p.134
Annexe 5 (Catégories d’erreurs, ERRANT)	p.135
Annexe 6 (Notre typologie, 2 versions)	p.136
Annexe 7 (Guide d’annotation final)	p.140
Annexe 8 à 11 (Code traitement des données INCEpTION)	p.153
Annexe 12 (Code accord inter-annotateurs global)	p.158
Annexe 13 (Code accord inter-annotateurs deux à deux)	p.159
Annexe 14 (Code baseline SVM)	p.160
Annexe 15 (Code modèle BERT)	p.161
Annexe 16 (Code analyse résultats BERT)	p.164
Annexe 17 (Résultats BERT multi-classes 9 epoch)	p.166
Annexe 18 (Répartition des niveaux CECR)	p.167
<u>Parties III</u> : Glossaire	p.169

Partie I. Projet :
« Annotation automatique de
textes d'apprenants FLE ».

Chapitre 1 : Introduction.

1. **Introduction :**

La langue est une œuvre vivante et mouvante, un flux continu de pensées et d'expressions. Elle évolue, se transforme et s'adapte aux besoins de ceux qui la parlent et l'écrivent. Cependant, au cœur de cette richesse linguistique, une problématique persiste : l'orthographe. Le français, en particulier, est réputé pour sa complexité orthographique, source de frustration pour de nombreux apprenants ainsi que pour les locuteurs natifs.

À mi-chemin entre les réflexions linguistiques ou sociologiques qui entourent le concept de l'orthographe (surtout dans un cadre pédagogique de transmission) et l'évolution inévitable du monde technologique et des intelligences artificielles, ce mémoire aura pour but ultime de concilier ces deux mondes en palliant au maximum leurs familles respectives.

En premier lieu, il convient de placer le cadre de ce mémoire. En effet, la première problématique est le manque flagrant de ressources pédagogiquement et scientifiquement exploitables, adaptées au français et aux apprenants de ladite langue. Que ce soit pour de la recherche en linguistique (travaux d'analyse d'erreurs, évaluation ...) ou encore pour la création de pédagogies adaptées aux erreurs des apprenants en français (FLE). Il est nécessaire de fonder ces objectifs sur une base de données quantitativement exploitable et qualitativement représentative. Le constat du manque de ressources pour le français est d'autant plus affligeant qu'il ne manque pas seulement des données pour la niche que constituent les apprenants français de langue étrangère, mais il manque également des ressources annotées en erreurs des locuteurs natifs du français. Or, il est clair que l'obtention de telles ressources est plus qu'envisageable, il suffit de regarder les corpus disponibles en anglais (les corpus de natifs ou bien « learner corpora »).

Une fois cet objectif cerné, une seconde problématique s'est imposée à nous, la problématique de l'importance sociale et historique de l'orthographe en français, et par-dessus tout, sa complexité. Il existe, en effet, de nombreux débats autour de son utilité, de ses règles et de sa norme, l'orthographe du français semble bien complexe à codifier et à identifier. Et pourtant, pour créer une ressource annotée en erreurs (d'orthographe) il est nécessaire de cerner la ou les normes desquelles découlent ces erreurs. Surtout si l'objectif, comme ici, est de rendre le corpus exploitable pédagogiquement ou scientifiquement.

Enfin, par la force des choses, l'option de l'intelligence artificielle s'est imposée à nous afin de faciliter la création de telles ressources textuelles annotées. Cette option a, par ailleurs, apporté son

lot de problématiques propres entrecroisées aux problématiques linguistiques citées plus tôt. En effet, en plus de la complexité linguistique que représente la classification des erreurs d'orthographe en français (spécifiquement les erreurs commises par les apprenants FLE), il faut que cette classification soit adaptable aux schémas de l'annotation automatique par réseau de neurones artificiels plus ou moins sophistiqués. Aussi, la question de la performance des modèles s'impose comme un problème central pour un projet de modélisation comme celui-ci. Puis évidemment, la dernière problématique réside dans le choix des modèles, au regard de ce qui se fait pour les locuteurs de l'anglais (natifs ou non), car tout comme la ressource, les modèles permettant la détection, correction ou l'annotation des erreurs en français viennent à manquer.

Un nouvel objectif vient alors se greffer à la création d'une ressource de textes FLE annotée en erreurs exploitable pédagogiquement. L'objectif de l'implémentation d'un outil automatique adapté au français et à ses apprenants. À notre échelle, lancer des outils relativement performants, calqués sur les outils anglophones ainsi qu'une ressource utilisable pour leurs entraînements, permettra peut-être d'ouvrir la porte à l'intelligence artificielle (apprentissage profond) dans le monde de la pédagogie, à l'image de l'enseignement assisté par ordinateurs.

Pour atteindre ces différents objectifs et bénéficier des outils nécessaires à un tel projet, j'ai eu la chance d'exercer pendant trois mois le poste de stagiaire puis de jobiste (emplois étudiant) au sein du Centre de traitement automatique du langage (CENTAL) à Louvain-la-Neuve et d'être supervisée par le professeur Thomas François et le docteur Rodrigo Wilkens. Au travers du CENTAL, l'idée d'une ressource annotée a pu prendre une dimension tangible grâce à un corpus de texte d'apprenants du français, mais aussi grâce à la possibilité de réaliser une véritable campagne d'annotations dudit corpus afin de constituer la base de l'entraînement des modèles. Le projet présenté ici trouve son cadre évidemment dans un mouvement de bien plus grande envergure lancé par le CENTAL : « CECR-FR Rator », qui constitue un projet d'évaluation automatique de la compétence écrite des apprenants du FLE, pour laquelle une ressource annotée est aussi nécessaire.

Ces pages ont alors pour but de retracer le cheminement vers la résolution des problématiques posées plus haut et ainsi atteindre les deux objectifs de ce mémoire, la création d'une ressource annotée en erreurs et de modèles permettant d'annoter automatiquement les erreurs dans des textes d'apprenants.

Le premier chapitre sera dédié au support théorique du projet. Ce support théorique s'ancre dans la linguistique pure avec les questions autour du concept d'orthographe, d'erreur et de norme. Le support linguistique est aussi obligatoire dans l'élaboration de la typologie d'erreurs qui permettra de classer par la suite les erreurs des apprenants. Cette partie linguistique permet donc de placer le

cadre scientifique nécessaire pour appréhender l'orthographe, la pédagogie et les particularités propres aux apprenants et au français. Ce premier chapitre portera aussi sur des considérations techniques liées à la modélisation et à l'aspect technique et informatique que la manipulation de réseaux de neurones artificiels exige. Le deuxième axe théorique est centré autour d'une revue de la littérature scientifique dans le domaine de la détection, de la correction, de l'annotation et du diagnostic d'erreurs automatiques. Dans un premier temps pour l'anglais, puis par extrapolation, pour le français et pour les erreurs d'apprenants.

Le deuxième chapitre retracera les choix méthodologiques qui forgent ce mémoire. Dans ce chapitre, toutes les étapes de création du projet seront détaillées et explicitées le plus rigoureusement possible. C'est à ce stade que l'opportunité de mon stage prend tout son sens, car c'est dans ce cadre que la plupart des choix méthodologiques ont été faits : la création de la typologie, la campagne d'annotation et la campagne de modélisation. Ce chapitre permet aussi d'ancrer les choix dans le cadre théorique posé dans le chapitre précédent.

Le troisième chapitre de ce mémoire reprendra les différents résultats du projet. Seront alors explorés différents aspects de résultats comme :

Les résultats de la campagne d'annotation et le degré de validité de la ressource constituée grâce à cette campagne.

La ressource en elle-même, grâce à une rapide analyse de cette dernière, analyse des erreurs et des performances.

Les différents résultats, performances des modèles implémentés et présentés dans le chapitre précédent.

Ce chapitre présentera aussi une discussion autour de la ressource et de la typologie, la question de l'efficacité des classes d'erreurs sur le plan linguistique, pédagogique et technique y sera soulevée. Puis une discussion autour des performances techniques des modèles testés sur les ressources francophones (le corpus sera décliné en différentes versions de test) et des différences de performances face à leurs homologues implémentés pour des corpus anglophones.

Chapitre 2 : Cadre théorique.

2. Cadre théorique :

Avec les objectifs principaux qui l'animent, pédagogiques et techniques, ce travail s'inscrit dans une double cadre théorique. D'une part, le projet d'annotation s'ancore dans la lignée des études purement linguistiques et des considérations pédagogiques. Et d'autre part, l'aspect automatique du projet nécessite le support théorique des études techniques dans le domaine de l'annotation automatique, et plus généralement du traitement automatique du langage. Cette section constitue alors le socle théorique préalable à tous les projets de TAL, à la fois un questionnement (socio)linguistique et une approche technique et pratique. La double problématique est élaborée ici, en quatre grandes sections. Il est, premièrement, nécessaire de prendre en compte les recherches linguistiques autour du concept même de l'orthographe et de l'erreur. La première section tentera donc de brosser le portrait de cette norme qu'est l'orthographe au travers de diverses théories linguistiques et sociales. Ce support théorique se concentrera également sur la problématique de l'erreur et de son traitement, plus particulièrement dans le cadre de l'enseignement du Français langue étrangère (FLE).

La seconde section de ce cadre théorique traitera de la question centrale de la présente recherche. En effet, le principal intérêt de l'annotation des erreurs étant de signifier à l'utilisateur qu'une erreur a été commise, cette section couvrira les notions de pédagogie « autodidacte », de l'intérêt du diagnostic d'erreurs et, enfin, de l'apport du TAL à l'apprentissage et à l'enseignement (*ELAO, enseignement des langues assisté par ordinateurs*).

L'objectif principal de la troisième partie est de retracer les intérêts de l'annotation automatique et les travaux réalisés dans ce domaine. Elle constitue donc le socle pratique et computationnel à proprement parlé.

Pour finir ce cadre théorique, la dernière et quatrième section concernera les véritables problèmes rencontrés lorsqu'il est question de traitements automatiques du langage naturel. Les problèmes traités seront entre autres ceux liés à la langue cible, surtout lorsque celle-ci n'est pas l'anglais, les problèmes d'accès aux données, mais également les problèmes de conceptualisations linguistiques liés au Français et au contexte du FLE.

L'objectif central est évidemment de fournir les informations nécessaires pour mener à bien un tel projet, mais également de contextualiser la problématique principale de ce projet de recherche.

2.1 l'orthographe, une norme ?

Avant d'envisager d'annoter les erreurs d'orthographe dans un corpus ou même de les détecter, il est primordial de comprendre et de définir les concepts de l'orthographe et de l'erreur. Qu'est-ce qu'implique cette norme d'écriture qui régit la transcription de l'oral à l'écrit ? En français surtout, c'est un concept qui prête à débat aussi bien chez les experts de la langue que chez le commun des mortels pour qui orthographe est synonyme de grandeur d'esprit et d'éducation. Il s'agit donc pour commencer de définir et de retracer l'évolution de la norme qu'est l'orthographe. Puis de caractériser les débats qui font rage dans la sphère linguistique autour de la norme et de conceptualiser l'idée même des erreurs. Il convient évidemment de contextualiser ces considérations dans le champ d'action de cette recherche, l'enseignement des langues, plus précisément du français. Toujours dans l'idée de caractériser et de distinguer l'erreur de la norme, je m'attellerai donc à la présentation de certaines typologies (taxonomies) d'erreurs représentatives des débats autour de ces dernières. Pour finir le questionnement autour de ces normes linguistico-sociales et ces débats autour du concept de l'orthographe, je retracerai les différents points de vue sur ces sujets dans le monde du TAL.

2.1.1 La place de l'orthographe française : définitions sociales et historiques.

Le français est reconnu pour être une des langues romanes les plus complexes, et ce, pour la simple et bonne raison que son système graphique semble complètement dissocié de l'oral. Pour définir l'orthographe et par conséquent définir le concept de graphie, il faut avant tout se pencher sur le manque de correspondance entre les graphèmes (unités graphiques/ écrites) et les phonèmes (unités phonologiques/ orales) de la langue. Retranscrire et classifier les règles du français et les normes d'écriture semble être une tâche plus ardue que pour d'autres langues. Les racines historiques et le rôle de l'écriture dans l'élaboration et l'évolution du français en sont les raisons les plus évidentes.

Dans un entretien retranscrit par Renée Holvault-Durocq, Jacques Chaurand explique les origines du français écrit par l'envie de « donner un vêtement au latin » (Chaurand et Goosse, 2006 :26). Dans l'ancien français, le but est alors de « traduire la langue parlée » en une forme « d'apparence latine » (Chaurand et Goosse, 2006 :27). La nécessité d'écrire le français ancien est basée sur le latin, ce qui en fait dès l'origine une écriture imparfaite pour sa fonction originelle de transcription d'une langue orale. L'envie de s'éloigner oralement du latin (concurrence entre le latin et l'ancien français dans les tâches religieuses et politiques) a forcé la langue écrite à s'éloigner du latin. Éloignement qui engendre dès lors une complication lorsque l'on compare le système graphique du français à celui des autres langues romanes

(phonétiquement plus proche du latin) comme le roumain, l'espagnol ou l'italien (Chaurand et Goosse, 2006 :42).

Par la suite, l'évolution du français écrit fut forgée par les considérations sociales de prestige, catalysées par l'Académie Française. En plus d'être basée sur une autre langue et de tenter de s'en éloigner à tout prix, cette dernière, comme les formes les plus anciennes de l'orthographe, confèrent un prestige à quiconque l'utilise. Lorsqu'il a fallu distinguer les mots qu'une orthographe trop proche du latin rendant homophones (1), l'étymologie ou la « pseudo-étymologie » a servi de guide. Il a suffi d'ajouter des lettres (bien souvent muettes) afin de distinguer, d'informer et de justifier ces ajouts et incohérences par le prestige de l'ancêtre latin.

1. *vit* en ancien français (proche du latin) qui servait de transcription à la fois au mot [vît] et au mot [vi]. L'ajout du *h* muet permet de les distinguer (avant l'arrivée de l'imprimerie qui introduira le caractère « u »).

(Chaurand et Goosse, 2006 :36)

L'histoire du français écrit semble donc être à mi-chemin entre enracinement maladif et réformes successives des normes. Dans son article retraçant l'histoire de l'orthographe, Nina Catach (1973b) retrace l'évolution de cette norme depuis les prémices du français (comme expliqué plus haut) jusqu'aux considérations actuelles.

Elle mentionne alors, que les méthodes palliatives d'ajouts de lettres muettes, de doubléments de consonnes ou encore de dérivations ne sont que les conséquences d'un alphabet insuffisant, incapable de distinguer les sons (1973b :13). Elle mentionne également l'impact de l'imprimerie dans la première « révolution de l'orthographe » et du moyen français. C'est en effet grâce à elle que le français se voit gratifié d'accents pour distinguer le nombre trop faible de voyelles (Catach, 1973b :14). À la renaissance, on voit apparaître une « nouvelle orthographe » allégée de certaines reliques du passé. Cette nouvelle pratique coexistera d'ailleurs avec l'ancienne orthographe (toujours synonyme de prestige et d'éducation). Au grand dam d'une orthographe qui se voulait plus simple, plus transparente et plus abordable, la coexistence de formes archaïque et de nouvelles formes astucieusement simplifiées donnera naissance à une complexité encore plus grande dans le système graphique (Catach, 1973b :14).

La coexistence ou la concurrence entre des formes nouvelles et leurs pendants latins ou ancien français créent une compétition dont le gagnant n'est autre que la forme la plus robuste au fil du temps. Une forme qui n'est donc pas nécessairement un choix conscient des usagers, ce qui rend la graphie d'autant plus aléatoire (Chaurand et Goosse, 2006 :36).

Pour en finir avec cet amas de complexités, le caractère normatif de l'Académie au temps de la Restauration n'aide en rien avec la réintroduction de l'ancien français, toujours motivé par le caractère glorieux et politique de ce dernier. En d'autres termes, l'histoire conflictuelle de l'orthographe et le fait qu'elle soit liée si étroitement à son passé font d'elle non seulement une proie facile aux critiques et aux réformes. Mais font également d'elle un monceau de règles tantôt cohérentes, tantôt archaïques tenant les unes aux autres par de vagues justifications linguistico-historiques. Et cela concerne aussi bien la graphie même des mots (les phonogrammes, morphogrammes et logogrammes de Nina Catach (1973a) expliquées plus loin) que les signes diacritiques, partie intégrante de l'orthographe lexicale.

Bon nombre de linguistes se sont essayés à définir et à formaliser les codes graphiques des langues orales. À l'origine, comme je l'ai mentionné, l'orthographe constituait uniquement un « système graphique qui fonctionne sur la notation des sons suivant le classique principe alphabétique » (Catach, 1973a :30). En effet, au XVI^e l'apprentissage du français écrit consistait seulement à apprendre son tracé et la valeur des lettres, ou des combinaisons de lettres (Blanche-Benveniste et Chervel, 1978 :12). Cette approche, bien qu'appréciable par sa transparence, est de toute évidence bien trop simpliste pour décrire et définir l'ampleur du phénomène graphique qu'est l'orthographe française.

Nina Catach en 1973a est l'une des pionnières ayant pour optique de décrire formellement les relations contradictoires et conflictuelles entre langue orale et écrite. Elle tente donc d'expliquer le système d'écriture comme un « plurisystème ». Les graphèmes (unités graphiques de la langue) ne sont, selon elle dans la conjonction, plus le pendant parfait des phonèmes (unités phonologiques de la langue) mais bien un engrenage de trois sous-catégories aux fonctionnements distincts. Le français écrit serait alors divisé en phonogrammes, morphogrammes et logogrammes. Avec cette division, Catach (1973a :42) fait littéralement fi des considérations grammaticales autour de l'orthographe et tend à concevoir l'écriture du mot pour le mot et non pour servir de palliatif aux manquements de la grammaire.

En explorant plus profondément le lien entre grammaire et orthographe, on s'aperçoit qu'une autre division formelle existe au sein de la communauté linguistique : la division entre « orthographe grammaticale » et « orthographe lexicale », cette dernière étant communément appelée orthographe d'usage. La séparation entre ces deux concepts se fait autour de deux axes principaux, l'axe morphologique et l'axe syntaxique. L'écriture d'un seul mot dépend donc ces deux axes distincts. Il en va d'ailleurs de même avec des constructions plus

complexes telles que des syntagmes ou des phrases (Chervel, 1973 :87). De ce fait, l'orthographe serait régie d'une part par l'usage, que seul l'imprégnation constante permettrait de maîtriser (Fayol et Pacton, 2005 / Chervel, 1973), et d'autre part par la règle. Selon Chervel (1973 :88), cette pratique purement pédagogique permet de conceptualiser le cheminement d'écriture qui part de la langue vers l'orthographe et de cette même orthographe, d'usage donc, vers la grammaire (explication grammaticale du phénomène). L'usage semble donc, ici encore, seul régisseur de l'écriture du français. Alors, comment expliquer la complexité de ces usages ? Un ensemble de « règles » a bien été créé au fil des siècles pour légiférer et uniformiser l'usage, ces règles retournent à la fonction première de l'orthographe, la transcription du système oral vers un système graphique.

Ici encore, la tâche de définition fait débat, certains linguistes tendent à considérer la langue écrite comme le pendant de la langue orale, les deux ne formant alors qu'une seule et même langue. Alors que d'autres, au regard des évolutions d'usage, voient les deux systèmes comme indépendants l'un de l'autre (Blanche-Benveniste et Chervel, 1978 :38-39). Cette vision presque bilingue démontre alors l'omission de la fonction première de la langue écrite : transmettre et faire perdurer un message.

Dans certaines langues cependant, l'unification des paradigmes oraux et écrit n'est en réalité pas une question. Pour le Turc par exemple, à quelques exceptions près, une lettre note un son et ne note que lui. Acquérir l'orthographe consiste alors à retenir les correspondances entre sons et lettres (Arrivé, 1993 :43). L'aspect fonctionnel de l'orthographe est donc à prendre en compte, surtout quand il s'agit de voir le statut social que l'orthographe permet d'acquérir. Outre son histoire tumultueuse, le « bien écrire » va de pair avec la lecture et le but premier de transmission du message. L'idée ici n'est pas d'emmener un jugement de valeur sur les lignes conductrices de notre orthographe. Mais bien d'en démêler toute la complexité dans l'unique but d'en améliorer l'acquisition et la pratique.

Pour en revenir à la grammaire, c'est aussi elle qui a forgé les multiples règles qui normalisent l'orthographe. Le français étant une langue flexionnelle, les suppléments graphiques créés au fil de l'histoire ne sont pas exempts de règles plus ou moins suivies et cohérentes (Jaffré, 2005 :28-29). D'ailleurs, Chervel en 1973 extrapole ce lien étroit entre grammaire et orthographe et en tire une définition en trois plans. Le plan phonographique qui concerne uniquement la transcription des sons à l'écrit, les ajouts permettant de dissocier les homophones (souvent enraciné dans l'histoire) et les spécificités de formation de chaque mot du lexique (étymologie ou morphologie interne). Le plan morphographique, propre à chaque mot, qui régit le comportement flexionnel des mots de la langue (2-3).

2. *chat* a au pluriel la forme flexionnelle *chats*
3. *Cheval* a au pluriel la forme flexionnelle *chevaux*

(Chervel, 1973 : 87)

Ces caractères flexionnels, comme le souligne Jaffré (2005 : 28) sont de cinq grands ordres : le nombre, le genre, la personne, le mode et le temps (pour les verbes). Enfin, le dernier plan est le plan syntaxique qui a une place discutable dans « l'orthographe » à proprement parlé, mais dont les règles et les lois sont bien souvent la cause principale d'erreurs et dont l'influence agit sur la forme écrite de la langue et donc la représentation écrite du réel.

Quoi qu'il en soit, les considérations des linguistes sont une chose, mais le système graphique du français est aussi une affaire publique, presque politique. C'est peut-être dans l'opinion publique que l'on trouvera une définition plus pratique de l'orthographe et de son importance en francophonie. Cet impact social est d'autant plus important quand il s'agit d'acquérir la norme d'une langue. C'est sur la base des conceptions sociales décrites ci-après que j'aborderai par la suite les questions d'erreurs et de normes qui jalonnent l'élaboration d'un outil à visée pédagogique et théorique. Considérer la langue comme une pure production sociale ou sociétale est aisé, mais qu'en est-il pour le système graphique de cette dernière. Le langage « naturel » évolue comme son nom l'indique par l'usage, mais l'écrit, lui, n'a selon Bernard Gardin (2006a :110) plus rien de naturel. Comme il a été démontré, l'orthographe exhibe à la fois les sons de la langue orale, la grammaire, l'histoire et la structure du lexique. C'est ce caractère artificiel qui pousse ses fondateurs à considérer l'orthographe comme une pratique inerte, une construction figée. Oubliant totalement de conserver son aspect fonctionnel, créant ainsi un système formel qui se vide de sa fonction.

L'écart entre culture morte imprimée dans l'écrit et l'évolution vivante de la langue se creuse (Gardin, 2006a :112). D'autant plus quand il s'agit d'éducation, Gardin fait d'ailleurs une habile comparaison entre les différents outils d'aide à l'orthographe et une calculatrice. Le point social de l'écrit transforme son apprentissage en une autorité abstraite de la norme. Le « devoir de mémoire » de l'orthographe soutenu par l'opinion publique ne prend d'ailleurs pas en compte les avantages d'outils modernes (Gardin 2006a :113).

Plus encore, Sautot (2001 :194) soulève le caractère subjectif de la norme. La position plus ou moins normative du lecteur impactera sa perception d'un écrit. La position presque vindicative des sociétés savantes au regard d'une sainte norme n'a fait qu'accentuer les prises de positions fortes au regard du caractère licite de certaines constructions écrites (Sautot, 2001 :194). Ces normes controversées sont pourtant si nécessaires pour la standardisation et la simplification des tâches humaines comme la communication et la lecture (Sebba, 2007 :160), mais également utiles dans des tâches automatiques comme la création d'un moteur de

recherche. C'est sans le moindre doute ces nouvelles avancées dans le monde de la linguistique (nouvel accès à un grand nombre de données, traitement de texte automatisé, plus grand débit d'impression) ainsi que la vulgarisation d'un sujet longtemps perçu comme élitistes qui ont poussé le commun des mortels à remettre une fois de plus la sacro-sainte orthographe en question.

Plus récemment encore, l'arrivée de l'écriture inclusive en français bouscule une fois de plus les normes et leurs perceptions. Cette nouvelle version de la norme a pour but de témoigner d'une conscience des genres. Elle préconise de créer des nouvelles formes pour contrecarrer la primauté du masculin très présente dans la langue française (Grouin, 2022 :127-128). Dès lors, une question se pose : si l'orthographe du français se définit entre autres par son caractère réformable et modulable, comment l'apprentissage de cette norme peut-il et doit-il se faire ? Par ailleurs, quelles normes choisir et faire appliquer aux machines dans l'idée de créer quelconque correcteur ou annotateur d'erreurs orthographiques, surtout en français.

2.1.2 L'erreur, rectifications de la norme et impact.

L'erreur, tiens, parlons-en. Si l'on suit une définition classique de cette dernière, prenons celle du Larousse (4), entre autres dictionnaires émérites et incontestables (pour les utilisateurs lambda de la langue). Dictionnaires qui font office, d'ailleurs, de guides ou de références en termes d'orthographe depuis la publication du premier *dictionnaire de l'académie* en 1694 (Catach, 1973b :16 / Honvault-Durocq, 2006 :226).

4. Erreur (*nom féminin*) :
« Tenir pour vrai ce qui est faux »,
« Chose considérée comme erronée, fausse par rapport à la vérité, une règle ou une norme ».

Définitions issues du Larousse en ligne,
(<https://www.larousse.fr/dictionnaires/français/erreur/30846>) .
Consultée le 16/01/2023.

L'erreur, d'orthographe ou non, serait une contradiction d'une vérité (absolue) ou d'une norme qui en serait la directe représentante. Or, dépendant du contexte, la notion d'erreur peut se parer d'injonctions et de connotations toutes autres, voire différer complètement de la définition absolue qu'en font les dictionnaires. Le contexte scolaire, celui qui nous intéresse le plus ici, a encore, pour beaucoup, une vision très traditionaliste de l'erreur. Selon Yves Reuter (2013 :17), la vision de l'erreur à l'école se définit en trois points cruciaux.

Le premier point n'est autre que le poids considérable qu'a l'erreur dans le parcours scolaire. La valeur accordée à l'erreur, d'orthographe et de grammaire surtout, sera développée plus loin dans cette section. Mais pour l'heure, le second point définit plus précisément quel est ce concept qui a, semble-t-il, une si grande importance dans la vie des écoliers, pendant et après leurs passages sur les bancs. Selon Reuter, dans le contexte scolaire, la notion d'erreur est peu construite « sur le plan définitionnel, sur le plan descriptif ou encore sur le plan de l'étayage explicatif ». Un manque criant de cadre que Reuter (2013 :20) explique par le concept d'*absence*. Et quand l'erreur est spécifiée, elle l'est bien souvent par le biais de sa désignation ou sa catégorisation directe dans le contexte de son apparition. En d'autres termes, on ne définit pas l'erreur, en tant que concept, on se contente de la pointer lorsqu'on la remarque. De plus, remarquer une erreur serait alors le symptôme d'un consensus, « une erreur est une erreur » universellement et de manière intemporelle (Reuter, 2013 :22). Or l'histoire nous l'a montré, cette norme qu'est le langage n'a de cesse d'évoluer au gré de son utilisation. Comment l'erreur d'orthographe peut-elle subsister pour elle-même si celle-ci dépend d'une norme aux paramètres variables et subjectifs ? Reuter (2013 :24) propose également une approche fonctionnelle à la définition traditionnelle de l'erreur. Cette dernière aurait pour fonctions premières d'évaluer et de confirmer. Évaluer, car elle représente la sanction, somme d'une mauvaise assimilation des concepts abordés, et confirmer dans le sens où elle démontre encore une fois que dans une vision traditionaliste, l'erreur est utilisée comme une source de mépris et non comme un outil.

En effet, si l'erreur est la déviance face à une norme, celle-ci permettrait de mesurer son écart avec la production. Ainsi évaluer le poids, la valeur de cette production au regard des erreurs commises (Ravestein et Sensevy, 1993 :2). Mais encore une fois, comment établir une norme, les critères pour le français étant bien souvent fixés sur un « français standard » très normatif. Alors que dans le cadre des apprenants FLE, par exemple, selon leurs objectifs, leurs niveaux et le lieu de l'apprentissage, cette notion de standard et de norme connaît une très grande variété (Rahmatian, 2007 :109). Le cas de l'apprentissage est très particulier en ce sens : on attendra rarement d'un apprenant dont le niveau est faible (jeune enfant ou FLE de niveau A1/ A2) qu'il respecte en tout point les normes graphiques, syntaxiques et même stylistiques lorsqu'il se prête à un exercice d'écriture (Rahmatian, 2007 :109).

Dans cette idée de gradation de la norme, on voit également émerger un autre terme, tantôt interchangeable avec l'erreur, tantôt moins imprégné de la notion d'échec, la *faute*. Certains linguistes s'attellent donc à différencier l'erreur de la faute, les deux termes étant, en orthographe, réduits à l'appellation unique « faute d'orthographe ». Un écart de la norme sera appelé « faute » quand il prend la forme d'un lapsus ou d'une inattention, une coquille par

exemple. L'erreur quant à elle existe quand l'apprenant ne sait pas, quand l'écart à la norme est dû à une méconnaissance de la norme graphique. Cette distinction rappelle d'ailleurs les théories de Chomsky où la faute se rapporterait à la performance et l'erreur à la compétence, dans ce cas-ci, de l'apprenant (Delen Karaagaç, 2006 :126).

Malgré ces échecs en termes de formalisme (dû dans le cas de l'orthographe à une norme mouvante), il reste des pans du concept même d'erreur qui demeurent inexplorés.

Premièrement, il faut traiter les possibles causes et les sources de ces erreurs. Nul doute que la difficulté de la norme y est en grande partie responsable. Comme précisé plus haut, la grande complexité du français est son alphabet trop peu adapté aux quelque 30 phonèmes qui composent la langue (Jung, 1973 :101). Un décalage qui a pour origine l'écart progressif qu'a pris le français face à la langue latine d'où il tient son alphabet.

Une autre difficulté source d'erreurs en français est le fait que les flexions grammaticales ne soient pas différenciables à l'ouïe (5). L'enseignement de l'orthographe dite grammaticale est alors considérablement complexifié.

5. *Girl* en anglais sera phonétiquement différent de *Girls*
[gɜ:l] vs [gɜ:ls].
Alors que *fille* en français est un homophone de *filles*.
[fij] dans les deux cas.

(Jung, 1973 :103)

Ces morphèmes grammaticaux sont d'ailleurs parfois prononcés dans le cas des liaisons. Elles sont très présentes en français et elles rendent une fois de plus la pratique de la langue écrite complexe (Jung, 1973 :104).

Deuxièmement, il est nécessaire de traiter l'impact et la valeur attribuée traditionnellement à l'erreur. Et confronter cette vision à de nouvelles conceptions essentiellement pédagogiques. L'aspect social pourrait éventuellement éclairer notre lanterne et nous permettre de comprendre pourquoi un élément aux contours si fluides a une telle importance dans l'acquisition de l'orthographe. En réalité, d'un point de vue purement pédagogique, l'erreur est le principal bloc constitutif de l'apprentissage (Ravestein et Sensevy, 1993 :112).

En 1984, Yves Reuter plaidait pour une nouvelle pratique de l'erreur, essentiellement dans le contexte de l'apprentissage. Il constatait entre autres l'impact négatif que le concept d'erreur avait socialement (1984 :117). En 2013, il persiste en attribuant à l'erreur un « poids considérable [...] aussi bien dans les discours sociaux que dans les discours scolaires » (Reuter, 2013 :17). Ce poids et cette valeur négative attribués à l'erreur d'orthographe vont évidemment de pair avec la valeur de prestige et d'éducation qu'a l'orthographe auprès du public francophone. Une faute ou une lacune dans ce domaine est réduite à un manque

d'éducation ou une faible grandeur d'esprit, au même titre que les académiciens privilégiaient « les formes *anciennes* pour faire la différence entre les savantes, les autres » (Gardin, 2006a :113). De plus, l'erreur est un marqueur d'échec qui repose sur l'élève uniquement (ou plus largement sur l'auteur de l'erreur) (Reuter, 1984 :118). En aucun cas la transmission du savoir et la nature du savoir ne sont ni remis en question ni mis en faute comme la cause possible des erreurs d'orthographe. Ce qui mène à un discours culpabilisant et anxiogène autour de l'erreur. Celle-ci existant de surcroît dans un système élitiste d'évaluation et de classification presque morale des auteurs (Reuter, 1984 :118). Extrêmement liée aux conceptions sociales, l'erreur telle qu'elle est envisagée traditionnellement perd sa fonction d'indicateur d'une évolution bien utile dans le cadre de l'apprentissage. D'autres manières de considérer les erreurs seront explorées dans les sections 2.1.3 et 2.2, la seconde traitant plus amplement de leur traitement dans le cadre d'outils assistés par ordinateurs.

Néanmoins, toutes ces considérations mènent donc à un seul et même constat, une réelle impossibilité d'avoir un cadre clair pour les erreurs. Pour l'heure, nous garderons comme ligne conductrice l'idée qu'une erreur correspond à un écart vis-à-vis d'une norme. Une norme qui porte à question, peut-être faut-il s'arrêter un instant sur les différentes normes et réformes qui concernent principalement les apprenants. En plus des débats sur la nature et l'impact de l'orthographe, le récent débat au sujet de certaines inepties dans la norme académique remonte au début des années 1990.

Un projet de régularisation de l'orthographe qui propose un certain nombre de réformes des points les plus incohérents de l'orthographe prend place et sème la discorde dans la francophonie tout entière (Honvault-Durocq, 2006 :225). La motivation principale étant de remettre l'orthographe à sa juste place, une orthographe qui « sert à écrire » (Gardin, 2006b :121). Le problème majeur de ces rectifications est que l'orthographe représente une institution. Ces travaux de régularisation ont lentement été discutés par l'état et les instances légiférantes (Honvault-Durocq, 2006 :226-227) mais ont mis, en Belgique par exemple, plus de 15 ans à être pris en main par l'état. En effet, ce n'est qu'en 2008 qu'une circulaire « invite » les enseignants à « enseigner en priorité les graphies rénovées »¹ tout en acceptant pour « une durée indéterminée », c'est encore le cas en 2023, la coexistence des formes anciennes et nouvelles .

1 Circulaire 2475 du 29/09/2008,
lien vers la circulaire: http://www.enseignement.be/index.php?page=26823&do_id=2675
(site de la fédération Wallonie-Bruxelles, pour l'enseignement) dernière visite le 10/03/2023.

Ceci couplé aux vives réactions d'une partie de la population au concept de « réforme » car il la perçoit comme trop radicale et est trop attachée à des idées telles que la beauté ou le charme de ses règles archaïques. Le tout est d'ailleurs agrémenté du devoir de mémoire et de l'idée de la « culture des belles lettres, des mots » (Cibois, 1991 :89). En réalité, l'Académie préconise à l'origine de faire confiance à l'usage, puis progressivement ses rectifications sont devenues des « recommandations » (Honvault-Durocq, 2006 :233). À tel point que le *dictionnaire de l'académie*, neuvième édition, fait alors part de 100% des rectifications. Cependant, les dictionnaires d'usage comme le *Larousse* et le *Robert*, eux, n'en présentent que 66 % témoignant ainsi de l'hostilité à l'égard des modifications (Honvault-Durocq, 2006 :235). Une fois de plus, c'est cette absence d'harmonisation et d'obligations qui rendent la tâche de définition des erreurs et plus largement de l'orthographe ardue. À tel point qu'à l'heure actuelle, il est reconnu trois usages bien distincts, l'ancienne orthographe uniquement, la nouvelle orthographe uniquement et l'utilisation des deux (Honvault-Durocq, 2006 :243).

Comme Rahmatian le mentionnait habilement, la norme peut donc varier en fonction des mœurs ou des régions de la francophonie où l'orthographe est appliquée (2007 :109). Elle va même changer sensiblement selon les ouvrages de référence et *grammaires préconisés*. Une mention qui se voit être confirmée par une étude menée par Buisson-Buellet et al. en 2012. Ils mettent en évidence les différences énormes de sentiments à la réception de cette « nouvelle orthographe » entre des pays du sud de la francophonie (Maroc, Algérie), du nord (France, Belgique, Suisse) et du Québec. Et entre ces pays du « nord » des différences seraient encore à soulever. Une seule chose est certaine, les enseignants, de toutes parts, sont maintenant « invités à enseigner cette graphie rénovée » (Farid, 2012 :2056). Pour concrétiser cette idée de graphie rénovée, de rectification ou encore de réforme, voici un tour d'horizon des points majeurs qu'elle aborde (Farid, 2012 :2057-2058).

A. Les traits d'unions et les mots composés :

Les numéros gagnent un trait d'union (*huit-cent-mille*)

Le pluriel des noms composés non soudés change : flexion du second terme.

La plupart des noms composés sont d'ailleurs soudés (*mangetout, branlebas*)

B. Les accents :

Une série de mots écope d'un accent grave (*allègement, cèleri*)

Une série de mots écope d'un accent aigu (*recéler*) ou des mots d'emprunt/d'origine latine.

L'accent circonflexe sur la voyelle *u* tombe et le tréma ne sert qu'à rendre une lettre audible.

C. Le pluriel des noms empruntés :

Ces mots suivent les règles françaises du pluriel (*maximum - maximums*)

D. La rectification d'un certain nombre d'anomalies : (annexe 1).

En somme, l'erreur semble résider dans l'appréciation du lecteur (professeur ou non) et dans l'application d'une norme plutôt qu'une autre. Il serait bon également de rappeler dans le cadre du FLE le concept de « relâchement de la norme », qui, comme je l'expliquais plus tôt, pousse les enseignants à considérer la norme plus ou moins fidèlement en fonction du niveau de l'élève. Ce constat affligeant soulève alors deux problèmes majeurs, l'acquisition de l'orthographe, surtout dans les classes de FLE, ne serait pas conflictuelle dès l'origine. Et ensuite, nous sommes en droit de nous questionner sur la bonne méthode pour rendre des outils informatiques suffisamment efficaces et compréhensifs si un consensus autour de la normalisation est impossible à obtenir.

2.1.3 La norme dans le contexte éducatif, enjeux du FLE et analyse des erreurs.

Dans la vision traditionaliste de la pédagogie, l'orthographe a longtemps été qualifiée comme « la science des ânes » (David, 2006 :169). L'élève se contentait de mémoriser « bêtement » une liste de règles et sa liste d'exceptions correspondantes, la moindre question sur l'orthographe d'un mot ou l'absurdité d'application d'une règle était d'ailleurs soldée d'un « parce que c'est comme ça ». Les élèves étaient donc de deux catégories, les bons, ceux pour qui l'orthographe ne posait pas de problèmes majeurs et les mauvais. Une approche somme toute passive de l'acquisition, où « l'orthographe s'impose aux élèves » (David, 2006 :171). Aucune attention n'est portée aux logiques sous-jacentes et à la compréhension des mécanismes. Cette vision réductrice de l'orthographe s'expérimente aussi lorsque l'on cherche à évaluer la position de l'école, des professeurs qui la composent, face aux rectifications expliquées plus haut. Dans une étude menée par Jean-Pascal Simon auprès de futurs enseignants, l'attitude de ceux-ci face aux nouvelles normes semble être bien inégale. Premièrement, selon le genre de l'enseignant, la tendance à appliquer plutôt la nouvelle ou l'ancienne orthographe varie énormément. Deuxièmement, selon les règles à appliquer, il se dégage une tendance globale pour certaines réformes à privilégier l'ancienne et pour d'autres la nouvelle graphie. Enfin, pour un bon nombre de règles, la tendance semble rester indéterminée (Simon, 2006 :209-210).

Encore une fois, le caractère symbolique de l'orthographe (6) et la « peur » des modifications lorsqu'elles s'éloignent énormément de l'ancienne graphie semblent en être les

facteurs principaux. On pourrait souligner aussi un certain effet d'habitude qui pousserait les utilisateurs d'une norme précise à en rejeter une autre, simplement comme un enfant rejette un aliment par manque d'habitude et non parce qu'il l'a goûté.

6. *Nénuphar*, rectifié en *nénufar* à la suite d'une fausse interprétation ethnologique (le mot venant en réalité de l'arabe et non du grec). Mais qui reste majoritairement orthographié avec *-ph* pour le « prestige » de sa prétendue origine.

(Chaurand et Goosse, 2006 :42)

Outre ces réflexions sur la norme dans le cadre scolaire, le plus complexe est d'évaluer la qualité de l'écrit de l'apprenant (Halsted, 1975 :76). En effet, bien que les règles nous permettent d'évaluer à quel point l'auteur de l'erreur s'en éloigne, le processus d'écriture est si complexe que bien souvent, il est difficile de placer une limite claire entre un « bon » écrit et un « mauvais » écrit. Cela est d'autant plus vrai dans l'enseignement d'une langue seconde. Qu'est ce qui justifie qu'un étudiant qui commet occasionnellement des « fautes » d'oubli est en réalité plus à même de monter de niveau qu'un autre qui maîtrise la grammaire basique (Halsted, 1975 :76), l'appréciation subjective du professeur ? Un cadre commun à l'apprentissage des langues existe bien, ce cadre vient alors se greffer aux normes déjà peu claires de l'orthographe. En effet, le *Cadre européen commun de référence pour les langues : Apprentissage, enseignement, évaluation* (ou CECR) est le fruit de nombreuses années de recherches en linguistiques et existe depuis sa publication en 2001. Il est composé de niveaux (allant de A1 à C2), chacun de ces niveaux comporte une liste de compétences nécessaires à l'évolution de l'apprentissage. Il a avant tout pour but de créer un cadre clair pour l'élaboration des programmes linguistiques (Strujić, 2021 :2). Un étudiant de niveau A1 ne sera donc pas évalué avec la même grille d'erreurs qu'un niveau C2, et ce dernier ne sera pas non plus évalué au même titre qu'un natif.

De plus, en restant obnubilé par l'évaluation des erreurs, les enseignants ont tendance à perdre de vue l'objectif principal de l'apprentissage d'une langue et de l'écriture : la communication (Halsted, 1975 :72). Une autre approche de l'enseignement des langues est alors envisageable, en s'appuyant sur l'expérience des élèves pour comprendre les raisonnements derrière les erreurs ou pour situer l'erreur par rapport au savoir (David, 2006 :178-179). Pour comprendre cela, il faut avoir recours à une analyse linguistique plus poussée : l'analyse d'erreurs. Approximativement 39 % des recherches en terme de didactique de l'orthographe sont en réalité des recherches descriptives (Jaffré, 1992). L'analyse et la comparaison des erreurs permettent entre autres d'arriver à une description solide des

processus d'erreurs chez les publics analysés (Katoozian, 2013 :65). Notamment, cette approche servirait dans les domaines didactiques, en comparant par exemple les erreurs des élèves en français langue maternelle (FLM) et celles des élèves de FLE (Katoozian, 2013). Mais, plus largement, cette méthode peut permettre des analyses diachroniques des erreurs, par conséquent mesurer les changements de la langue ou même servir dans le domaine de la linguistique forensique (linguistique appliquée au domaine légal).

Aussi prometteuse que cette technique puisse être, elle nécessite évidemment deux choses cruciales, d'abord, une typologie des erreurs claires et une catégorisation précise pour pouvoir extraire une description utile (Burt et al.,1982 :139). Ensuite, il faut avoir accès à suffisamment de donnée, annotées en erreurs de préférence (James, 1998 :117). C'est là que le TAL entre en jeu, car, comme nous le verrons plus loin, annoter manuellement un grand nombre de textes est onéreux et extrêmement chronophage.

2.1.4 Annotation manuelle, typologies des erreurs d'écriture.

Il est donc nécessaire pour la bien portance du domaine de l'analyse d'erreurs d'avoir une classification précise de celles-ci. Cette classification est plus souvent nommée typologie d'erreur d'orthographe (ou TEO) (Sautot, 2020 :1) ou encore « taxonomie » dans les ouvrages traitant de l'anglais. Toujours selon Sautot (2020 :15), les TEO sont des outils pédagogiques très appréciables, mais pas seulement. Ils sont aussi les outils d'analyse théorique de la langue, selon le prisme linguistique choisi pour les élaborer, il est possible d'analyser différents aspects de la langue écrite ou des dynamiques du bilinguisme (Sautot, 2020 :2). Les TEO sont donc à la fois l'instrument théorique et l'outil qui permet de mettre en œuvre les considérations théoriques ainsi créées. Ces outils s'appuient sur des corpus d'erreurs et sont plus ou moins détaillés selon la tâche, le but de la typologie et le linguiste qui s'attelle à leurs créations.

Dans l'enseignement, les TEO sont avant tout un outil d'analyse destiné aux enseignants (Sautot, 2020 :16). Pourtant, ils sont pour l'apprenant un outil d'autonomisation face à l'erreur encore bien trop peu exploité. Par-dessus le marché, ils permettent de remettre de l'ordre dans le fouillis de l'orthographe française pour donner aux enseignants une grille d'évaluation concrète et adaptable selon le niveau et les lacunes du groupe classe (Sautot, 2020 :16). Plus formellement, les typologies sont des listes organisées selon des critères précis et utiles, ils peuvent ainsi se révéler commodes pour distinguer des cas d'erreurs ambiguës (James, 1998 :102). Ces critères, comme je l'ai dit, diffèrent en fonction du prisme linguistique et du but de la typologie. Plusieurs approches peuvent être suivies dans la création d'une typologie. La langue est évidemment un facteur important dans le choix du prisme linguistique, mais la

profondeur de la recherche en est un également. Ne sont présentés ici que les travaux « pionniers » ou majeurs de ces différentes approches. Il est évident que, pour l'anglais par exemple, d'autres typologies plus ou moins élaborées ont été créées (surtout quand on sait que tout travail d'annotation en nécessite une) mais elles s'articulent en général autour des approches citées ci-dessous.

Burt et al. (1982 :150-160) proposent une approche qu'ils qualifient « de surface ». Cette approche fait fi des considérations linguistiques et considère les erreurs telles qu'elles apparaissent à la surface du texte. Selon eux, les erreurs se classifient donc en 5 catégories (7). Les exemples sont en anglais d'origine, mais le grand avantage de cette taxonomie est qu'elle est adaptable à d'autres langues. La première classe sont les erreurs d'*omission*. Il est important de préciser que cette catégorie ne concerne que les cas où l'omission rend la construction inacceptable grammaticalement. La deuxième contient à l'inverse les erreurs d'*additions* qui fonctionnent exactement comme la classe précédente. En troisième et en quatrième position, on retrouve les erreurs de *malformation* (internes aux mots) et d'*ordres* (des mots dans la phrase). En fin, la cinquième classe concerne les erreurs dites d'*hybridation*. Dans cette catégorie, on retrouve en réalité le mélange de deux (ou plus) constructions grammaticales qui forment une construction inacceptable.

7. * *He'll pass his exam and I'll [0] too.* (Omission)
 - * *I didn't went there yesterday.* (Addition)
 - * *womans.* (Malformation)
 - * *He every time comes late home.* (Ordre)
 - * *The one only thing I want.* (hybridation)
 - [*The one thing I want* ou *the only thing I want*]
- (Burt et al., 1982 :150-160)

Richards (1970 :3-4) propose lui une classification toute autre. Il choisit pour sa classification le prisme de l'acquisition d'une seconde langue. En trois catégories distinctes, il explore les erreurs et les sources des erreurs typiques lors de l'acquisition d'une nouvelle langue. Ici aussi, la typologie et les exemples sont basés sur l'anglais, mais la typologie est évidemment adaptable à l'acquisition de n'importe quelle langue. La première catégorie regroupe les erreurs *inter-langagières* (interlingual). L'objectif est de distinguer une première fois les erreurs dues uniquement à l'interférence avec langue maternelle des deux autres catégories. Pour la deuxième et la troisième classe, Richards s'attaque aux erreurs *intra-langagières* (intralingual) et *développementales*. Ces erreurs peuvent encore être plus précisément classifiées. En effet, il propose en plus une classification précise qui se base sur la source de ces erreurs. De manière générale, elles concernent toutes deux les erreurs

d'applications des règles de la seconde langue et le manque de connaissances de la langue par les apprenants.

Un autre type d'approche existe aussi dans le monde de la linguistique. James (1998 :105) qualifie cette approche des TEO de « classifications par catégories linguistiques ». Plusieurs linguistes s'y sont essayés, tous, avec plus ou moins de précision. Entre autres, Sautot (2020 :5) cite une fois de plus la linguiste et historienne de la langue, Nina Catach. En 1986, elle propose une typologie basée sur les catégories linguistiques, son plurisystème (annexe 2). Dans sa proposition très aboutie et en adéquation avec sa tentative de description du système graphique du français, elle s'attaque avant tout aux erreurs commises par les natifs. Sa typologie est avant tout divisée en quatre catégories distinctes et s'articule autour d'une description formelle des types d'erreurs commises. Hors des divisions linguistiques communes, elle s'attache à son système propre composé de *morphogrammes* (erreurs à dominante morphogrammique) et *logogrammes* (erreurs à dominante logogrammique). Les *phonogrammes* étant un peu hors de l'écrit, elle mentionne également les erreurs « extra graphiques » comme une catégorie à part entière séparée des erreurs « graphiques ».

Plus formelle encore, il est intéressant de citer le projet d'annotation de corpus d'apprenants de l'anglais mené par Sylviane Granger (2022). Ce projet, contrairement à Catach, s'attaque aux erreurs des apprenants et plus précisément des apprenants de l'anglais, tout en utilisant des conceptions linguistiques plus générales et, de ce fait, plus adaptables à d'autres langues. Dans le manuel d'annotation « the error tagging manual 2.0 » la typologie d'erreur y est extrêmement complexe. Les catégories linguistiques telles que la morphologie, la syntaxe, le lexique et les ponctuations y sont reprises en plus des erreurs « de surfaces » comme la répétition, l'omission ou l'ordre des mots. Dans chacune de ces catégories, les erreurs sont reprises au cas par cas pour élaborer une taxonomie des erreurs des apprenants et permettre de simplifier considérablement l'annotation manuelle de textes.

Enfin, Câslaru (2016) propose une adaptation de la typologie de Granger simplifiée et qui correspond plus exactement aux enjeux des apprenants roumains du français (annexe 3). Ici la typologie se compose de trois grands niveaux interconnectés. La classification à proprement parler réside dans le niveau 1, les catégories linguistiques ainsi que les erreurs qui y sont associées (morphologie, syntaxe et lexique). Le niveau 2 regroupe le lien fort entre erreurs et la nature, la classe grammaticale. Et le niveau 3 reprend les travaux de Richards (1970) pour commenter le caractère inter-langagier de l'erreur. Ici les erreurs « de surface » sont incluses dans des catégories linguistiques comme la syntaxe par exemple.

Comme son nom l'indique, le traitement automatique du langage naturel traite évidemment en priorité les considérations autour du langage. Outre les avancées techniques que nous traiterons dans le point 2.3, il était primordial de théoriser et de cadrer les études dans le domaine de la langue pour pouvoir évaluer au mieux les outils de TAL (surtout ceux destinés à la recherche en pédagogie ou en linguistique). Le TAL a donc tout intérêt à se pencher sur ces réflexions (socio)linguistiques dans tous les domaines de son champ d'application.

Entre autres, les analyses d'orthographe et de normes sont importantes pour les phases de normalisations, préalables à toutes tâches de TAL (annotateur morphosyntaxique, moteurs de recherche, etc.) (Emirkanian, 1989). Mais ces questions sont aussi abordées dans le cas des aides à l'orthographe et à la grammaire que l'on retrouve dans des outils de traitement de texte par exemple (Fontenelle, 2003). En plus de cette approche pratique, le monde de la pédagogie aussi s'appuie de plus en plus sur des techniques computationnelles (applications interactives, sites web). La place de l'erreur traditionnelle se modifie peu à peu pour devenir un outil d'apprentissage. Et les réflexions sur l'impact socioculturel et l'absurdité de l'orthographe sont progressivement revues pour regagner une place plus utilitariste. En d'autres termes, comprendre et classer l'erreur permet donc de mettre en question l'impact et la portée de la « grande » orthographe, mais aussi de créer des outils numériques qui facilitent le traitement et l'apprentissage des langues.

C'est pourquoi le concept d'orthographe, bien que difficile à définir, se devait d'être clarifié. Clarification qui a pu être faite par le concept de norme vraisemblablement mouvante. La variation au sein même de l'orthographe est ce qui la rend si inaccessible tant actuellement qu'aux yeux de la société. D'autant plus dans le domaine de l'enseignement qui se bat constamment pour démêler les différentes visions et interprétations de l'orthographe. Néanmoins, avec l'évolution de la pédagogie et les outils adaptés, des campagnes de clarification ont vu le jour. Et avec elles, les linguistes se sont essayés à définir, coder et formaliser la norme en dehors des considérations historiques et sociales. Ce sont les typologies d'erreurs et l'envie de fournir une ressource utile aux travaux d'analyse d'erreurs qui nourriront les réflexions contenues dans la suite.

2.2 Approches pédagogiques, le cas du traitement automatique du langage.

Comme toujours, l'homme d'aujourd'hui cherche à simplifier ses tâches en les automatisant. La sphère de la pédagogie et de l'enseignement ne fait évidemment pas exception. Loin des pédagogies traditionnelles et de leurs problématiques, les acteurs de l'enseignement 2.0 ont de nouveaux besoins en termes d'outils et de ressources. Comme introduit dans la section précédente, la linguistique et la pédagogie ont de plus en plus besoin de ressources claires et utiles, ces ressources seront décrites

dans la première section. Cette section permettra aussi de justifier le besoin de telles ressources, surtout dans le monde de la pédagogie et du FLE.

La seconde section mettra en lumière les nouvelles pédagogies citées plus haut tout en expliquant le besoin actuel d'approches automatiques. Je m'engage dans cette section à décrire le TAL comme un outil, ici au service de la pédagogie, mais également plus loin au service de la recherche. Cette description se fera par le biais de projets pédagogiques numériques innovants et ayant donc besoin des ressources, TAL ou non, mentionnées dans la section précédente.

2.2.1 Annoter l'erreur, une ressource pédagogique nécessaire.

Dans un premier temps, quand il est question d'annotation d'erreur, il est question de corpus, des textes écrits par le public cible de l'étude et contenant évidemment des erreurs à annoter. Ici, l'attention se porte sur les corpus écrits par des apprenants des langues (le FLE dans le cadre de la présente étude). D'emblée, les textes d'apprenants, qu'ils soient natifs ou dans l'acquisition d'une langue étrangère, sont très peu représentés dans les corpus classiques de grande envergure (Wolfarth et al., 2017 :1). Pour pouvoir analyser les spécificités des écrits des apprenants, les linguistes ont donc recours à des corpus dits d'apprenants (*learner corpora*) ou des *corpus inter-langues (IL corpora/ L2 corpora)* (Granger, 2003 :465). Ces corpus sont donc des collections électroniques de textes. Leur grand avantage est qu'ils sont numériques, contrairement aux ressources habituelles en pédagogie, ce qui permet d'avoir, une plus grande quantité de données, et d'avoir recours aux technologies TAL pour traiter ces données (Granger, 2003 :465-466).

L'enjeu pour analyser ce type de base de données est double. En premier lieu, afin d'utiliser des outils informatiques poussés (autres qu'une simple analyse de fréquences), les textes des apprenants doivent être normalisés avec des outils adaptés aux spécificités des erreurs d'apprenants. Pour l'analyse d'erreurs également, les outils doivent être calibrés sur les spécificités propres aux écrits d'apprenants. En deuxième lieu, (Granger, 2003 :466), il faut réunir suffisamment de données, dans la langue cible, pour pouvoir traiter les données avec ces outils informatiques poussés. Cette problématique sera entre autres traitée dans le point 2.4 sur les enjeux du TAL pour le FLE. Ces corpus d'apprenant sont, à l'heure actuelle, composés essentiellement dans une optique de diagnostic d'erreurs (Wolfarth et al., 2017 :1) et de pédagogie adaptable ou autodidacte.

Ensuite, une fois le corpus constitué, les stratégies d'apprentissages peuvent se mettre en place. Habituellement créées manuellement, ces stratégies pour l'orthographe et l'écriture d'une langue seconde tournent toutes autour d'une problématique principale, le diagnostic

d'erreurs. C'est-à-dire comprendre la source, la raison et la nature de l'erreur pour adapter l'apprentissage. Les diagnostics sont donc utiles à la fois pour que l'enseignant adapte son enseignement et pour que l'apprenant lui-même adapte ses connaissances (James, 1998 :185). Pour l'enseignant, ces corpus et le diagnostic des erreurs peuvent permettre d'adapter les cours aux problématiques des élèves, grâce aux connaissances obtenues par l'analyse des corpus. Pour les étudiants, grâce à l'apport du TAL et des outils numériques, ces diagnostics sont utiles, par exemple, pour la production de retours pédagogiques, pour la suite, j'emploierai le terme anglais *feedback* plus concis.

Le *feedback* dans l'apprentissage des langues peut se définir comme une indication faite à l'étudiant que l'usage d'une norme est incorrect (La Russa, 2021 :71). Cette indication est faite pour permettre de détecter l'erreur et d'alerter l'apprenant, puis de l'orienter vers l'usage approprié de la norme (La Russa, 2021 :72). Il constitue donc une forme de soutien vers une amélioration autonome de l'étudiant. Ce processus permet à l'apprenant de recevoir une information sur la qualité de sa production (Poltorak, 2011 :32). Bien que certains linguistes considèrent la correction pure comme un type de *feedback* explicite l'intérêt principal de la production de *feedback* (métalinguistique) (Poltorak, 2011 :36) est de permettre à l'apprenant une recherche active dans la norme pour produire par lui-même sa correction (Bosc-Miné, 2014 :318). Les *feedbacks* peuvent donc être plus ou moins élaborés et plus ou moins explicites quant à l'information donnée à l'apprenant. Dans le cadre scolaire, c'est souvent l'enseignant lui-même qui, lors d'une « correction » donne un *feedback correctif* (La Russa, 2021 :75), on parle alors ici du *feedback traditionnel*. Mais avec les avancées technologiques de ces dernières décennies, l'enseignant peut se décharger d'une part de travail. Il laisse alors des outils pédagogiques adaptés produire des *feedbacks* et d'autres aides à l'apprentissage automatiquement, directement à partir de la production de l'apprenant et sur base des corpus et des diagnostics mentionnés plus tôt. Il est question ici de *feedbacks* électroniques (Poltorak, 2011 :36). Plus précisément encore, le domaine de recherche qui regroupe le pendant numérique de la pédagogie se nomme par les acronymes ALAO ou ELAO.

2.2.2 ELAO / ALAO, les apports et le besoin d'une approche automatique.

La première conception d'un apprentissage aidé par les mécanismes tiers naît en 1809 aux USA (Antonadis, 2010 :2211). Dans les années septante, après une vague idée d'un enseignement général assisté par ordinateurs, l'intérêt se porte sur la possibilité de rendre cet enseignement « intelligent », c'est là que l'*EIAO* (enseignement intelligemment assisté par ordinateur) naît (Antonadis, 2010 :2212). Depuis son origine, l'objectif premier est d'anticiper

et de diagnostiquer les erreurs ainsi que d'améliorer les stratégies d'enseignement en personnalisant l'outil aux besoins précis. La pédagogie linguistique ne met pas longtemps à s'intéresser à ce domaine grandissant. L'ALAO (apprentissage des langues assisté par ordinateurs) explose avec l'essor d'internet et a donc pour champ d'application des corpus représentatifs de la langue (native et des apprenants) avec comme approche : le « data driven learning » (Antonadis, 2010 :2214).

Avant l'essor du traitement automatique du langage et surtout des récentes approches par réseaux de neurones, le monde de l'enseignement semblait retissant à l'idée d'utiliser les outils d'ALAO (Antonadis, 2010 :2214). En effet, les experts déplorent un manque de fiabilité et de personnalisation dues aux approches par règles (trop peu fiables) et les approches statistiques (trop rigides et peu adaptables) (Mars, 2016 :25). Malgré un début en demi-teinte, on reconnaît à l'ALAO des avantages indéniables, entre autres l'optimisation du temps, la motivation et la réduction de l'anxiété, car la tâche est réalisée en autonomie (Mars, 2016 :15). De plus, avec les avancées actuelles dans les domaines de l'intelligence artificielle et du deep-learning (apprentissage profond en français, utilisation du terme anglophone plus répandu), le TAL du 21^e siècle pallie les limitations par une plus grande fiabilité et une meilleure optimisation en fonction des besoins d'enseignement (niveau de l'élève, lacunes particulières, langue maternelle) (Mars, 2016 :24).

En 1992, Chanier considérait déjà le TAL et l'ALAO comme l'avenir de la pédagogie, car tous deux centrés sur une « modélisation de l'apprenant » basée donc sur une analyse et une étude précise (supervisée par l'homme ou non) des corpus d'apprenants (Chanier, 1992 :27). Un autre grand avantage de l'ALAO est évidemment l'immédiateté du *feedback*, pour le cas de l'apprentissage de l'orthographe, une fois l'erreur détectée chez l'apprenant, le « tag » ou le type d'erreur peut être renvoyée indiquant de cette manière à l'apprenant l'endroit exacte où l'erreur commise doit être corrigée (Mars, 2016 :25). Ces retours peuvent, comme pour les *feedbacks* traditionnels, être peu explicites, simplement une mention « incorrect » à la place de l'erreur. Mais ils peuvent être aussi très explicites avec la génération d'un *feedback* complet indiquant le type d'erreur et des pistes de correction. Là où un *feedback* très général, ne traitant que du type d'erreur commise par exemple, s'apparentera à une tâche de classification (Hanawa et al., 2021 :9721). Un type ou une classe d'erreur attribué à une séquence générera une sortie prédéfinie correspondante aux erreurs de cette classe. Les *feedbacks* explicites, prenant en compte les mots exacts de l'entrée par exemple, devront bénéficier de structures plus proches de la traduction ou de la simplification (architecture *encoder-decoder*). En effet, ce type de *feedback* nécessite la génération d'une sortie adaptée directement aux mots présents dans la phrase erronée en entrée (8) et non une explication générale.

8. *I reached *to New York.*

Feedback comment : « reach is a transitive verb. It does not need a preposition before the object. »

(Hanawa et al., 2021 :9721)

Outre la génération de *feedback* qui nécessite des technologies poussées et plus d'attention de la part des chercheurs, l'annotation des erreurs dans un texte joue un rôle dans l'évaluation automatique de textes (Cutumisu et Lu, 2021 :73). Le programme se base sur une liste de caractéristiques prédéfinies (erreurs d'orthographe, seuils de compétence CECR, ...) pour évaluer le document par une note ou bien par un niveau CECR.

Il existe aussi d'autres approches d'ALAO innovantes et qui font écho à la place de l'erreur dans l'éducation. C'est le cas de « l'anti-correcteur » qui annote non seulement les erreurs mais aussi les formes correctes. De cette façon, l'outil valorise la connaissance au lieu de dévaloriser les failles et apporte une vision positive de l'erreur en pédagogie (Delbar et al., 2016 :5-6).

D'autres méthodes d'annotation plus fiables sont évidemment nécessaires pour gagner la confiance des enseignants et permettre aux chercheurs de théoriser l'enseignement des langues sur base de données complètes et variées.

Néanmoins, le domaine de l'ALAO, au regard des avancées du monde de la technologie depuis une petite dizaine d'années, connaît un nouvel essor. Des nouveaux corpus destinés à l'analyse d'erreurs et aux *feedbacks* sont composés depuis 2019 (Hanawa et al., 2021 :9720) et ce plus seulement en anglais. Enfin, des appels de projets² sont lancés pour promouvoir la recherche dans le domaine de l'acquisition des langues. Le monde multilingue que nous connaissons est en train de vivre une révolution numérique sans précédent et l'éducation est un secteur encore trop peu exploré alors qu'il est le centre névralgique de la communication langagière. Les ressources textuelles sont de plus en plus grandes et l'annotation de plus en plus fastidieuse. Comme la première section l'indique, il est nécessaire d'avoir accès à des données qualitatives et qualitativement renseignées, et ce, plus uniquement pour la recherche scientifique, mais aussi pour l'élaboration d'outils utiles au monde. Les outils, d'ailleurs, se développent de plus en plus dans un univers connecté et numérique. C'est pourquoi les prochaines sections sont dédiées aux méthodes et aux enjeux du TAL dans le domaine

² International Journal of Artificial Intelligence in Education: « Call for Papers: Educational NLP for a Multilingual World: Research, Applications, and Challenges », lancement décembre 2022.
<https://www.springer.com/journal/40593/updates/23181128> (visité le 20/01/2023)

de l'éducation et plus précisément dans la construction d'un corpus annoté en erreurs à des fins pédagogiques.

2.3 L'annotation automatique, approches techniques.

Depuis le début de cet exposé sur les enjeux qui animent la présente recherche, un point semble s'imposer comme central. Aussi bien dans l'éducation que dans les besoins de la recherche ou dans les avancées en termes de technologie, il faut des données sur lesquelles s'appuyer. Dans le cas des erreurs d'apprenants il faut avant tout des textes écrits par le public cible. Mais les textes bruts ne sont évidemment pas suffisants. Pour analyser, exploiter, voire corriger les erreurs ou encore guider les apprenants dans leurs apprentissages, il faut trouver les erreurs en question. Une tâche linguistiquement pas si simple comme les premiers chapitres de cette recherche l'ont démontré. L'annotation des textes bruts est donc une étape indispensable. Des questions se posent alors : comment annoter et plus largement qu'est-ce que l'annotation, quelles en sont les méthodes et techniques. Le but de cette section est entre autres de répondre à ces questionnements. Dans un premier temps, l'attention sera portée sur l'annotation au sens large et à la détection des erreurs, que ce soit pour une analyse ou de la correction. Puis, dans un second temps, je m'attellerai à décrire les outils qui permettent d'assister l'humain dans cette mission fastidieuse. En effet, comme l'humain a toujours cherché à compenser son manque de rapidité par les machines et l'automatisation des tâches, les outils d'annotation automatiques virent le jour bien avant les avancées techniques de ces dernières années. Et pour finir, le point le plus important de cette section sera de rendre compte des méthodes actuelles, qui permettent d'atteindre les résultats les plus prometteurs grâce aux avancées en termes de structures à réseaux de neurones depuis la fin des années 2010.

2.3.1 L'annotation (détection) d'erreur, implications et ressources.

Comme démontré précédemment, les difficultés de la langue écrite, en français, mais pas uniquement, sont une source de failles que l'être humain n'a de cesse de combler par la pédagogie, mais également par des outils. L'art de trouver les erreurs dans un texte est souvent mis au service d'une autre approche sensiblement moins pédagogique, la correction et son pendant technologique, la correction automatique. Selon James (1998 :235), les travaux d'analyse d'erreurs ainsi que les corpus annotés servent entre autres à guider la correction des erreurs. Toujours selon lui, la correction peut être explicite et externe, équivalente au feedback explicite, c'est-à-dire qu'elle présente directement la forme normalisée. Ou elle peut être implicite, c'est-à-dire qu'elle sert d'annonce pour une correction autonome (James, 1998 :236-

237). Pour les utilisateurs lambdas, la correction des erreurs est l'implication TAL la plus présente et l'une probablement des plus utilisées en tant que telle. En effet, quel simple outil de traitement de texte n'est pas pourvu d'un outil qui détecte les erreurs et propose la version « correcte » la plus probable.

Outre cette utilité flagrante et ce besoin de palier à l'erreur humaine, les experts de la langue argumentent en faveur de l'annotation et du feedback plutôt que de la correction « bête et méchante ». Il est, d'un point de vue évolutif, plus intéressant de prévenir les erreurs pour qu'elles ne se reproduisent plus, obligeant ainsi l'auteur de la faute à comprendre véritablement l'origine du problème (James, 1998 :237-240). De plus, dans le TAL aussi, des chercheurs ont démontré qu'un outil qui détecterait les erreurs et les classerait selon des catégories bien précises apporterait des informations nécessaires à l'amélioration des outils classiques de correction d'erreurs (Bryant et al., 2017 :800-801). Si l'on considère qu'il faille « nourrir » les intelligences artificielles de données pour qu'elles apprennent à corriger, au plus les données sont renseignées et commentées, au mieux se porteront les résultats de ces IA.

De manière générale, l'étape d'annotation automatique est précédée par une étape d'annotation manuelle très codifiée. Les projets d'annotation entièrement manuels existent aussi, souvent dans l'idée de créer des corpus d'erreurs utilisables, mais elles tendent à disparaître au profit de petites campagnes d'annotation qui permettent de « nourrir » les outils de classification automatiques (Mitton, 1996 :41-43). En effet, l'annotation manuelle à grande échelle amène son lot de contraintes. Annoter de grandes quantités de données est une tâche extrêmement chronophage et coûteuse (Antoine et al., 2019 :4). Elle peut être réalisée par des experts de la langue ou sous forme de crowdsourcing (Tetreault et al., 2010). Ou encore, elles prennent la forme de campagnes à grande échelle faisant appel à un grand nombre d'annotateurs (experts ou non).

Dans tous les cas, de telles campagnes nécessitent l'élaboration d'un guide contenant toutes les balises (tags) possibles et les utilisations de ces dernières (Voutilainen, 1999). De plus, gérer le caractère subjectif des erreurs et bien souvent les catégories vagues ou difficiles à manier avec une quantité conséquente d'annotateur ne permet pas d'atteindre des *accords inter-annotateurs* [1] satisfaisants (Dagneaux et al., 1998 :164-165). Les méthodes TAL promettent donc des campagnes d'annotation réduites doublées d'une rapidité et d'une fiabilité d'annotation accrue (Dagneaux et al., 1998 :166). Dans l'ordre, les étapes d'élaboration d'un annotateur automatique sont :

- La définition des types d'erreurs,
- Sur base de la typologie, la définition des balises (*tag set*),
- Créer un guide d'annotation pour la campagne,
- Annoter manuellement une partie des données,
- Implémenter l'annoteur sur base des données et du *tag set*,
- Phases de test.

(Lüdeling and Hirschmann, 2015 :136-140)

Une fois la typologie choisie comme décrit plus haut, il est aisé d'en dériver une série de balises qui correspondent aux catégories d'erreurs. En fonction de la précision de la typologie, les balises seront plus ou moins précises et engloberont plus ou moins d'erreurs. En 2003 (:2-3), Granger propose une version « bêta » de son guide d'annotation de 2022 qui exploite les tags sur deux niveaux de description. Un niveau qui concerne la catégorie d'erreurs (ex. Erreurs morphologiques) et un niveau qui détaille l'entièreté des erreurs possibles (annexe 4). D'autres outils d'annotations sont moins précis et ne permettent que d'identifier où se trouve l'erreur. Dans ces cas-là, les tags sont binaires, *I* pour incorrect et *C* pour correct. C'est le cas entre autres de l'annotation de la version *Cambridge learner corpus* annoté pour la détection d'erreurs (Yannakoudakis et al. 2011). Un autre exemple de corpus annoté pour la détection et le diagnostic d'erreurs est le *FALKO corpus* (Lüdeling et al., 2005) qui lui est annoté selon différentes hypothèses et besoins. Il adopte une annotation des erreurs sous forme de différents niveaux linguistiques ainsi qu'une annotation morphosyntaxique indépendante.

En somme, les « tag sets » ou balises sont un pan important de l'élaboration d'un annoteur automatique. Dans la section suivante, les annoteurs automatiques, ou non, seront décrits en détail avec une description des tag sets et des méthodes utilisées à travers le temps.

2.3.2 Approche chronologique (et non exhaustive) des outils d'annotation automatique.

L'utilisation de l'expertise humaine est évidemment la première intuition pour la constitution de corpus annotés en erreurs. L'annotation manuelle se place alors avant les avancées techniques permettant une automatisation du processus, mais elle est également privilégiée encore aujourd'hui dans des analyses purement linguistiques et pédagogiques. C'est le cas par exemple des travaux d'analyse d'erreurs sur peu de données comme celui de Satake en 2020.

Bien que récent, cette étude des erreurs dans des textes d'apprenants et l'influence d'un corpus annoté dans la correction, se base sur des annotations manuelles et non une approche automatique. Ici, le choix des méthodes manuelles s'explique principalement par l'objectif principal de l'étude qui se concentre sur l'influence des corpus dans l'apprentissage d'une langue seconde. L'expertise humaine prévaut dès lors sur les qualités de rapidité, de quantité et d'économie des approches automatiques.

Plus proche du travail présenté dans ce mémoire, l'étude menée en 2010 par Rozovskaya et Roth propose un double corpus annoté et normalisé. Datée de 2010, l'approche de l'annotation y est principalement manuelle bien qu'elle soit assistée par un programme facilitant le placement et le choix des différents tags. Pour constituer ce double corpus, les données ont été sélectionnées depuis deux grands corpus d'apprenants puis par une phase de sélection par *4grams* pour limiter les différences de niveau flagrantes. L'annotation manuelle est donc faite par trois experts (étudiants) de la langue anglaise et natif de cette dernière. Le choix de la typologie est plus simplifié que celles présentées par Granger (2003) ou par Câslaru (2016). Elle ne comprend que sept catégories d'erreurs décrites dans le tableau 1.

Type d'erreur	Description
Erreur d'article	Toute erreur impliquant un article
Erreur de préposition	Toute erreur impliquant une préposition
Nombre	Erreurs impliquant une confusion pluriel/singulier d'un nom
Forme verbale	Erreurs sur le temps et les flexions des verbes
Forme du mot	Lexème correct, mais suffixe erroné
Orthographe	Erreur d'orthographe
Insertion, commission ou remplacement	Autres corrections n'entrant dans aucune autre catégories

(Tableau 1) catégories d'erreurs trouvées chez Rozovskaya et Roth (2010:32).

L'intérêt principal de cette étude, bien que la typologie soit simpliste et peu justifiée, est le calcul de l'accord inter-annotateurs. Il est aisé d'imaginer que trois natifs, experts de la langue, arrivent à un taux d'accord relativement élevé. Or les résultats des calculs d'accord de cette étude sont bas, ce qui selon Rozovskaya et Roth (2010) s'explique par la complexité de la tâche et la variabilité d'interprétation des différentes balises. L'accord varie alors entre 56 % à 78 % avec un Kappa allant de 0,16 à 0,40.

Pour pallier ce type d'écarts flagrants, Tetreault et al. en 2010, (toujours) proposent une approche innovante pour l'annotation manuelle d'erreurs. Ils passent par les outils du net, mais cette fois-ci pas uniquement par une interface d'annotation, mais bien par la main d'œuvre trouvable sur internet. The Amazon Mechanical Truk permet de faire les tâches d'annotation via le *crowdsourcing* (ou la production participative en français). Ici l'accord s'améliore légèrement (un kappa moyen de 0,6) avec une réduction considérable du temps et des coûts d'une campagne d'annotation manuelle.

Ces avancées pour simplifier la vie des annotateurs et améliorer les campagnes sont efficaces, mais n'égaleront jamais les résultats que l'on atteint avec des approches technologiquement plus puissantes. Le machine learning (ou apprentissage automatique en français) s'est imposé depuis le début des années 2000 comme le nouveau standard en matière de performance dans le monde de la linguistique computationnelle. La tâche d'annotation des erreurs se rapproche en outre d'une tâche de classification des tokens [25] (unités de langage) selon les tags (classes, balises) définis au préalable.

Même si ces approches deviennent obsolètes avec l'arrivée des *réseaux de neurones* [20], en 2017, Zhao et al. prennent le parti d'annoter les erreurs grammaticales du chinois avec une approche par *n-grams* [14]. Cette approche est tout particulièrement adaptée aux spécificités du chinois, car elle permet de combiner une approche par caractères (avec un modèle de lettres, prédiction des suites de lettres), par mots (avec un modèle de mots, prédiction des suites de mots), un modèle *bi-gram* et un modèle *tri-gram* qui combinent les deux modèles de mots et de lettres. Ces modèles permettent de détecter si les mots ou séquences ainsi composés existent dans un dictionnaire de formes correctes, le cas échéant, détecter l'erreur dans le texte. Mais il est clair que dans l'optique d'annoter l'erreur, ce modèle ne permet que de déterminer deux classes, correcte et incorrecte. C'est encore bien loin de l'idée de diagnostiquer l'erreur commise tout en étant prometteur pour des outils de normalisations.

Une autre approche pour détecter les formes erronées dans un texte sont les grammaires de précisions (Wagner et Foster, 2007). Dans cette étude de 2007, les deux approches citées plus haut sont combinées à celle des arbres de décision pour repérer les formes incorrectes dans un corpus de texte. Les formes détectées sont malheureusement restreintes (9) et toujours aucun diagnostic n'est posé après la détection.

9. Les mots manquants, les mots superflus, les erreurs d'orthographe résultant en un mot existant (calculé par la distance d'édition) et les erreurs d'accord.

(Wagner et Foster, 2007)

Retour sur les erreurs du chinois, Xiang et al. (2015) propose une méthode de détection et de diagnostic des erreurs utilisant l'*ensemble learning* [7]. Le but est de détecter les erreurs et d'indiquer le type d'erreurs selon une typologie précise. Différentes méthodes de *machine learning supervisés* [3] sont testées pour la classification et la détection des erreurs (*Naive Bayes*, *arbre de décision*, *SVM*, *MLE*) ainsi que des méthodes d'ensemble comme l'*ensemble learners Adboost (AB)*, *Random Forest* [18] (*RF*) et *Random Feature Subspace (RFS)*. Ici, comme l'objectif principal est le diagnostic, la tâche se rapproche en effet de la classification. Les erreurs sont classées en quatre types en plus de la balise « correct » : l'ordre, la redondance, l'omission et les erreurs de choix de mots. L'étude montre des résultats satisfaisant avec les méthodes d'ensemble, surtout l'*ensemble learner Adboost* avec une mesure *F1* [29] pour la détection de 0,6347 et pour la classification de 0,5289. Avec une sélection de features spécifique (*POS* [16] *Trigram and probability*) et le modèle RBF les résultats de la détection sont encore plus prometteurs *F1* de 0,6648 (Xiang et al., 2015 :102).

D'autres approches, moins mathématiquement sophistiquées, permettent aussi de classifier les erreurs automatiquement. C'est le cas du programme *ERRANT* introduit par Bryant et al. en 2017. L'approche automatique est globalement une approche par règles permettant d'assigner 25 catégories d'erreurs (annexe 5). L'objectif étant d'ajouter des données au programme de correction automatique pour en améliorer les performances et en faciliter l'évaluation. L'avantage principal de cette approche est qu'elle ne nécessite pas de données prédéfinies pour l'entraînement, ce qui évite le problème de quantité de données et de biais qu'ont les approches de machine learning dû à des données trop peu diversifiées (Bryant et al, 2017 :3). Le programme comporte donc 50 règles, ce qui rend cette méthode entièrement déterministe et par conséquent plus stable dans les prédictions que les méthodes de machine learning. Néanmoins, les résultats prometteurs (95% des erreurs classifiées sont considérées comme des classifications « bonnes » ou « acceptables » par les vérificateurs) ne permettent pas de couvrir un nombre illimité de cas et les règles devraient être revues et corrigées selon le public ou la langue cible.

Comme je l'ai mentionné plus haut, l'état de l'art que représentait le machine learning jusqu'au début des années 2010 est désormais supplanté par une avancée majeure dans le domaine de la technologie, le deep-learning ou apprentissage profond en français. Depuis

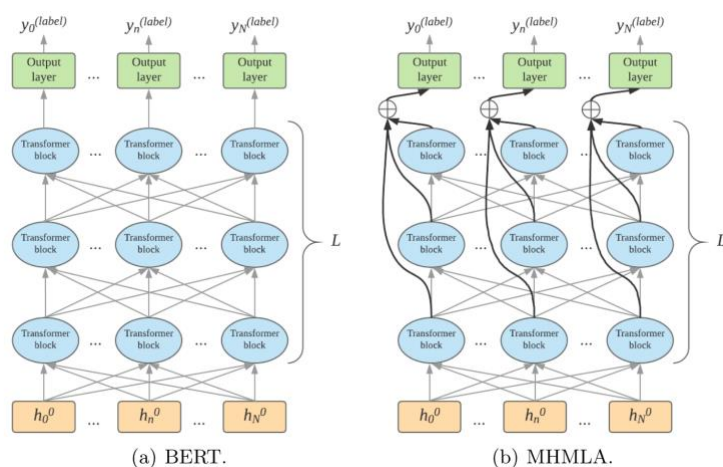
2016, des équipes de recherche se sont lancées dans la conquête du deep learning pour des tâches d'annotation et de classification des erreurs d'apprenants. En 2016 donc, Rei et Yannakoudakis proposent une architecture à réseaux de neurones pour résoudre l'annotation des erreurs en une *sequence labelling task* [22]. Le modèle reçoit une séquence de token en input et produit en output la probabilité que la séquence soit correcte ou incorrecte selon un contexte donné. L'étude teste six architectures de réseaux de neurones différentes : *convolutionnel*, *réseaux récurrents bidirectionnel*, *LSTM bidirectionnel* et une version multicouche de chaque. Les expérimentations sont faites à partir du *FCE-public* corpus (Yannakoudakis et al. 2011) annoté manuellement avec une typologie de 70 types d'erreurs. La détection se fait alors au niveau du token, c'est-à-dire que chaque token de la séquence d'erreur est annoté comme incorrect. Les résultats de la version simple du *Bi-LSTM* [5] sont ceux avec la mesure F0,5 la plus prometteuse de toutes les architectures testées, 41,1 avec le corpus de test (Rei et Yannakoudakis, 2016 :1185).

En 2017, Kaneko et al. se basent sur les résultats prometteurs des *Bi-LSTM* pour proposer une amélioration éventuelle. Toujours sur les mêmes données, ils proposent d'incorporer aux architectures *Bi-LSTM* des *plongements lexicaux* [15] (*word embeddings*) spécifiques aux erreurs et à la grammaticalité (*E&GWE*) plutôt que les plongements classiques (tel que *W2V**) (Kaneko et al., 2017 :45). Les résultats de cette version en comparaison de celle de Rei et Yannakoudakis (2016) sont sensiblement plus prometteurs avec un F0,5 de 41,4 et de 44,06 sur un corpus alimenté plus largement par le *Lang-8* corpus. Ces résultats sont satisfaisants, mais depuis 2017, le monde du TAL connaît une énième révolution. Les *architectures Transformers* [6] tendent à devenir le nouveau standard, et ce, y compris pour la classification et la détection des erreurs. Plusieurs études utilisent donc « *Bidirectional Encoder Representation from Transformers* » ou BERT pour les intimes, pour surpasser les résultats obtenus avec d'autres architectures telles que les *Bi-LSTM*.

Bell et al. (2019) proposent d'augmenter les approches *Bi-LSTM* qui font état de l'art (dans la même idée que Kaneko et al., 2017) avec des plongements lexicaux tels que BERT, ELMo et Flair où seul BERT agit avec des architectures Transformers. La base du modèle est donc un annotateur séquentiel *Bi-LSTM* auquel est ajoutée une couche d'*embedding* pré-entraîné. Sans surprises, sur le corpus *FCE* utilisé plus haut, la version incorporant le BERT *embedding* atteint les résultats les plus prometteurs avec une mesure F0,5 de 57,28, ce qui confirme la supériorité des architectures Transformers.

Toujours en 2019, Kaneko et Komachi espèrent encore améliorer les résultats en utilisant BERT comme architecture de détection des erreurs grammaticales. Ils proposent en réalité un « multihead multi-layer attention, MHMLA » sur chaque couche pour *fine-tuner* [8] BERT

(adapter l'IA à la tâche de classification) (fig. 1). Ainsi, l'idée est d'atteindre une représentation spécifique à la tâche de détection en appliquant à chaque sortie de couche du bloc Transformers une couche d'*attention* [13].



(Fig. 1) Architectures de BERT seul et de MHMLA sur BERT pour la détection d'erreurs grammaticales (Kaneko et Komachi, 2019 :4).

Pour plus de clarté, je présenterai ici les résultats basés sur le même corpus que celui utilisé dans les autres études (*FCE* corpus). Pour ce qui est des résultats, la baseline (BERT seul pré-entraîné) atteint déjà des résultats plus que satisfaisants avec un F0,5 de 59,47. Or, le modèle présenté dans l'étude de Kaneko et Komachi atteint des résultats encore plus impressionnants pour un F0,5 de 61,65.

Bien que prometteuses, ces approches ne permettent pas (en l'état) de catégoriser les erreurs. En effet, la plupart des approches de deep learning présentées se contentent de détecter les erreurs et de les classer selon deux tags « correct » et « incorrect ». Cette vision est intéressante, certes, mais comme je l'ai démontré, l'utilité du diagnostic et de la classification des erreurs selon leurs types n'est plus à prouver, tant au niveau pédagogique que pour la recherche linguistique. Même dans les tâches de détection pure, la classification et le type d'erreurs pourraient être bénéfiques. Un objectif annexe qui permettrait effectivement d'améliorer les résultats des modèles neuronaux de détection d'erreurs grammaticales et orthographiques (Rei et Yannakoudakis, 2017 :36). De plus, ces objectifs annexes peuvent donner lieu à d'autres études (à différents niveaux) ou à des outils plus complets que la simple détection (feedback ou correction pédagogique par exemple).

Les autres limitations des modèles présentés plus haut sont les données, bien souvent trop limitées, et l'accès à de telles méthodes pour d'autres langues que l'anglais. Le français a

évidemment ses problématiques propres et l'adaptation des modèles à ces dernières n'est pas toujours chose aisée.

Encore plus récemment, en 2023, une étude basée sur les erreurs dans des rédactions universitaires en français a été publiée (Noreskal et al., 2023). Cette étude est intéressante pour deux raisons : d'une part, il s'agit de l'une des rares faisant appel à des réseaux de neurones pour des corpus francophones et d'autre part, car la détection des erreurs, bien que plus spécifique que celles présentées dans ce mémoire, reste une détection et donc une classification multi-classe au lieu d'une classification binaire. Cependant, le corpus de base diffère catégoriquement de celui qui anime le présent projet. En effet, il consiste en un ensemble de textes universitaires rédigés par des étudiants natifs du français et de plus les erreurs analysées ne concernent que les erreurs commises dans des cas de phrases coordonnées. Malgré le caractère très spécifique, les erreurs ont été traitées via une typologie d'erreurs (5 catégories pour 8 types d'erreurs). La constitution du corpus annoté est par ailleurs passée par la phase de campagne d'annotation, dans ce cas-ci, le corpus étant restreint, seuls deux annotateurs ont été mobilisés. Pour la détection automatique, deux expériences ont été menées, à l'image de la littérature : une expérience concernait la détection binaire (correct/incorrect) et la seconde une détection multi-classes. Contre toute attente, les *Baselines* [4] (apprentissage de surface) utilisées par les chercheurs sont une classification réalisée grâce à une *machine à vecteur de support (SVM)* [23] et une *Random Forest*. Pour la phase d'apprentissage profond, les chercheurs ont utilisé deux versions francophones de BERT : CamemBERT et FloBERT, toutes les deux entraînées sur des grandes quantités de données francophones et finetunées sur le corpus d'écrits universitaires annotés. En somme, pour la détection binaire, les algorithmes d'apprentissage profond obtiennent des F-mesures supérieures à 0,71 avec un pic à 0,77 pour l'expérience de CamemBERT entraîné avec un corpus déséquilibré. Mais l'étude a obtenu des résultats quelque peu décevants pour la détection multi-classe avec une valeur de *précision* [17] trop basse et des erreurs d'étiquetage.

Ces nouvelles approches de l'intelligence artificielle, calquées sur le fonctionnement humain, pourraient, à l'avenir, devenir suffisamment fiables pour que les acteurs les plus réticents du monde de la linguistique et de l'enseignement commencent à y prêter réellement attention. Mais il est clair que pour arriver à cet idéal, les modèles devront encore pallier leurs limites. Entre autres, les approches automatiques devront considérer des données plus diversifiées et représentatives, être de plus en plus explicables pour permettre aux novices de leur faire confiance, mais aussi, elles devront aussi s'adapter à la demande et aux besoins des utilisateurs de la langue. Le constat actuel est tel que les ressources viennent à manquer et que les outils ne sont pas adaptés à la demande en diagnostic

d'erreur du monde de la pédagogie. Ces limitations seront discutées dans la section suivante. Sur base des outils et des avancées présentés dans la section précédente, le projet présenté ici essayera aussi, humblement, de fournir une ressource pédagogique/ théorique ainsi que les outils nécessaires pour sa création et son analyse.

2.4 Spécificité liée à la langue et au public cibles.

La quasi-totalité des outils présentés dans la section précédente traitent des erreurs des apprenants de l'anglais (*ESL* ou « English as a second language ») et, pour certains, des erreurs commises par les natifs de la langue. L'objectif de la présente étude est alors d'étendre cette connaissance à d'autres langues et en l'occurrence au français et à ses apprenants. Le début de cet exposé s'est attelé à comprendre et théoriser la complexité du français et à rendre compte de la nécessité d'adapter les avancées technologiques au monde de l'éducation (du FLE). Quiconque voulant étendre ces connaissances et ces modèles à d'autres langues que l'anglais se heurtera alors à deux problèmes majeurs. Le premier, l'accès aux données (et leur traitement) est d'autant plus flagrant si l'on veut traiter non pas la langue des natifs, mais bien celle des apprenants. Pour le second, les outils eux-mêmes doivent être adaptés aux spécificités propres aux langues étudiées. Mais ces derniers posent encore un autre débat non négligeable, celui de l'évaluation. Car il est nécessaire, une fois l'outil créé, d'évaluer son efficacité avant d'oser le proposer à des fins académiques, de recherche ou pédagogiques.

2.4.1 Le problème des données.

Un rapide coup d'œil sur le site du CECL (*center for english corpus linguistics de l'UCLouvain*) nous permet de nous rendre compte du manque criant de corpus d'apprenant (ou *learner corpora* en anglais) et ce, surtout pour le français. Or, les études ont démontré que la taille et le nombre de corpus nuisait aux résultats des outils de détection et de correction des erreurs. De fait, un corpus restreint ou trop peu varié, offre trop peu de variation dans les erreurs lors de l'entraînement et par conséquent une qualité de détection et de classification moindre (Mizumoto et al., 2012 :870). Outre l'accès à une quantité suffisante de données, il faut également s'assurer que les langues maternelles des apprenants soient suffisamment variées pour être sûr que l'outil de détection appréhende correctement les erreurs interférentielles, c'est-à-dire les erreurs où la langue maternelle interfère dans la structure de la seconde langue (Bechichi, 2017).

Non seulement ces données doivent être suffisamment nombreuses et variées, elles doivent bien souvent être annotées en erreur (peu importe la précision de l’annotation). Il est clair que les outils présentés précédemment sont encore des outils nécessitant un *apprentissage supervisé**, ne serait-ce que dans l’étape du *fine-tuning*. Une condition s’ajoute alors à l’obtention d’un corpus utilisable, son annotation manuelle, et de ce fait, une main d’œuvre disponible à cet effet. Le crowdsourcing est, on l’a vu, une première solution aux campagnes d’annotations coûteuses, mais il peut être également une bonne source de produit brut volontaire. C’est le cas du corpus *Lang-8* issu d’une campagne participative où les utilisateurs sont encouragés à écrire un texte dans leur seconde langue, pour recevoir un feedback de natifs sur leurs écrits, mais surtout pour nourrir une base de données³. Le problème majeur de cette approche est le manque de contrôle sur le niveau d’expertise des annotateurs ainsi que sur les métadonnées disponibles lors de la création de tels corpus, mais aussi, un grand nombre de participants implique un grand nombre de données non exploitables (Bryant et al., 2022 :23).

Une alternative pour avoir des corpus d’erreurs représentatifs est la création de corpus d’erreurs artificiellement générées (Felice et Yuan, 2014). Cette approche a le vent en poupe depuis moins d’une dizaine d’années et est déjà en train de devenir le standard pour l’entraînement des outils de correction automatique (*GEC, grammatical error correction*). Très efficace notamment, car le concept est de partir d’un texte correctement rédigé pour le « brouter » généralement par des approches par règles, afin de générer le texte correspondant contenant des erreurs. Cette génération permet ainsi d’éviter d’avoir besoin de main d’œuvre humaine pour mettre en parallèle le texte correct et l’erreur (Bryant et al., 2022 :24-25). Il faut tout de même garder à l’esprit qu’à l’heure actuelle cette méthode reste peu optimale surtout quand il faut créer des erreurs typiques d’apprenants. Les travaux d’analyse d’erreurs et les corpus d’erreurs existants sont là aussi nécessaires pour diagnostiquer et ensuite générer des erreurs artificiellement.

Néanmoins, une fois le problème des données et de l’annotation réglé, il faut encore bénéficier d’un outil efficace pour la tâche donnée.

³ Initié par Mizumoto en 2011, le corpus *Lang-8* est une base de données multilingues issue d’un outil en ligne d’entraide entre des natifs et des apprenants d’une langue. (Cf. Mizumoto et al., 2012)

2.4.2 Le problème des outils.

Un certain nombre de travaux se sont déjà essayés à l'adaptation de modèles existant pour l'anglais vers d'autres langues moins représentées et moins documentées. C'est le cas, par exemple, de l'adaptation du projet de détection *ERRANT* (Bryant et al., 2017) en une version adaptée pour le grec nommée *ELERRANT* (Korre et al., 2021). Étant donné, la nature de l'outil d'origine (approche par règles) l'adaptation à une autre langue est assez aisée. Il « suffit » d'adapter les règles préexistantes à la langue et au public cible. Toujours est-il que cette version garde les limitations de son programme citées plus haut.

D'autres approches tentent de créer des outils directement désignés pour de l'annotation multilingue, mais, par la force des choses, ces méthodes sont surtout développées dans le cadre de la correction d'erreurs grammaticales (*GEC*) (Rothe et al., 2021). De fait, l'offre et la demande (chez Google pour ne citer qu'un seul exemple) est bien moindre du côté pédagogique et analytique de l'annotation et du diagnostic des erreurs.

Néanmoins, des projets d'outils construits directement sur une langue autre que l'anglais et se basant sur les avancées permises par la richesse des données en anglais existent. Et ce même pour des langues syntaxiquement et morphologiquement très éloignées, comme l'arabe (Belkebir et Habash, 2021). Le projet *ARETA* présenté en 2021 traite des spécificités de l'arabe et des erreurs propres aux apprenants de cette langue. Il se base aussi sur une dérivation d'*ERRANT* travaillant ainsi avec des corpus parallèles de textes normalisés et erronés et produisant une détection et une classification des erreurs propres à la langue (Belkebir et Habash, 2021 :509-600). Des projets comme celui-ci sont la preuve que l'annotation des erreurs des apprenants, bien que peu aisée, a de l'avenir aussi bien dans la recherche que dans la pédagogie.

Un autre problème majeur concernant les outils d'annotation est leur méthode d'évaluation. En effet, l'intuition première serait d'évaluer la qualité de détection et d'annotation des erreurs avec l'expertise humaine. Or sur de grands corpus, l'utilisation de l'expertise humaine serait longue, coûteuse, voire, inenvisageable (Tetreault et al., 2014 :6-7). L'intérêt est alors d'utiliser l'expertise humaine pour créer le *gold standard* [10] qui permettra ensuite d'évaluer l'annotation proposée par le système (Tetreault et al., 2014 :9). Néanmoins, même avec une comparaison à un gold-standard, plusieurs personnes sont nécessaires pour évaluer l'exactitude d'une annotation. Pour ce qui est de l'évaluation quantitative, une chose est sûre, on observe peu de consensus au sujet des métriques utilisées pour analyser la qualité de la détection ou de l'annotation. Ce manque d'unicité se traduit alors par une comparaison

difficile entre les différents outils (Chodorow et al., 2012 :618). Le choix de la métrique sera alors décisif pour la présentation et la fiabilité des résultats.

En somme, l'étude de la complexité de l'orthographe française a permis de mettre en lumière différents points importants pour le présent projet. Dans un premier temps, l'impact social de l'orthographe et sa nature bien souvent discriminante démontrent la nécessité d'améliorer d'une part la pédagogie de l'orthographe en elle-même et d'autre part, ils démontrent la nécessité d'adapter et d'améliorer les outils nécessaires à une bonne acquisition de ce pan de la langue. Le concept de « bonne » acquisition réside avant tout dans une dédramatisation des erreurs, qui, on l'a vu, sont intrinsèques à l'histoire et au fonctionnement de la langue écrite. Mais aussi dans une approche active de la correction desdites erreurs, permettant aussi de les transformer en outil pédagogique. Dans un second temps, l'étude des spécificités du français et de ces différentes normes permet d'y voir plus clair autour du concept même de l'erreur. Une fois ces bases posées, l'analyse des textes d'apprenants permet alors de trier, classer ces « erreurs » afin de les rendre exploitables. C'est dans cette optique bien précise que le questionnement sur les outils technologiques et le TAL entre alors en jeu. Il est clair que la vaste majorité des travaux d'analyse d'erreurs (en tous genres et pour tous types d'objectifs) est encore réalisée à la sueur du front des experts de la langue. Or une automatisation de ces tâches (d'annotation, d'analyse, de diagnostic et de feedback) permettrait entre autres de sauver du temps et de l'argent pour les recherches dans ce domaine. Et il est clair que, comme dans tous les domaines, les avancées technologiques riment bien souvent avec découvertes et démocratisation de ces découvertes. Cette automatisation contribuerait également à créer des outils pédagogiques accessibles et utiles aussi bien pour les enseignants en classe de FLE que pour les étudiants eux-mêmes en recherche d'autonomie. La triste pandémie faisant rage depuis 2019 est d'ailleurs en partie responsable du regain d'intérêts pour l'apprentissage général comme pour celui des langues assistées par ordinateurs (ALAO). C'est donc bien dans ce contexte que le projet présenté dans la suite s'inscrit, reposant alors sur l'expertise de ses pairs pour proposer un outil d'annotation spécifique au français langue étrangère.

Chapitre 3 : Méthodologie.

3. Méthodologie :

Il est clair qu'après les constats de l'analyse de la théorie, il y a dans le domaine du FLE encore de la place pour les avancées et la recherche. Il semble nécessaire, dans l'état actuel des choses, de créer une ressource utile et des outils efficaces pour créer et traiter cette ressource. Néanmoins, ce type de projet s'apparente à la problématique du serpent qui se mange la queue. On l'a vu, la plupart des outils technologiques fonctionnent encore sous le système de l'apprentissage supervisé. C'est-à-dire, qu'ils ont besoin d'un guide, d'un exemple, pour pouvoir produire des résultats proches de ceux de l'homme et pouvoir être appliqués à des données non vues. Dans le cas de l'annotation, il faut donc fournir à l'outil des données correctement annotées (par l'homme), en quantité suffisante, pour lui permettre de reproduire l'annotation au reste des données. Or, le constat linguistique est tel qu'il y a un manque flagrant de ressources annotées en erreurs, et ce, surtout en français.

Comment répondre aux besoins en données analysables par la recherche en linguistique et en pédagogie de manière efficace si les outils technologiques nécessitent eux-mêmes une telle ressource en quantité suffisante ? De plus, les outils technologiques, par le fait qu'ils permettent de faciliter grandement la tâche de création de ressources, sont utiles pédagogiquement. Une fois la brèche ouverte avec des algorithmes efficaces, il sera aisé de transformer un annotateur en quelque chose d'intéressant pédagogiquement (feedback automatique) ou en un correcteur automatique intelligent adapté aux erreurs et aux textes d'apprenants. Tous ces projets, aussi nécessaires soient-ils, ont besoin d'une phase de réflexion sur la mise en pratique, réflexion qui sera amorcée ici.

Il est, par ailleurs, dans l'intérêt scientifique de créer une ressource annotée en erreur de manière cadrée et réfléchie. Il me faudra documenter et justifier chaque étape de création mise en lumière par l'amorce théorique du chapitre précédent. Pour les parties plus techniques de linguistique computationnelle, la rigueur exige une documentation et une justification précises des méthodes suivies pour arriver aux résultats. C'est guidée par cet objectif de documentation et par la double quête que la théorie a mise en lumière que je vais, dans ce chapitre, m'atteler à la description de la méthodologie employée pour arriver aux résultats qui seront eux présentés dans le chapitre suivant.

Dans le cadre de mes études, il m'a d'ailleurs été demandé de faire un stage de sept semaines dans un laboratoire de recherche. Vu mon projet de mémoire, c'est tout naturellement que j'ai effectué ce stage au CENTAL (centre de traitement automatique du langage de l'UCLouvain). C'est durant ce stage que j'ai pu expérimenter et avancer sur la partie pratique de mon mémoire, partie qui va animer ce chapitre méthodologie et le suivant. Ce que j'exploiterai ici, sera entre autres le fruit d'un travail

conjoint réalisé avec mes maîtres de stage (le professeur Thomas François et le docteur Rodrigo Wilkens).

Dans un premier temps, je documenterai et expliquerai l'élaboration de la typologie d'erreur ainsi les choix en termes d'annotation manuelle et de création du gold standard. Les deux premiers points de ce chapitre seront donc consacrés aux réflexions linguistiques de la typologie d'erreurs des apprenants FLE (section 3.1) et de l'annotation humaine (point 3.2) qui prend la forme ici d'une campagne d'annotation.

Dans un second temps, chronologie oblige, une fois les ressources manuelles et les considérations linguistiques effectuées, il sera temps d'entrer dans la partie technique du projet. Les deux points qui suivront permettront donc d'expliquer et de justifier les choix des modèles et des algorithmes pour la tâche d'annotation. Puis, d'aborder concrètement les moyens mis en œuvre pour créer et entraîner les modèles, description brève des données, outil utilisé, etc. Et de présenter les méthodes et les métriques d'évaluation utilisées afin d'embrancher sur le chapitre « résultats » qui analysera et détaillera les résultats obtenus par le biais des moyens méthodologiques présentés ici.

3.1 Typologie des erreurs.

Comme démontré, l'erreur d'orthographe et les normes d'orthographe tendent à faire débat dans les sphères linguistiques et dans les sphères sociales. Là où l'erreur se définit, par un écart vis-à-vis d'une norme, la norme reste mouvante ou tout du moins adaptable. Une solution est alors toute trouvée pour cadrer tant au niveau pédagogique qu'au niveau de la recherche les normes sur lesquelles on s'appuie pour déterminer l'erreur d'orthographe : la typologie d'erreurs d'orthographe (TEO) ou la taxonomie pour les plus anglicistes. Mais là encore, selon la finalité et les besoins, la typologie peut répondre à différents critères, différentes normes et décrire les erreurs rencontrées plus ou moins précisément. Avant d'avoir une ressource annotée en erreurs, il est donc nécessaire de passer par une phase de réflexion purement linguistique pour fixer le cadre de ce que l'on considère comme une erreur. Cette typologie une fois créée sera par ailleurs le point de départ de la suite de l'élaboration de la ressource.

Plusieurs typologies ont servi d'inspiration et de point de départ pour la création de la mienne. Je vais d'abord partir de ces dernières et en discerner les failles qui m'ont poussée à extrapoler ma version. Ensuite viendra la justification de certaines des prises de positions vis-à-vis de la norme ou devrais-je dire des normes au regard du public cible de l'étude et des objectifs. En fin, il sera temps de présenter la typologie telle qu'elle sera utilisée par la suite. Cette présentation sera justifiée et commentée d'un point de vue pédagogique (public cible), linguistique (rigueur scientifique) et technique (mise en pratique d'un point de vue automatique).

Les inspirations principales sont les typologies de Granger (2003) et de Căslaru (2016) toutes les deux adaptées aux apprenants, toutes les deux pour le français. Mais mes recherches ne se sont pas arrêtées là. Outre les typologies, j'ai aussi passé du temps à étudier les travaux d'analyse d'erreurs d'apprenants pour coller au plus près de la réalité dans ma propre typologie. Une fois mes recherches faites, mes maîtres de stages et moi-même avons eu plusieurs discussions pour sortir les éléments clés de la typologie que je présenterai ici.

Dans le cas de la typologie de Granger (2003), pour le français, les questionnements majeurs tournaient autour du caractère très, voire trop complet, de la classification. En effet, je me suis inspirée de son intéressante approche bidirectionnelle qui prend en compte à la fois des catégories linguistiques comme grandes classes et des fautes typiques des apprenants comme classification plus fine. Mais ces catégories linguistiques, au nombre de neuf, semblaient complexes à mettre en œuvre dans l'optique d'une campagne d'annotation (en vue d'un projet d'annotation automatique). Techniquement, des classes comme le registre ou le style sont difficilement analysables sans contexte. Ce sont des erreurs, certes, mais dans l'idée d'adapter la typologie à tous types d'apprenants (tous niveaux CECR confondus) il était compliqué d'évaluer un style (lourd / obscur) ou un registre (choix de syntaxe ou de lexique inapproprié) sans avoir d'information sur le niveau des apprenants. Les catégories d'erreurs relevant du discours ou de la pragmatique ne feront donc pas partie de la typologie finale. En effet, d'un point de vue linguistique, peut-on vraiment considérer les erreurs de pragmatiques ou de discours comme des erreurs d'orthographe ? Plus encore, pédagogiquement, ces erreurs sont-elles réellement propres aux apprenants, si l'on en croit les grilles d'évaluations du CECR, elles n'entrent en compte qu'à partir des niveaux C1 voir C2. Ce groupe de niveau « C » ne représente pas, on le verra, la majorité des apprenants ayant recours aux outils pédagogiques que nous tentons de développer ici ni ceux les plus représentés dans le corpus.

Le ressenti global au regard de cette typologie est qu'elle est certes très informative, mais qu'elle semble difficile à manipuler (surtout pour des annotateurs moins « experts » de la langue française »). Aussi, elle paraît difficilement exploitable pour en développer un outil d'annotation automatique à visée pédagogique. Les *erreurs de lexique*, par exemple, sont peu explicables et se confondent avec les *erreurs de grammaire*, ce qui rend opaque la décision de les scinder en deux catégories.

D'un extrême à l'autre, la typologie de Căslaru (2016) est fortement inspirée de celle de Granger, la complexité en moins. Seules trois catégories linguistiques sont exploitées : lexique, morphologie et syntaxe. Cette typologie couvre également moins d'erreurs, ce qui peut s'expliquer par le fait que la langue maternelle des apprenants (le roumain) soit commune dans l'étude qui a nécessité la typologie. Cette explication permet d'ailleurs à Căslaru d'ajouter un troisième niveau d'annotation

concernant le caractère inter-langagier ou intra-langagier de l'erreur. Ce dernier niveau est extrêmement intéressant d'un point de vue pédagogique, mais n'est malheureusement pas réalisable techniquement dans la présente étude, les langues maternelles étant variées et souvent pas référencées lors de l'annotation (manuelle et automatique). Bien que plus condensée, cette typologie est bien moins intuitive et moins informative.

Le prisme de Câslaru permettra de réduire la complexité de la typologie de Granger (entre autres les erreurs de lexique et de grammaire ou encore de registre et de style). Mais il faudra l'agrémenter et la compléter par les finesses de la typologie de Granger et par l'étude de travaux d'analyse d'erreurs d'apprenants. Ces travaux me permettent de me rapprocher des erreurs présentes dans les écrits d'apprenants avant même d'avoir accès aux corpus à annoter et sans devoir moi-même passer par ce type d'études excessivement chronophage.

Les erreurs dues aux phénomènes d'euphonie du français (élisions, contractions ...) sont, par exemple, mises en lumière par les travaux sur les apprenants du français dont la L1 (langue première/maternelle) est l'arabe, le perse ou l'espagnol (Bourray, 2017 / Viémond 2021). Ces dernières sont manquantes dans les typologies de Granger et Câslaru ou du moins, elles n'ont pas de catégories propres. Ces travaux m'ont d'ailleurs, permis de mettre en avant les erreurs dues aux verbes pronominaux et de les séparer des erreurs de flexion des pronoms (ces erreurs seront expliquées et exemplifiées plus loin).

D'autres questions ont été soulevées par mes maîtres de stage lors de la réalisation du guide qui permet aux annotateurs de prendre en main la typologie et d'annoter correctement. Bien que les points de vue à ce sujet soient les plus objectifs possible, il a fallu décider laquelle de la nouvelle ou de l'ancienne orthographe privilégier. D'un point de vue pragmatique, la nouvelle orthographe étant encouragée par les instances légiférantes dans l'enseignement des langues, il serait bon de trancher pour cette norme-ci. Mais il est clair que certains irréductibles gaulois continuent à enseigner l'ancienne orthographe, et ce, surtout dans les pays où la francophonie ne possède pas d'instance qui permet d'en dicter les lois. Nous avons donc décidé à l'image de nos gouvernements, de tolérer l'existence des deux normes. Nous avons aussi décidé d'utiliser la grammaire de Grevisse (le bon usage et le petit bon usage), pour trancher sur d'autres conventions et normes moins connues, même des natifs. Utiliser ce type d'ouvrage m'a permis de fixer/cadrer les erreurs de ponctuations inexistantes dans la typologie de Câslaru.

Il est temps, à présent, de passer en revue les catégories de la typologie en détail. Cette description me permettra de préciser les sources et de justifier les choix méthodologiques. J'accompagnerai pour cela mes explications d'exemples tirés des différentes sources citées plus haut et s'il le faut, de passages plus théoriques pour défendre certaines de mes prises de positions. J'en profiterai pour

exploiter les discussions enrichissantes que j'ai pu avoir avec mes maîtres de stage lors de nos réunions hebdomadaires.

Le tableau 2 représente dans un premier temps les deux niveaux de classifications choisis pour la typologie ainsi qu'un bref aperçu des erreurs en elles-mêmes.

Lexique	Morphologie	Syntaxe
Erreurs de casse	Erreurs du verbe	Erreurs de structuration, de cohérence
Erreurs de diacritiques	Erreurs de complémentation	Mot manquant
Erreur de choix de mots	Erreurs d'euphonie	Mot superflus
Erreurs de graphie et de forme	Erreurs de flexion	Erreurs de répétitions ou de redondance
		Ordre des mots erroné
		Erreurs de ponctuation

(Tableau 2) tableau récapitulatif de la typologie d'erreurs des apprenants FLE en deux niveaux.

Comme je le disais, la structure en trois catégories linguistiques est directement tirée de la typologie de Câslaru mais le contenu de la typologie en est moins étroitement lié et s'inspire de diverses sources.

Les erreurs de syntaxe concernent plus largement l'articulation des constituants de la phrase. La classe des mots manquants est tirée directement des typologies « de surface » comme celles de Burt et al. (1982 :150-160). Cette classe comme celle des mots superflus est assez explicite, elles concernent toutes les deux les syntaxes correctes à l'exception d'un mot ou d'un groupe manquant ou en trop. C'est-à-dire que l'erreur survient si en supprimant l'élément superflu ou en ajoutant l'élément manquant, la construction est correcte d'un point de vue syntaxique (10:11).

10. *Qui mélange *entre la modernité et le classique / Qui mélange la modernité et le classique* (mot superflu)
11. **j'ai pas / je n'ai pas [construction tolérée à l'oral mais incorrecte à l'écrit si on suit Le Bon Usage]* (mot manquant)

Légèrement plus complexes, les erreurs de répétitions ou de redondances concernent toutes les constructions dans lesquelles un élément n'est pas superflu (ne rend pas la construction agrammaticale) mais où il crée une sensation de lourdeur dans le texte. Ceci englobe par exemple la répétition de la conjonction « et » dans la coordination ou encore le phénomène de redoublement du

sujet qui est contraire à la valence verbale* (nombre d'actants ou de recevants acceptables par le verbe) française (12:13).

12. **le domaine social *et économique et politique / le domaine social ,économique et politique*

(par convention seuls les deux derniers éléments d'une énumération peuvent être joints par la conjonction « et »)

13. **l'homme elle va / l'homme va ou elle va*

(Construction plus complexe, le verbe « aller » n'accepte qu'un seul sujet)

Pour les erreurs de ponctuation, il a fallu dans un premier temps fixer les normes que bien souvent les natifs eux-mêmes tendent à oublier. Selon la grammaire de Grevisse (2008 :122).

« On ne ménage pas d'espace *avant* le point, la virgule, les points de suspension (voir N. B. ci-dessous), la parenthèse fermante, le crochet fermant ; — *après* la parenthèse ouvrante, le crochet ouvrant.

Il y a une espace *avant* les guillemets ouvrants, la parenthèse ouvrante, le crochet ouvrant, le tiret ; — *après* le point, la virgule, le point-virgule, le double point, les points d'interrogation et d'exclamation, les points de suspension, les guillemets fermants, la parenthèse fermante, le crochet fermant, le tiret.

Il y a une espace, mais non sécable (c'est-à-dire que cette espace ne peut coïncider avec le passage à la ligne) *avant* le point-virgule, le double point, les points d'interrogation et d'exclamation, les guillemets fermants, — *après* les guillemets ouvrants. — Autrement dit : une ligne ne peut commencer par un point-virgule, un double point, un point d'interrogation ou d'exclamation, des guillemets fermants, une parenthèse fermante, — ni se terminer par des guillemets ouvrants. ».

Ces erreurs concernent donc aussi bien les ponctuations manquantes, de la mauvaise gestion des espaces avant et après tous les signes de ponctuations (primaires et secondaires).

Enfin, hormis les erreurs d'ordre des mots qui sont, je pense, suffisamment explicites, la dernière classe d'erreurs syntaxiques concerne l'articulation des écrits du français. Les erreurs de cohérence interviennent lorsque l'apprenant commet un mauvais choix dans les connecteurs logiques (conjonction de coordination et de subordination) (14:15). Ce choix erroné provoque une erreur de cohérence dans la construction du texte au global, ce qui justifie le caractère syntaxique de l'erreur.

La difficulté pour moi dans ce cas-ci a été de réduire au maximum la classe pour ne pas tomber dans des erreurs de pragmatique et de sémantique qui, nous l'avons décidé, ne rentraient pas dans le cadre de cette typologie d'erreurs.

14. *ils sont petits *or forts/ ils sont petits mais forts*

(conjonction de coordination erronée)

15. **comme tu étais venu tu n'aurais pas raté la fête / si tu étais venu [...]*

(conjonction de subordination erronée)

Les catégories linguistiques « lexique » et « morphologie » sont par ailleurs toutes deux intrinsèquement liées au mot/ au token en lui-même. La majeure différence est que la catégorie « morphologique » comme son nom l'indique, n'englobe que les erreurs relevant de la morphologie (construction interne comme la flexion, la dérivation ...) (Chevrel 1973 :87) alors que la catégorie « lexicale » pourrait s'apparenter à l'idée que les linguistes se font de l'orthographe d'usage. C'est-à-dire la forme propre des éléments du lexique ainsi que des notions de choix de certains éléments (d'un point de vue lexico-sémantique).

Dans la catégorie des erreurs lexicales, on retrouve donc la vaste classe des erreurs de graphie et de forme. Celle-ci s'attaque directement aux erreurs qui concernent la forme physique des mots. Elle s'apparente à ce que le commun des mortels appelle l'orthographe qui s'apprend par l'usage ou par des listes de règles et d'exceptions à connaître par cœur. La classe des erreurs de graphie reprend donc tous les types de coquilles (oublis de lettre, inversion, substitution d'une lettre par une autre) (16) mais aussi les cas très fréquents chez les apprenants d'écriture phonétique (17) (Bourray, 2017 :170-171). Ces dernières sont, je l'accorde, quelque peu complexes à reconnaître, mais l'idée est qu'une fois le mot mal orthographié prononcé, on puisse reconnaître quel mot l'apprenant a voulu écrire. L'idée est que l'enchaînement de graphèmes est incorrect, pour autant, l'enchaînement de phonèmes, lui, donne lieu à un mot acceptable. Ces erreurs sont extrêmement courantes à cause du manque de correspondances entre les graphèmes et les phonèmes du français, un phonème peut avoir différentes formes écrites et un graphème, selon le contexte, peu avoir différentes formes orales également.

16. **ps* / *pas*, (omission d'une lettre)
 **qunad* / *quand* (inversion de l'ordre des lettres)
 **écomonie* / *économie* (idem)
17. **labirint* / *labyrinthe* (prononcé [*la.bi.ʁɛ̃t*] dans les deux cas)

Toujours dans cette catégorie linguistique, on retrouve les erreurs de casses et les erreurs de signes diacritiques. Les erreurs de casses regroupent toutes les majuscules mal placées ou oubliées alors que les erreurs de signes diacritiques concernent les erreurs de placement ou d'oublis des signes diacritiques (apostrophes, guillemets, trait d'union et accents) (18:19).

18. **portebagage* / *porte-bagage* (agglutination)
 **centre ville* / *centre-ville* (omission)
 **fôret* / *forêt* (*mauvais placement*)
19. **paris* / *Paris* (omission)

Pour finir avec cette catégorie lexicale, nous avons décidé de traiter deux cas d'erreurs dans le choix de mot. Lorsqu'une phrase contient un mot ou une construction erronée (mal choisie) et que cette construction existe en français, l'erreur de choix de mot relève donc d'un mot inapproprié dans le contexte (20). Alors que quand le mot mal choisi n'appartient pas au lexique du français (ce compris les néologismes et l'emprunt toléré en français) le choix de mot relève d'un mot non-standard (21). Cette distinction permettra par la suite, si la ressource est mise à la disposition de la communauté scientifique, d'analyser les emprunts aux langues premières ou à d'autres langues secondes par le biais d'analyse de la classe mot non-standard et des renseignements sur la L1 (ou les autres L2) fournis dans la ressource. Ces deux classes sont présentes dans la typologie de Câslaru mais la classe des mots inappropriés est aussi inspirée de la typologie de Granger avec les erreurs de « figement ». Ces erreurs concernent la violation des cooccurrences du français (expressions figées) faisant partie aussi du lexème du français. Quand la construction d'une expression lexicale est erronée, la construction est donc inappropriée au même titre que les tokens seuls. Dans les erreurs non-standard (résultant en une forme inacceptable dans le lexème francophone) sont classées les erreurs de dérivation, bien que relevant de la morphologie. Le procédé dérivation permet de créer de nouveaux mots du lexique, or la violation de ces procédés entraîne donc des mots non-standard dans le lexique, ce qui explique le classement dans la catégorie « lexique ».

20. *pêche / poisson

*cours religieux / cours de religion (erreur de cooccurrence)

Je *voudrais bien aller / je voudrais bien y aller

21. *window / fenêtre (emprunt anglais)

*évolutionner / évoluer (suffixe)

Enfin, les erreurs morphologiques se rapprochent en effet du concept d'orthographe grammaticale. On y retrouve les erreurs de flexion au nombre de 3, la flexion en genre propre au français, la flexion en nombre, mais aussi la flexion du pronom plus complexe que simplement en genre et nombre. Comme je l'ai mentionné plus haut, l'analyse d'erreurs m'a permis de distinguer des erreurs courantes chez les apprenants dans l'utilisation des pronoms. L'erreur se trouve aussi bien dans sa relation anaphorique (mauvais genre, mauvais nombre) mais aussi dans sa flexion grammaticale (vestige des déclinaisons en cas du latin) (22).

22. *Je *toi le donne / je te le donne (à ne pas confondre avec les verbes pronominaux)*

(erreur de « déclinaison »)

*Quand tu seras là, je veux *vous raconter / quand tu seras là, je veux te raconter.*

(erreur dans la relation anaphorique « accord en nombre »)

Aussi mises en avant par les travaux d'analyse d'erreurs, les erreurs d'euphonies sont inexistantes des typologies d'erreurs, or les phénomènes d'euphonie font partie intégrante de la langue et sont plus qu'intéressants à analyser du point de vue des apprenants. Quand on parle d'euphonie en français, cela regroupe tous les cas d'harmonisation de la prononciation par l'altération de la forme écrite (23). Encore une fois, avec mes maîtres de stage, nous avons trouvé judicieux d'ajouter cette dimension « inédite » à la typologie pour enrichir toujours plus la ressource qui en résultera.

23. *de le / du

*ce aspect / cet aspect

*la parterment (ici une erreur d'euphonie l' et non la, mais également une erreur au niveau du mot « parterment » qui est non standard)

Il ne reste maintenant que deux classes d'erreurs non explorées, par souci de place, vous trouverez la description détaillée des erreurs du verbe qui sont au nombre de six en annexe avec le guide d'annotation (Annexe 6). En effet, les erreurs qui concernent la conjugaison du français et plus largement les verbes sont tellement spécifiques que nous avons trouvé opportun de toutes les

regrouper dans une classe spécifique. Dans les autres typologies, ces erreurs étaient éparpillées dans les catégories « grammaire » et « morphologie » ce qui rendait l'analyse de telles erreurs complexes. Elles regroupent des erreurs de conjugaison, mais aussi des erreurs relevant des règles de concordance des temps très présente en français ou encore des erreurs autour des verbes pronominaux. Cette dernière sous classe de verbes est encore une fois trop peu étudiée, probablement par manque de ressources. Il est alors intéressant de considérer les verbes pronominaux (les verbes et leurs pronoms) comme une classe à part entière. De plus, au regard des programmes d'apprentissage et des attentes des niveaux CECR la maîtrise de ces verbes est importante.

Le dernier type d'erreurs est directement tiré de la typologie de Granger, bien qu'ayant subi quelques simplifications. Chez Granger, on retrouve quatre types d'erreurs de complémentation, pour les quatre classes grammaticales acceptant des compléments (les noms, les adjectifs, les adverbes et les verbes). Cette distinction me semblait lourde et peu utile, j'ai donc préféré créer une classe regroupant toutes les erreurs de complémentation (24), peu importe le POS* du mot auquel le complément se rapporte. Le parti pris est de concentrer la réflexion sur la mauvaise construction du complément et non sur le mot auquel il se rapporte (ce dernier important peu dans la construction du complément). Le fait d'avoir regroupé tous les types de compléments en une seule classe d'erreurs permet aussi de s'attaquer aux erreurs qui concernent les compléments circonstanciels. Ceux-là même qui ne sont repris dans aucune typologie, manquement qui nous a fait trancher mes maîtres de stages et moi en faveur de cette classe plus simple, mais plus complète que ses équivalents chez Granger.

24. libre *à choisir / libre de choisir.

(complémentation adjectivale)

*se transformer *dans / se transformer en*

(complémentation verbale)

*Il va à Paris *dans le train / Il va à Paris en train*

(complémentation de phrase)

Gros point de réflexion, la typologie que je viens de présenter se veut donc la plus utile et la plus complète possible. Elle constitue le point de départ de la création de la ressource annotée en erreur et par conséquent des outils permettant l'annotation automatique ainsi que des outils pédagogiques. D'un point de vue pédagogique aussi, la typologie étant basée sur des erreurs réelles d'apprenants elle offre un canevas fixe pour la gestion et l'analyse des erreurs dans les classes de FLE. Tout en naviguant entre les différentes normes et les différentes manières de les appliquer, je me suis efforcée de coller le plus possible aux problématiques des apprenants du français. Ce faisant, j'essaye de

stabiliser la norme mouvante afin de rendre l'erreur exploitable aussi bien par les chercheurs en linguistique que par les techniciens du TAL en passant évidemment par les enseignants et autres étudiants désireux d'utiliser les ressources et les outils qui découleront de tels projets. Dès lors, la suite logique à cette typologie sera de décrire les étapes clés des deux grands projets qui animent ce mémoire. Dans un premier temps, la création de la ressource de base par une campagne d'annotation puis, l'aventure du deep learning pour la création de l'outil d'annotation automatique.

3.2 Campagne d'annotation.

Mon stage de sept semaines m'a permis de participer à un projet lancé par le CENTAL en 2021. Projet qui entre complètement en résonance avec les objectifs de mon mémoire. Le suivi de ce projet m'a apporté le cadre et les ressources nécessaires pour mener à bien les deux phases de ce mémoire, à savoir la création de la ressource et la création d'un outil d'annotation. Bien que le projet « CECR-FR Rator⁴ » ait pour objectif principal « l'évaluation automatique des compétences écrites des apprenants de FLE » les étapes préalables à cet objectif correspondent tout à fait à certaines étapes nécessaires à mon projet de mémoire. Ce constat confirme bien que la détection d'erreurs et l'annotation de ces dernières représentent non seulement une tâche à part entière, mais également les fondations de beaucoup de projets dans le monde de l'éducation (ALAO) et de la recherche (voire de l'industrie, avec des outils toujours plus poussés pour analyser et normaliser les données).

La première étape du projet du CENTAL est donc la « préparation du corpus de production des apprenants du FLE et diffusion d'une section à la communauté scientifique ». C'est en prévision de cette première étape que la création de la typologie d'erreurs s'est inscrite. Et toujours dans l'optique de créer une ressource, un corpus utilisable et intelligemment annoté en informations utiles, nous avons eu besoin de main d'œuvre.

Une fois la typologie créée, il fallait exploiter ce cadre et l'étendre pour créer des balises efficaces ainsi qu'un guide d'annotation. Ce guide sert de point de départ à une étape de grande ampleur, la campagne d'annotation manuelle. Comme je l'ai dit, l'annotation manuelle d'une ressource entière est très onéreuse et chronophage, sans compter la main d'œuvre qui, dans le cas d'une typologie aussi complète, se doit d'être sensiblement qualifiée pour la tâche d'annotation linguistique.

4 Informations disponibles sur le site web du CENTAL : <https://uclouvain.be/fr/instituts-recherche/ilc/cental/cecr-fr-rator.html> (consulté le 6 juin 2023)

Pour créer une ressource représentative pour la communauté scientifique (utilisable telle quelle ou pour nourrir des algorithmes toujours plus gourmands en données), il faut impérativement avoir recours à l'annotation automatique. Peu importe le niveau et le type d'annotation, les outils actuels nécessitent (surtout pour des langues autres que l'anglais) des données pré-annotées sur lesquelles s'entraîner. La campagne d'annotation à moyenne échelle est donc notre seule possibilité pour fournir des données suffisamment précises et annotées afin de développer des outils d'annotation qui soulageront la main d'œuvre humaine. Cette phase cruciale sera donc décrite et documentée en détail dans cette section avant de passer à la partie technique de mon projet.

Avant d'espérer annoter et créer une ressource il faut évidemment un corpus. Ce corpus doit être composé de textes représentatifs du public cible de notre projet, les apprenants du français comme langue étrangère, autrement dit des textes de FLE. C'est là que mon stage m'a été le plus bénéfique, car en participant au projet et en faisant coïncider mes propres recherches à celles de « CECR-FR Rator » j'ai pu bénéficier de leurs accords avec FEI (France Éducation Internationale) et donc de leur corpus de textes d'apprenants du FLE. Ce corpus (un ensemble de textes se voulant exhaustifs dans un domaine et numériquement exploitables) est composé de textes écrits par des apprenants lors d'épreuves écrites du TCF (test de connaissance du français). Il nous permet avant tout d'avoir accès à un nombre conséquent de textes d'apprenants sous une forme numérisée et formatée. Les apprenants les plus représentés dans le corpus sont les apprenants de niveau A2 à B2 mais tous les niveaux du CECR y sont représentés (la description précise du contenu du corpus est à trouver dans le chapitre suivant). Les textes sont globalement tous de longueurs équivalentes et le corpus est renseigné au sujet de la langue première et du niveau de l'apprenant (ressources utiles à exploiter dans des travaux d'analyse d'erreurs par exemple), la description détaillée du corpus et des divisions faites se trouve à la section 4.1 (chapitre 4 : Résultats).

Pour mener à bien l'annotation de ce corpus, il fallait cadrer les grandes lignes de la campagne d'annotation. Dans les faits, les annotateurs, à moins d'être comme moi, à l'origine de la typologie, ne sont pas capables d'emblée de répondre à la tâche qui leur est demandée. Il était clair que dans le cas de la présente campagne, les annotateurs (leur profil détaillé sera décrit plus loin) ne seraient pas aptes et que des phases d'ajustement seraient nécessaires avant de pouvoir annoter la majorité des textes. Nous avons donc, dans un premier temps, décidé de diviser la campagne en trois grandes étapes. Pour ce faire nous avons donc dû aussi diviser le corpus en trois parts cohérentes pour coller aux étapes.

La première étape, qui vise également à valider la typologie et le guide d'annotation, se compose de 100 documents (textes d'apprenants de niveaux différents) (Annexe 18) choisis aléatoirement dans

la ressource. Pour cette étape, les annotateurs annotent les 100 documents disponibles afin de vérifier une première fois leur capacité à réaliser la tâche d'annotation. La seconde étape est similaire à la première puisqu'elle se compose aussi de 100 documents annotés par tous les annotateurs. Elle permet, comme la première, de surveiller les annotations et leurs constances. Enfin, les annotateurs sont lancés sur la dernière phase, celle des annotations individuelles sur le reste des données disponibles dans le corpus. Le descriptif complet des données et de leurs divisions entre les annotateurs sera effectué plus loin (chapitre 4 principalement).

Au préalable, nous l'avons vu plus tôt, il faut extrapoler la typologie déjà créée pour développer des balises qui permettront de faciliter l'annotation et la gestion des textes annotés. Les balises doivent être les plus concises et les plus objectives possible (Granger et al., 2015) pour être exploitables pédagogiquement, scientifiquement et techniquement. Par catégories d'erreurs, l'objectif était donc de rendre compte clairement des types d'erreurs dans une balise concise et facile à manier. Les balises proposées par Granger et al. en 2022 dans son propre guide d'annotation des erreurs d'apprenant ESL (« English as second language ») ont constitué l'inspiration principale pour cette phase. Ma motivation étant que, comme celle présentée ici, la typologie d'erreurs qu'ils proposent se compose de deux niveaux, un niveau linguistique (catégorie large) puis un niveau d'erreurs (type d'erreur spécifique). Pour représenter ce double niveau, ils abordent la balise comme porteuse de la double information. C'est-à-dire une balise unique qui rend compte des deux niveaux à la fois (25).

25. *La balise <FS>* (Elle se compose de la balise <F> :la catégorie des erreurs de forme et concerne les erreurs de graphie « spelling » d'où l'ajout de la lettre S.)

The Louvain Error Tagging Manual Version 2.0
(Granger et. al. 2022 :9)

Pour ce qui concerne mes propres balises, le Dr. Wilkens a soulevé le caractère opaque des balises de Granger et al. Nous avons donc décidé d'exploiter cette idée de doubles balises, mais de les rendre plus explicites. Ainsi, les catégories linguistiques n'étant qu'au nombre de trois, il était facile de les distinguer même en n'utilisant qu'une seule lettre. J'ai par ailleurs pris la liberté d'explicitier plus amplement la partie de tag consacrée à la classe d'erreurs (26). Cette décision de rendre les balises les plus explicites possibles était avant tout pour faciliter la tâche des annotateurs et éventuellement améliorer leurs performances.

26. *La balise <L-GRAPHIE>* (Elle se compose de la balise <L> : la catégorie des erreurs du lexique et concerne les erreurs de graphie, d'où l'ajout de la mention « GRAPHIE »).

Une fois le canevas des balises élaboré, les futurs annotateurs auront besoin d'un guide (mode d'emploi) de la typologie afin de réaliser au mieux la tâche. Ce guide d'annotation doit être clair, complet, mais ni trop dense ni trop long pour ne pas surcharger l'annotateur. L'objectif étant de les rendre suffisamment compétents pour la tâche sans en faire un cours magistral de linguistique (trop onéreux et chronophage). Le point de départ de la rédaction du guide est évidemment la typologie, mais cette dernière doit être agrémentée d'un bon nombre d'informations pour permettre aux annotateurs, même novices dans le domaine ou non-expert, de comprendre et d'annoter en suivant la philosophie de la typologie et du projet. Ayant déjà été annotatrice pour des campagnes du CENTAL, j'avais, par chance, déjà utilisé des guides d'annotation. De plus, j'avais en ma possession « The Louvain Error Tagging Manual Version 2.0 » (Granger et al. 2022 :9) qui constitue un document efficace sur lequel se baser. La typologie d'erreurs étant déjà très complexe, il m'a fallu être extrêmement explicite dans la description des erreurs et dans l'explication du placement des balises. Ce point est crucial si on espère réaliser une campagne efficace, en d'autres termes : avoir des annotateurs qui annotent tous de manière uniforme. En dehors des descriptions des erreurs, j'ai aussi ajouté une série d'exemples permettant de comprendre dans quel contexte la balise devait être utilisée. Ces exemples étaient, d'abord, tirés des différentes typologies que j'avais prises en exemple ou des travaux d'analyse d'erreurs, puis, quand j'ai eu accès à certains documents du corpus, tirés directement des épreuves du TCF. Une entrée typique du guide à ce stade précoce de la campagne s'articulait comme suit (Fig. 2) :

1. **<L>** Erreurs lexicales:
 - 1.1 **<DIA>** Erreurs de diacritiques,

Toutes les erreurs qui concernent les signes diacritiques:

Utilisation des accents

**océan / océan (oubli)*

**fôret / forêt (mauvais placement)*

**nécessaire / nécessaire (ajout)*

**français / français (s'applique aussi pour la cédille)*

Utilisation du trait d'union (agglutination de mots composés ou division de mots simples)

**portebagage / porte-bagage (agglutination)*

**centre ville / centre-ville (oublis)*

**contre-action / contraction (division)*

Utilisation de l'apostrophe

**d y / d'y*

(Fig. 2) Entrée « erreurs de diacritique », dans la catégorie des erreurs lexicales [nom, explication, puis série d'exemples]

À ce stade, le guide est clairement trop peu explicite, surtout si on prend en compte le profil des annotateurs. Au nombre de cinq, les annotateurs, ou devrais-je dire les annotatrices, sont pour la

plupart issus d'études littéraires en français, mais ne sont pas experts en linguistique et encore moins FLE. La typologie étant particulièrement complexe et les choix particulièrement pointus en termes de linguistique et de pédagogie, je me devais d'améliorer le format des entrées du guide. Mes maîtres de stage-ayant plus d'expérience que moi en ce qui concerne les campagnes d'annotations, ils m'ont suggéré d'ajouter une série de contres exemples ou de mentions « attention » qui permettrait aux lecteurs de se focaliser sur ce qu'il faut faire en sachant quels sont les écueils ou pièges à éviter. Afin de distinguer toutes les différentes parties d'une entrée dans le guide, j'ai pris la décision d'y ajouter un code couleur clair et fixe pour chacune des entrées. Il est vrai que le guide pris en exemple n'exploite pas autant les couleurs, et c'est l'un de ses points faibles qui le rend peu lisible et surtout peu clair.

Pour mettre mon guide à l'épreuve et étoffer sa création, l'annotation manuelle a d'abord été testée par mes soins, sans l'utilisation d'une plateforme et avec comme seul canevas les versions « bêta » du guide. C'est grâce à cette phase de « pré-test » que certains cas particuliers non discutés dans la typologie ou dans le guide (tous deux très généraux encore à ce stade) ont pu être détectés et intégrés à ces derniers. Par la suite, lors de la première vraie phase de la campagne, le guide a encore été éprouvé dans les conditions réelles, ces améliorations progressives ont accompagné chaque étape de la campagne avant d'aboutir à la version « finale » présentée en annexe (Annexe 7). Cette étape cruciale a mis en lumière par exemple une incompréhension générale vis-à-vis du placement des balises. Il est vrai que le placement exact des balises n'était pas clairement répertorié pour chaque balise dans le guide. De plus, ces informations avaient été données à l'oral lors de la réunion de formation des annotatrices, mais rien n'avait été figé dans le marbre. J'ai donc ajouté une mention particulière à chaque balise précisant où la placer et comment (par défaut en orange pour chaque entrée du guide). Avec mes maîtres de stage, nous sommes partis du principe que dans le guide, il valait mieux répéter les informations à toutes les étapes plutôt que de compter sur la compréhension de l'implicite par les annotateurs (non experts et non familiers avec la typologie). Voici l'évolution d'une entrée typique du guide dans sa version « finale » : (Fig. 3).

1. <L> Erreurs lexicales :

1.1 <DIA> Erreurs de diacritiques,

Toutes les erreurs qui concernent les signes diacritiques sont à classer dans cette catégorie.

NE PAS CONFONDRE, signe diacritique (trait d'union, accent, apostrophe) et signe de ponctuation (parenthèses, points, guillemets).

Utilisation des accents.

Annoter uniquement le mot qui comporte l'erreur d'accent.

*ocean / océan (omission)

- Y compris, l'absence d'accents sur les majuscules -

*fôret / forêt (mauvais placement)

*nécessaire / nécessaire (ajout)

*français / français (s'applique aussi pour la cédille)

Utilisation de l'apostrophe.

*d y / d'y

Toujours annoter des deux mots et l'espace qui composent la construction erronée.

Attention ne concerne pas les erreurs dans la rédaction des phénomènes d'élision et de liaison qui sont annoté <M-EUP>.

Utilisation du trait d'union (agglutination de mots composés ou division de mots simples) .

Attention, lorsqu'un espace indésirable est placé entre le trait d'union et le mot, cette erreur est aussi considérée comme une erreur de diacritique.

Toujours annoter les deux mots, l'espace et le tiret qui composent la construction erronée.

*portebagage / porte-bagage (agglutination)

*centre ville / centre-ville (omission)

*contre-action / contraction (division)

*porte- bagage/ porte-bagage

(Fig. 3) Entrée « erreurs de diacritique », dans la catégorie des erreurs lexicales.

Pour finir, le guide d'annotation de Granger et al. m'a permis d'ajouter un dernier élément important à mon guide. Élément qui a nécessité beaucoup de réflexion, car il allait régir toute la campagne d'annotation. Il s'agit des « guidelines » ou de la conduite à suivre lors de l'annotation. Ces lignes directrices dépendent à la fois de l'objectif de la tâche, ce qu'on veut annoter précisément et la raison, mais également de l'outil d'annotation utilisé pour faciliter la campagne (voir paragraphe suivant). Elles permettent aussi d'introduire des concepts fondamentaux comme certaines justifications linguistiques ou encore la philosophie-même de la tâche. Toutes ces informations permettent aux annotateurs de comprendre l'ampleur du travail et de s'informer afin de réaliser l'annotation au plus proche de ce qui est attendu par l'équipe du CENTAL pour leur projet ou par ma personne pour mon mémoire. Dans ces consignes, j'aborde, entre autres :

- le concept du *token**
[c'est à dire l'annotation au plus proche du mot (sans subdiviser un mot et en évitant au maximum d'annoter plusieurs tokens/mots d'un coup)],
- le concept de style et de registre
[ils sont à bannir dans la tâche qui est demandée (ce qui est grammaticalement correct est correct peu importe si cela « sonne » mal pour une personne native du français)],
- Ou encore, quelques rappels linguistiques
[gestion des noms propres, des constructions emphatiques, de la nouvelle orthographe, etc.].

Ces directives sont construites au fur et à mesure des phases de test et de vérification des annotations pour pallier les « erreurs » que les annotatrices commettent dans les premières étapes. Pour ne citer qu'un exemple, au bout de la première phase d'annotation, les annotatrices ont eu tendance à faire preuve de « paresse » à la fin de leurs séries d'annotations en annotant une seule balise si plusieurs mots voisins comportaient la même erreur. C'est donc dans les « guidelines » qu'il fallait les « rappeler à l'ordre » en justifiant la pratique par le fait que la gestion des annotations par la suite nécessite une annotation séparée. Toujours pour pallier les « erreurs » et répondre aux questionnements des annotatrices et toujours grâce au guide de Granger et al. (2022), il fut nécessaire d'ajouter une quatrième catégorie d'erreurs. Cette catégorie nommée « ERR » n'est à utiliser qu'en dernier recours lorsque la phrase, la construction syntaxique ou le texte ne font aucun sens. Cette catégorie regroupe donc tous les cas d'incompréhension générale, c'est-à-dire les cas où même en corrigeant erreur par erreur (en suivant les principes du guide) il est impossible de démêler ce que l'apprenant a voulu écrire et quelles sont les véritables erreurs commises.

Les phases de la campagne n'ont pu être lancées qu'à une seule condition, trouver un outil facilitant l'annotation et la gestion des annotations. Cet outil devait répondre à des critères très précis :

- Permettre la customisation du tagset (des balises),
- Permettre la superposition des balises (cas où plusieurs erreurs coexistent),
- Permettre une utilisation en ligne et une installation sur le serveur du CENTAL,
- Permettre une gestion des documents par annotateur et un profil particulier par annotateur,
- Éventuellement une vue sur le temps passé par annotation,
- Un format d'import et d'export des données facilement manipulable,
- Une gestion du gold-standard intégré ainsi qu'un outil de vérification des annotations.

Après plusieurs investigations, mon choix s'est porté sur le programme INCEpTION⁵ (Klie et al., 2018). Le seul critère que la plateforme ne remplissait pas est celui de la gestion du temps, critère qui était d'ailleurs optionnel. La grande facilité qu'INCEpTION apporte est la gestion du Gold standard intégrée et la vérification des annotations. Cela m'a permis de gagner un temps considérable étant donné que j'étais chargée de créer le Gold standard et de vérifier les annotations dans la dernière phase de la campagne. Nous avons décidé, par ailleurs, de vérifier cinq documents par tranche de 50 documents annotés par annotateurs. Néanmoins, le choix de ce programme m'a causé du fil à retordre pour l'analyse des données, le problème ainsi que les solutions mises en place seront discutés dans le chapitre 4.4 (retraçant la partie pratique du mémoire) ainsi que dans les annexes (Annexe 8).

Pour finir, il a fallu se pencher sur l'analyse des résultats et sur l'évaluation de l'efficacité de la campagne. Pour mesurer la qualité des annotations, il est de commune mesure de calculer l'accord inter-annotateur lors des premières phases d'une campagne, pour être plus précise, dans notre cas, de mesurer une fois l'accord à la moitié de la première phase, puis entre les deux phases de 100 documents (après une réunion de clarification avec les annotatrices) et en fin sur le dernier jeu de 100 documents. Pour ce faire j'ai mesuré l'accord via la formule de l'*alpha de Krippendorff* (mesure qui permet plus que deux annotateurs et plus que deux seules balises) (Fig. 4).

⁵ source et documentation : <https://inception-project.github.io>
Dernière visite : 17 mars 2023.

$$\hat{\alpha}'_K = \frac{P_a^* - P_e}{1 - P_e}, \text{ where } \begin{cases} P_a^* = (1 - \epsilon_n)P_a + \epsilon_n, \epsilon_n = 1/2n \\ P_a = \sum_{k,l} w_{kl} P_{kl}, P_e = \sum_{k,l} w_{kl} \pi_k \pi_l. \end{cases}$$

(Fig. 4) formule commune pour le calcul de l'alpha de Krippendorff (Artstein et Poesio, 2008).

Il a donc été nécessaire d'effectuer ces mesures à plusieurs niveaux, entre les annotations de tous les annotateurs et entre tous les annotateurs et le Gold standard. Le tout à trois reprises, le Krippendorff alpha a été calculé sur chacune des étapes des phases de test et de création du Gold standard. De plus, pour avoir une précision maximale, nous avons décidé (avec la même formule) d'évaluer les performances de chacun des annotateurs (deux à deux) vis-à-vis du Gold standard. Pour plus de précision, mes maîtres de stage m'ont aussi encouragée à mesurer le pourcentage d'accord par le biais de tables de contingences (entre les annotatrices et le Gold toujours). Ces résultats seront présentés et discutés dans le chapitre 4. Ils seront également accompagnés d'une analyse de la ressource ainsi créée. Pour effectuer ces calculs, j'ai réalisé un petit script python (Annexe 12 - 13). Ces scripts également seront explicités dans le chapitre suivant.

Le choix de réaliser une campagne d'annotation peut être remis en question d'un point de vue dépense de temps et d'énergie. Mais une campagne d'annotation reste sans conteste la méthode la plus répandue pour obtenir un nombre conséquent de données annotées tout en gardant un œil sur la qualité de la production. La ressource créée par cette étape m'a permis de nourrir les outils TAL d'annotation automatique avec des données réelles représentatives des erreurs d'apprenants FLE. Dans les sections suivantes, je reviendrai sur les choix en termes de modèles (réseaux de neurones artificiels) puis d'entraînement et d'évaluation de l'annotation produite par ces modèles.

3.3 Choix des modèles.

Pour en revenir sur le projet mené par le CENTAL, une des étapes de ce dernier est « [l'] annotation automatique des phénomènes linguistique par IA pour décrire la compétence écrite des apprenants du FLE ». Pour ce faire il faut des modèles IA suffisamment performants pour annoter les textes de manière fiable. On l'a vu, les méthodes les plus fiables à l'heure actuelles sont celles réalisées avec des réseaux de neurones artificiels. Pour le français, peu de ces méthodes ont été testées et mises en œuvre. Il est intéressant ici d'en tester plusieurs afin de, non seulement, créer un modèle performant, mais aussi, de tenter de réduire l'écart entre les recherches sur l'anglais et sur le français. De plus, comparer les résultats de différents modèles avec ceux obtenus pour des tâches similaires en anglais ne peut être qu'enrichissant du point de vue de la recherche (peu importe la langue cible). J'ai donc pris la décision, au regard des modèles déjà expérimentés par le passé, d'utiliser dans un premier lieu un LSTM bidirectionnel comme *baseline** (point de départ de la comparaison des résultats). Cette méthode n'est clairement plus la plus performante à aujourd'hui, mais elle permettra d'intégrer le projet dans le continuum de projets effectués dans le passé. Cela donne aussi un autre point de comparaison entre le français et l'anglais (vérification des performances avec notre modèle / nos données et les modèles / données prévues pour l'anglais). Au fil des expériences effectuées et de nouvelles publications plus proches du projet, le choix de la baseline a été modifié pour un modèle d'apprentissage de surface et non plus d'apprentissage profond : un SVM. L'évolution vers ce choix et ses raisons sera évidemment documentée dans ce chapitre.

Puis je m'attaquerai à une architecture bien plus performante, l'architecture Transformers et son maître incontesté : le modèle BERT. Il est important de noter que pour toute la partie de création des modèles de deep learning, je travaille avec *Keras* [12] et donc *Tensorflow* [24] comme bibliothèques.

Le premier modèle de détection d'erreurs construit était un réseau de neurones récurrents bidirectionnel (Bi-LSTM) avec une couche d'embeddings (plongements lexicaux) et des couches « fully connected » pour la classification. Le but de ce modèle de base était de prédire si une phrase contient des erreurs ou non. Pour cela, il prend en entrée, une séquence de mots représentée par des embeddings pré-entraînés et utilise un mécanisme de récurrence pour tenir compte du contexte des mots dans la phrase. Ce premier modèle se basait en réalité sur ceux produits pour des textes anglais et une classification binaire (correct / incorrect). Les exemples de Bi-LSTM testé par Rei (2017) et Rei et al. (2016) avait donc guidé mon choix vers cette baseline. D'une part, le modèle était adapté au français et d'autre part il était décliné en trois versions distinctes selon le nombre des classes présentes dans mon guide. Nous avons donc un modèle pour chaque version du corpus :

- Une version du corpus pour la détection d'erreurs (classes binaires)
- Une version adaptée à la détection du premier niveau d'erreurs (5 classes linguistiques)
- Et une version reflétant l'entièreté du guide (23 classes d'erreurs).

Pour adapter les modèles d'exemples au français, il suffisait de faire appel à des plongements lexicaux pré-entraînés sur des corpus en français. En effet, GloVe, par exemple, est un algorithme qui produit des vecteurs d'embeddings de mots à partir de corpus de textes. Cependant, les embeddings de GloVe sont basés sur des corpus de textes en anglais, donc il est possible que les performances de votre modèle ne soient pas optimales pour des textes en français. Pour obtenir de meilleures performances sur des textes en français, il est recommandé d'utiliser des plongements lexicaux pré-entraînés spécifiquement pour le français. C'est ce que je me suis donc attelée à faire en modifiant mes dictionnaires d'embeddings et mes indexes avec FastText (Bojanowski et al., 2016). Comme GloVe (Pennington et al., 2014) et d'autres, FastText est un modèle de plongement de mots pré-entraîné qui peut être utilisé pour créer des représentations vectorielles de mots (ce que l'on recherche ici). En plus d'offrir un large choix de langues sur lesquelles l'algorithme est entraîné (dont le français évidemment), FastText utilise des sous-mots (subwords) pour construire des représentations de mots.

Une fois ce modèle optimisé pour la gestion de textes en français, il fallait encore qu'il soit capable de prédire plus de deux classes. Il a été simplement nécessaire de changer dans mon modèle même la fonction d'activation de la couche de sortie en une fonction *softmax* ce qui m'a permis de définir un nombre de classes supérieur à deux. En fonction de la typologie, les classes sont variables, donc j'ai dû ajouter un argument « nclasses » modifiable dans l'appel même de la fonction de construction du modèle. J'ai aussi ajusté la forme de la sortie de la dernière couche en fonction du nombre de classes (nclasses) en spécifiant ce nombre dans la couche de *Dense*.

Malheureusement, ce modèle ne semblait pas fonctionner correctement pour les tâches de détection et surtout pour les détections multi-classes. De plus, dans l'implémentation-même, le Bi-LSTM semblait rencontrer quelques difficultés. Après une discussion avec Rodrigo Wilkens, j'ai pris la décision de ne pas investir d'énergie dans la résolution de ce problème pour plutôt me retrancher sur une alternative qualitative me demandant moins de réflexion. En effet, l'idée première dans l'utilisation au préalable du Bi-LSTM était d'avoir un point de comparaison stable entre notre projet et ceux effectués en anglais. Or, vous le verrez, les résultats du BERT prouvent déjà une continuité certaine entre ces projets et le nôtre.

L'énergie dépensée dans la résolution de ce problème uniquement pour avoir un point de comparaison avec le BERT paraissait alors une perte de temps. Surtout quand on sait que l'objectif

principal de ce projet est d'ouvrir la voie vers des modèles de plus en plus performants dans la classification des erreurs d'orthographe. Néanmoins, il est nécessaire de pouvoir situer à corpus identique les performances du modèle central du projet. Pour parvenir à cela, nous avons pris le parti, non pas d'inclure l'étude dans le continuum des études anglophones sur les Bi-LSTM, mais bien de partir sur la base d'une étude de 2023 sur un corpus en français (Noreskal et al.). Comme on l'a vu dans le support théorique (section 3.2 du chapitre 2), pour ce projet, l'apprentissage de surface est privilégié comme baseline. Nous avons donc décidé de simplifier considérablement l'étape de la baseline au profit de l'analyse du modèle BERT et de ses performances (élément central du projet). C'est tout naturellement que nous avons alors fait le choix d'une machine à vecteurs de support (SVM) pour effectuer ces résultats préliminaires.

L'idée est d'avoir un modèle basique et fiable, le plus performant possible tout en sachant que les résultats ne seraient pas glorieux. Le SVM devra être capable de réaliser les trois classifications : binaire, à 5 classes et à 23 classes et être adaptée au français. Pour ce qui est du corpus, il a fallu faire un corpus d'entraînement et un corpus de test, à l'inverse des deux autres modèles testés qui requièrent généralement un corpus de développement (de taille similaire au corpus de test) aussi. Pour cela rien de plus simple, les corpus « TRAIN », « TEST » et « DEV » existants déjà à ce stade, il a suffi de réunir les corpus « TRAIN » ET « DEV » en un seul et même corpus d'entraînement. Pour l'implémentation du SVM, nous avons fait appel à différentes librairies disponibles en python. Premièrement, la librairie « *sklearn* » [21] qui permet de manipuler (créer et entraîner) des modèles simples comme ici le SVM. C'est cette librairie qui permet aussi dans ce cas de calculer les métriques d'évaluation de ce modèle comme la F1, l'« *accuracy* » [2], la précision et le *rappel* [19] (discutés dans la section suivante). C'est aussi toujours avec *sklearn* que je peux encoder mes balises de sorte que le format corresponde à celui attendu pour l'entraînement du/des modèles. La librairie a d'ailleurs été utilisée pour encoder les labels (fonction « *LabelEncoder* ») du Bi-LSTM, obsolète, et du BERT. Ensuite, pour maximiser les performances du SVM, la première idée a été d'utiliser des embeddings (efficaces pour les Français) pré-entraînés plutôt que des méthodes triviales comme TF-IDF. La question de quel embedding utiliser pour le français avait déjà été posée lors de l'implémentation du Bi-LSTM. Nous avons donc exploité les plongements lexicaux de FastText (plus précisément le modèle entraîné sur des textes en français, téléchargés directement depuis leurs sites⁶). Pour les manipuler, il est utile de faire appel à une autre librairie bien connue en traitement automatique du langage pour gérer des modèles pré-entraînés : « *Gensim* » [9]. Une fois les données d'entrée

6 téléchargement du modèle FastText via le site de la librairie FastText même :

(consulté le 12 juin 2023).

encodées et sectorisées grâce aux embeddings, il est temps de passer aux choix pour l'entraînement du SVM en lui-même. D'emblée, les choix sont des choix par défaut puis par le biais d'expériences (lancement successif et comparaison des résultats). Il faut optimiser des valeurs pour obtenir les résultats les moins catastrophiques, pour ne pas dire les meilleurs. Afin d'éviter de lancer l'entraînement un nombre incalculable de fois, le plus simple est d'effectuer ces choix d'optimisation (choix des hyperparamètres, noyaux, cout et gamma) via une recherche sur grille : « GridSearch »]. Une fois les meilleures combinaisons de paramètres identifiées, ['C' = 0.1, 'Kernel' = linear, 'Gamma' = 0.1], il est trivial de lancer l'entraînement du SVM afin d'avoir des résultats dignes d'une baseline.

Pour ce qui est de la principale architecture expérimentée dans le cadre de mon projet de mémoire et du projet du CENTAL, nous cherchions une approche plus prometteuse que le Bi-LSTM ou les SVM qui devient progressivement obsolète depuis plus de cinq ans. Au regard des progrès en termes de performance proposés par les approches *Transformers* et les résultats prometteurs vu dans les études de détection d'erreurs pour l'anglais, nous avons décidé d'adapter une instance pré-entraînée de BERT à notre tâche de diagnostic d'erreurs (annotation multi-classe). Les travaux d'annotations d'erreurs automatiques pour le français (apprenants ou natifs) étant peu exploités, voire inexistant, nous avons procédé à l'implémentation d'une version simple de BERT pour réaliser la tâche d'annotation. Cette version permettra d'avoir un point de comparaison avec la baseline en espérant que les résultats corroborent la réputation plus performante de BERT. Mais elle permettra aussi à d'autres travaux futurs d'avoir un ordre de grandeur en termes de résultats ainsi que, je l'espère, une baseline solide sur laquelle construire des modèles encore plus performants pour le français.

La construction d'un modèle de détection d'erreurs à l'aide de BERT (Bidirectional Encoder Representations from Transformers) implique plusieurs étapes clés. Tout d'abord, il est important de comprendre les avantages de l'utilisation de BERT par rapport à un modèle LSTM (*Long Short-Term Memory*) qui a été largement utilisé pour résoudre les problèmes de classification ou que des SVM qui il y a encore une dizaine d'années régnaient en maître sur les tâches de classification linéaires. Cependant, ces modèles plus anciens ont leurs limites en termes de performance et de précision. BERT est l'architecture de transformer la plus populaire et le modèle le plus utilisé pour l'anglais. CamamBERT, quant à lui, est une architecture RoBERTa formée pour le français. Les Transformers sont capables de capturer les dépendances de longue portée dans le texte en utilisant une approche d'*attention*, ce qui les rend très efficaces pour les tâches de traitement de texte telles que la classification, la génération de texte et la détection d'erreurs.

La première étape pour construire un modèle de détection d'erreurs à l'aide de BERT consiste à charger le modèle BERT pré-entraîné à partir de la bibliothèque *HuggingFace* [11] Transformers. Il a été pré-entraîné sur de grandes quantités de données pour apprendre des représentations de mots riches en sémantique.

Une fois que le modèle BERT pré-entraîné est chargé, il doit être adapté à la tâche spécifique de détection d'erreurs. Cela implique de *finetuner* le modèle sur les données d'entraînement en utilisant une technique appelée "*transfer learning*". Ce qui veut dire que je vais ajuster les poids du modèle BERT pré-entraîné aux spécificités de la tâche de détection d'erreurs en utilisant les données d'entraînement spécifiques à cette tâche. Je commencerai donc par l'ajuster à l'image des exemples pour l'anglais (Kaneko et al. 2020) pour pouvoir comparer fidèlement les modèles francophones aux résultats des modèles anglophones. Il suffira d'ajuster la version française des modèles anglophones de détection d'erreurs (classification binaire) aux classifications multi-classes du corpus d'entraînement.

Pour commencer, il faut charger le modèle de BERT pré-entraîné depuis HuggingFace (bibliothèque choisie pour gérer ce type de modèles pré entraînés) en utilisant la classe « *BertForSequenceClassification* », qui est un modèle BERT pré-entraîné spécifiquement conçu pour la classification de séquences. Le gros avantage de HuggingFace est qu'il permet d'avoir un environnement complet et une interface facile à manier pour utiliser tout ce que BERT a à proposer. La partie de pré-traitement des données se fait donc en utilisant directement le *Tokenizer* [26] proposé par HuggingFace et directement adapté à l'usage dans BERT. La classe « *BertTokenizer* » convertit les phrases en vecteurs de nombres que le modèle BERT peut comprendre. Une fois les données converties et pré traitées, il suffit de les passer dans le modèle de BERT pour la classification séquentielle (étape de fine-tuning). C'est-à-dire entraîner une nouvelle fois le modèle sur les nouvelles données. Ce nouveau modèle dédié à la détection d'erreurs peut prédire les classes des séquences d'entrées grâce à la fonction « *forward* », la sortie de cette fonction est un tenseur qui contient les probabilités prédites pour chaque classe. Enfin, il ne restera plus qu'à évaluer les étiquettes (balises) prédites par le modèle en les comparant aux étiquettes attendues (gold-standard). Les métriques d'évaluation ainsi que la production des résultats seront détaillées dans la suite de la section « Méthodologie ».

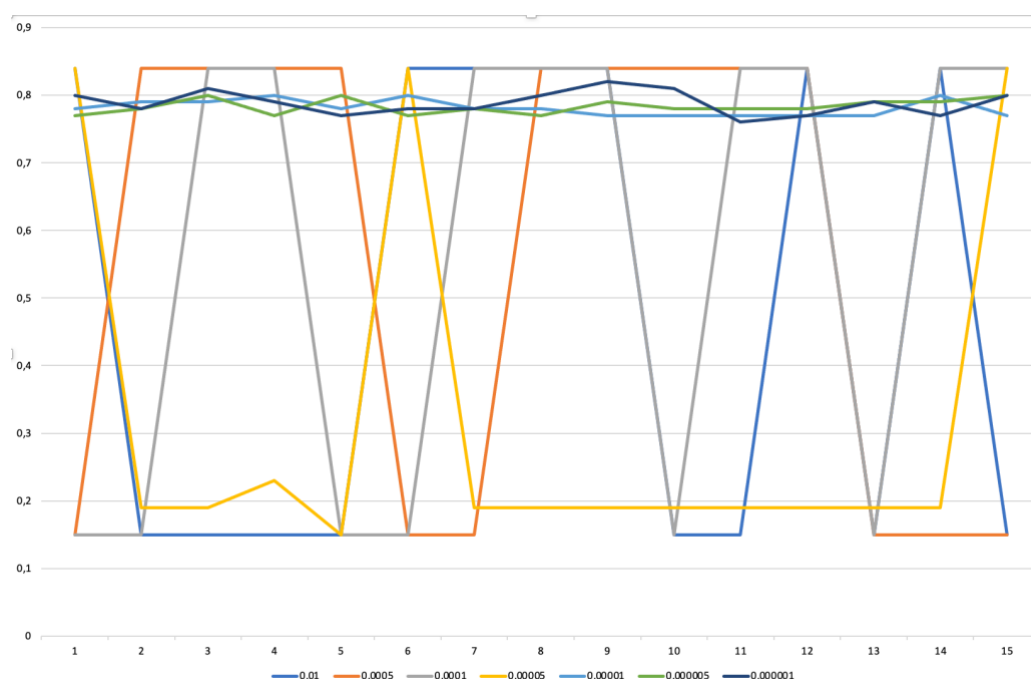
Ici encore, grâce à HuggingFace et au fait que BERT soit pré-entraîné sur un nombre de données (non étiquetées) énorme, l'adaptation à une classification binaire ou multi-classe est extrêmement simple. Il suffit de changer le nombre de classes dans les paramètres lors du chargement du modèle : `[num_labels = n]` où `n` peut varier entre 2 et le nombre de classes présentes dans nos données. Il faut aussi penser à adapter la fonction de perte (*loss*), pour la classification multi-classe, je préférerai la

« *sparse categorical cross-entropy* » qui est surtout utilisée lorsque les étiquettes sont fournies sous forme d'entiers, dans notre cas des nombres de 1 à len (liste des balises) [soit la longueur de la liste de balises utilisée]. Alors que pour la classification binaire, j'ai opté pour une fonction de perte binaire : « *binary cross-entropy* ». Pour adapter ces codes issus des exemples anglophones aux données en français, rien de plus simple : il sera nécessaire de charger une version de BERT pré-entraîné sur du français. Donc, au lieu d'utiliser « *bert-base-uncased* » nous avons choisi « *camembert-base* » (Martin et al., 2020). Ce modèle est pré-entraîné sur un très grand nombre de textes rédigés en français avec une méthode plus récente appelée mBERT, qui combine plusieurs techniques d'entraînement multilingue pour améliorer la performance sur les langues romanes telles que le français. Il faut aussi adapter le Tokenizer qui ne sera plus la classe « *BertTokenizer* » mais bien la classe « *CamembertTokenizer* » qui fort heureusement fonctionne pour le reste exactement de la même manière.

Mais comme pour le SVM, un modèle basé sur les paramètres et hyper paramètres par défaut, n'assure pas systématiquement les meilleurs résultats. Donc, comme pour le SVM, nous avons procédé (grâce à l'aide précieuse de Monsieur Wilkens) à une série d'expérimentations afin d'évaluer les combinaisons les plus performantes.

- La première expérience consistait à identifier le nombre d'epoch le plus optimal. Pour y parvenir, nous avons comparé les valeurs de « Loss » et d'« accuracy » à chaque epoch (entre 1 et 10) de l'entraînement sur le jeu de données binaires, le plus prometteur. Le choix de se limiter à 10 epoch se base sur l'analyse minutieuse de la littérature, notamment chez Noreskal et al. (2023 :9). Le choix le plus efficace était un entraînement entre 5 epoch et 9 epoch.
- La deuxième expérience a été d'optimiser l'« optimizer » du modèle, le choix de « ADAM » correspondait à la littérature, mais le pas d'apprentissage (« Learning rate ») peut encore varier. Pour l'expérimenter, il suffisait de passer cet élément en argument du programme puis de le faire varier et d'observer les meilleurs résultats. Pour effectuer ces expériences, nous avons mis à l'épreuve la classification la plus performante (classification binaire) en faisant varier le Learning rate, l'objectif étant de comparer les valeurs d'accuracy afin d'identifier quel Learning rate et quel nombre d'epoch sont les plus performants (Fig. 5).

- Enfin, après une observation préliminaire du corpus et des résultats, une dernière chose devait être améliorée : le poids accordé aux balises lors de l'entraînement. En effet, le corpus contenant bien plus de token « corrects » que de token « incorrects » il fallait corriger ce déséquilibre afin d'éviter un raccourci qui négligerait les classes minoritaires (qui contribue moins à l'évolution de la *Loss* du modèle). Pour rétablir l'équilibre en fonction de la présence des balises dans le corpus, il a suffi d'utiliser la fonction `class_weight` avec l'option 'balance' de `sklearn` qui permet en fonction du nombre de chaque balise de « balancer » leurs poids pour contrecarrer le déséquilibre induit par le corpus.



Le graphique représente l'évolution des performances (en fonction du nombre d'epoch d'entraînement) du modèle CamemBert selon une variation du learning rate allant de 0,01 à 0,000001. Les meilleurs résultats sont obtenus avec les Learning rate (marine, bleu ciel et vert) c'est-à-dire 0,00001, 0,000005 et 0,000001, avec un pic à de performance à 9 epoch pour le learning rate 0,000001.

(Fig.5) graphique de l'évolution de la valeur « val_accuracy » en fonction du nombre d'epoch. Les lignes représentent la variation du Learning rate.

Au regard du corpus (décrit dans le chapitre suivant) on sait qu'environ 85% des tokens sont en réalité des formes correctes et 15% sont des erreurs. Les quatre premiers Learning rates sont donc inutilisables car ils oscillent entre une valeur de 0,84 et de 0,15 ce qui indique qu'ils prédisent de manière générale 100% de formes correctes ou 100% de formes incorrectes selon les epoch. Les trois autres Learning rates (les plus faibles) semblent équivalents, le plus performant étant $1e-6$ au bout de 9 epoch (val_accuracy = 0.82).

Avec cette optimisation et celles citées plus tôt, il ne restait plus qu'à mettre les modèles à l'épreuve du corpus de test et d'en analyser les résultats.

Ces deux approches (BERT et SVM) me permettront d'analyser au mieux les performances et de les inscrire dans le continuum des projets d'annotation et de détection d'erreurs déjà effectués par le passé. Avec des modèles performants, j'espère entre autres ouvrir la porte à l'expérimentation pour créer des outils de plus en plus utiles pédagogiquement. L'implémentation des modèles est d'ailleurs disponible en annexe (Annexe 14 - 17). Mais l'outil créé ici va permettre en lui-même d'augmenter considérablement la taille de la ressource annotée manuellement, avec le plus de fiabilité évidemment. Il permettra aussi d'annoter d'autres textes (d'autres tests de niveau de langue, des textes d'apprenants réalisés en classe de FLE ...) afin d'alimenter les sources et de permettre des analyses d'erreurs de plus en plus représentatives ou de plus en plus adaptées à des groupes classes spécifiques. Dans la suite de cette méthodologie, je présenterai les corpus que je mentionne dans cette section ainsi que les méthodes d'évaluation et de test des modèles présentés ici. En effet, rien que dans les versions présentées, les modèles permettent d'obtenir trois types d'annotations pour deux types de corpus : des annotations binaires et multi-classes (quatre et plus) pour des corpus anglophones (données issues des différents travaux choisis comme guide et baseline du projet présenté ici) et francophones.

3.4 Corpus, entraînement et évaluation des modèles.

Il est temps maintenant de discuter des ressources nécessaires à l'entraînement et au test des modèles décrits plus haut. De tels modèles ont besoin d'un nombre conséquent d'instances de texte pour pouvoir être performants dans la tâche qui leur est demandée. D'autant plus que les modèles que j'ai présentés et choisis d'utiliser ont besoin de données annotées afin de réaliser leurs apprentissages. Que ce soit pour le SVM qui ne fonctionne qu'en apprentissage supervisé ou pour le BERT qui, après un entraînement sur de grandes quantités de données non supervisées, nécessite aussi dans l'étape du fine-tuning de données annotées, les ressources se doivent d'être de taille conséquente afin d'optimiser les résultats. Il est évident qu'ici aussi le fait d'avoir eu accès aux données disponibles au CENTAL est un avantage considérable. Les données en anglais fleurissent et il est aisé de répliquer ou d'améliorer les avancées techniques des travaux basés sur l'anglais en gardant des corpus similaires (en termes de taille ou de type) que ce soit pour adapter des modèles existants à d'autres problématiques ou d'améliorer des modèles existants en étant sûr de garder des données équivalentes (pour ne pas biaiser les résultats). Pour le français, les choses se compliquent, comme je l'ai montré dans le support théorique de ce mémoire, les travaux sur le français sont peu nombreux, voire inexistant pour la tâche qui nous anime ici.

Certes certains travaux existent, mais ils sont bien souvent obsolètes et font appel à de techniques dépassées depuis longtemps (et demandant évidemment moins de données). Après avoir tenté de décrire et de documenter au mieux les choix en termes d'algorithmes, je vais maintenant décrire et documenter nos choix en termes de données. Ce sont ces données qui mèneront aux résultats présentés et discutés dans le chapitre suivant. Dans un premier temps, je m'attaquerai aux différents types de données récoltées, leurs traitements et leurs formats. Aussi, je m'attarderai sur certains choix méthodologiques concernant les données. Puis, dans un second temps, je reviendrai plus près des modèles pour concrétiser la phase d'évaluation (sur base des données évidemment) qui permettra de faire le pont avec la section « résultats » qui suivra.

Pour commencer les expérimentations et la création des algorithmes, il a fallu partir des travaux réalisés sur l'anglais. Ces travaux pour les deux modèles testés permettront d'inscrire le projet en français dans la lignée de ceux déjà présentés, mais aussi de comparer avec plus de précision l'efficacité des modèles. Les travaux francophones n'étant pas actuels, je n'ai aucun point de comparaison avec des données ou des mesures d'évaluation de modèles. Utiliser des exemples issus de travaux existants et efficaces m'a permis d'implémenter les modèles en français (forgés sur ceux cités plus haut) avant même d'avoir suffisamment de données pour les tester et les entraîner (sans accès direct aux données en français) chronologiquement j'ai donc pu avancer simultanément sur la

campagne d'annotation, sur les questionnements linguistiques et sur les modèles. Pour rester dans cette idée de comparaison et pour coller au maximum aux données en français, j'ai choisi de travailler avec des modèles entraînés sur des corpus de textes d'apprenants (Learner corpora). Chacun des trois modèles de baseline en anglais a été entraîné et testé sur un même corpus (c'est d'ailleurs de tous les tests le corpus qui permet systématiquement les meilleurs résultats). Il s'agit du FCE dataset, que j'ai déjà présenté dans la section 2.3.2 du chapitre « cadre théorique ».

Ce corpus est composé de phrases en anglais écrites par des apprenants (non natifs), qui ont passé l'examen Cambridge First Certificate in English (FCE), ce qui se rapproche fortement du type de textes de la ressource en français. Bien que le corpus soit annoté en types d'erreurs, les chercheurs anglophones se contentent de n'utiliser que la version du corpus destinée à la détection d'erreurs (classes binaires : correct ou incorrect). De fait, si l'objectif principal est de pouvoir comparer mes modèles à ceux réalisés pour l'anglais, rien ne m'oblige à traiter les quelque 80 balises d'erreurs proposées par Nicholls (2003). De plus, la typologie d'erreurs utilisées⁷ dans le dataset de Cambridge ne ressemble en aucun cas à la typologie d'erreurs présentée ici, donc là encore, peu de comparaisons possibles. La version pour l'erreur détection est disponible au format CONLL, c'est-à-dire, un fichier TSV avec une colonne avec les tokens puis une colonne avec la balise (*i* pour incorrect et *c* pour correct). Ce format a en plus le point positif d'être extrêmement facile à gérer, les séquences sont déjà tokenisées et pré traitées (pas d'encodage extraordinaire, pas de ponctuation ou de signes superflus, etc.) et les balises sont facilement récupérables pour les encoder sous un format numérique accepté par les modèles (binaires dans ce cas-ci). Pour toutes ces raisons, il était intéressant de faire correspondre mon corpus de test et d'entraînement au format CONLL du FCE dataset.

Nous aurions aussi pu décider de ne pas ancrer le projet dans la lignée de ceux déjà réalisés pour l'anglais et de récolter des données supplémentaires par le biais des données synthétiques. Néanmoins, la génération de données synthétiques reste extrêmement chronophage. En effet, il faut développer des algorithmes capables de générer des erreurs de tous les types présents dans la typologie, et ce, dans des proportions similaires à celles présentes dans les textes originaux. Or, sans annotation, il est impossible de réaliser les travaux d'analyse d'erreurs nécessaire à cette procédure. N'étant pas plus avantageuse, j'ai décidé de continuer la campagne d'annotation, aussi longue soit-

⁷ typologie d'erreurs basée sur une combinaison d'un type factuel d'erreur (ex. Quelque chose manque, quelque chose est en trop, besoin d'un remplacement, ...) avec un POS et d'une précision si l'erreur provient d'un problème grammatical (nom comptables, accords, flexions, ...). Toutes ces combinaisons rendent la typologie extrêmement complexe à gérer et à comprendre en plus de créer un nombre gigantesque (ingérable) de balises possibles.
(Nicholls, 2003)

elle, pour récolter suffisamment de données annotées pour l'étape d'entraînement et de test des modèles.

Pour les données en français, il a fallu dans un premier temps les exporter de la plateforme d'annotation INCEpTION. C'est là que j'ai rencontré le plus de difficultés. En effet, le format d'export « Web-anno-tsv », à première vue similaire (et compatible) au format CONLL est en réalité extrêmement difficile à gérer. Pour normaliser les données, modifier la tokenisation présente et gérer les tags particuliers, j'ai décidé d'aborder le problème comme un problème scientifique. Ma méthode de résolution et les codes qui l'ont permise sont disponibles en annexe (Annexe 8 - 11). Le résultat de cette méthode passe par des annotations au format CONLL donc un TSV (ou CSV) avec une colonne Token et une colonne Label. Cette normalisation me permet de traiter de la même manière les données en anglais et les données en français. De ces données, trois versions distinctes sont exploitables (les trois existent en toutes lettres ou déjà au format numérique accepté par les modèles) :

- La version composée de 23 labels différents (toutes les balises de la typologie + la balise ERR),
- La version composée uniquement de 5 labels (les quatre catégories linguistiques + la balise ERR),
- Et la version binaire (*c* pour correct et *i* pour incorrect).

Pour finir, une fois les modèles entraînés sur les données décrites ci-dessus, il sera temps d'analyser les performances de ces modèles et de discuter les résultats. Pour l'évaluation sur les données de test j'ai décidé de me concentrer sur la précision et le *rappel*, mais également sur la mesure F1 (moyenne harmonique de la précision et du rappel) afin de toujours garder un point de comparaison constant avec d'autres études. Les comparaisons des performances entre mes différents tests seront au centre de mon attention, il s'agira de savoir quel type de corpus atteint les meilleures performances et pourquoi (comparaison anglais-français, mais aussi entre les différents types de données en français elles-mêmes). Mais la comparaison se fera entre les résultats d'un même modèle sur un même corpus, pour les annotations multi-classes, j'effectuerai une analyse des erreurs du modèle. C'est-à-dire qu'il faudra aller regarder les performances du modèle pour la prédiction de chaudes classes et vérifier entre autres dans quel cas le modèle a le plus de difficultés. Il aurait été intéressant d'ouvrir le champ de l'expérimentation avec l'ajout éventuellement de données annexes comme les étiquetages morpho-syntaxique ou la L1 de chaque texte pour « nourrir » le modèle. Bien qu'intéressant, je préfère me concentrer à produire des outils satisfaisants, seuls et une ressource annotée de manière effective et fiable avant de me lancer dans de plus amples expériences. Et canaliser mon analyse pour être capable

de comprendre, d'expliquer et éventuellement résoudre (au moins sur papier) les lacunes de mes modèles. Je laisserai donc la place à la recherche pour s'emparer de mes propositions et les faire évoluer vers d'autres sphères. Toutes ces considérations ainsi que les résultats seront discutées dans le chapitre suivant.

Voilà le plan d'attaque de ce projet que j'ai déployé autour de deux axes principaux, à l'image du projet de mémoire en lui-même. En plus des choix méthodologiques qui me sont propres, j'ai eu la chance d'être encadrée par mes maîtres et promoteurs de stage pour ancrer le mémoire dans un projet de plus grande ampleur. Les objectifs de ce projet sont donc d'une part la création d'une ressource annotée en erreur. Ce qui implique, par conséquent, la création d'une typologie d'erreurs précise, utile et documentée, d'un guide d'annotation complet et la gestion de la campagne d'annotation au global. Le maître mot de ces étapes était ici la rigueur qui a permis de proposer une ressource documentée et solide pour l'analyse d'erreurs, la pédagogie et évidemment les expérimentations techniques. D'autre part, comme le projet de CENTAL s'articule autour d'outils d'annotation et mon mémoire également, j'ai procédé à l'élaboration de deux modèles de deep learning qui seront entraînés et évalués entre autres sur la ressource créée lors de la campagne d'annotation. Aussi bien la baseline que le modèle utilisant BERT seront discutés et évalués dans le chapitre suivant. Chapitre qui comportera aussi les résultats de la campagne d'annotation (accord inter-annotateurs, pourcentage d'accord, Gold standard) ainsi qu'une courte analyse des erreurs. Toutes ces informations me permettront, supportées par la théorie et les choix méthodologiques, de pointer les points forts de mon approche, mais aussi les limitations et éventuelles erreurs commises lors des étapes clés de ce projet.

Chapitre 4 : Résultats.

4. Résultats :

Ce chapitre présente les résultats des expérimentations décrites dans le chapitre précédent. Ces résultats concerneront dans un premier temps le corpus et la campagne d'annotation, une approche plus traditionnellement linguistique permettant une légère analyse des erreurs et une compréhension des données. Dans un second temps, nous entrerons dans les considérations techniques avec les résultats des modèles testés ainsi qu'un tour d'horizon des modèles cités dans la littérature et dans le chapitre 2. En fin, il sera temps d'exploiter au mieux ces résultats en les interprétant au regard des données, de la baseline et des performances trouvées dans la littérature. Pour conclure par un passage en revue des différentes limitations et difficultés rencontrées lors de l'élaboration de ces résultats.

4.1 Description des données.

Pour la création de la ressource annotée en erreur et l'entraînement des modèles de Machine Learning, nous avons eu recours à une campagne d'annotation. Afin de nourrir cette campagne et d'entraîner/tester les modèles, nous avons, grâce au projet conjoint entre le CENTAL et France éducation international, accès à un corpus de 6500 textes écrits par des apprenants lors des épreuves du TCF (test de connaissance du français). Il est intéressant, avant plus amples analyses et utilisations de ce corpus, de regarder de plus près les textes le composant. Pour y parvenir, je base ici mes affirmations sur le travail du docteur Rodrigo Wilkens et des chercheurs CENTAL en général qui se sont chargés, dans le cadre de leur projet (évaluation automatique de textes d'apprenants) de répertorier/analyser avec précision le contenu du corpus. Ce sont eux aussi qui se sont chargés de faire la division des 6500 textes en différents Batch et d'en attribuer un certain nombre à chacun des annotateurs de la campagne. C'est donc majoritairement leur travail (issu d'un rapport datant d'avril 2023) que je vais présenter ici (François et Wilkens, 2023).

Dans les 6500 textes des passations des épreuves écrites du TCF, il y a évidemment différents niveaux de connaissance de la langue, évaluée grâce au système de niveaux du CECR allant de A1 à C2. Est aussi représentée dans le corpus, une large variété de langues maternelles (L1) des apprenants passant le test. Ces informations, bien que non utilisées dans la présente étude, sont cruciales pour la ressource en elle-même et sa validité académique. En effet, ce sont des informations utiles et importantes pour toute analyse plus approfondie du corpus et elles peuvent s'avérer nécessaires dans les phases d'amélioration des modèles entraînés via la ressource (Rei et Yannakoudakis, 2017).

Les principaux niveaux représentés dans le corpus sont bien évidemment les niveaux intermédiaires allant de A2 à C1 compris. Les niveaux les plus faibles et les plus forts sont également représentés, mais dans des proportions bien moindres. Ces écarts de proportion sont, d'une part, cohérents avec le besoin qu'ont les apprenants de niveaux intermédiaires de s'exercer pour atteindre des niveaux tels que C1 et C2. Et d'autre part, ils sont alignés sur le fait que les apprenants de niveau A1 ne sont pas suffisamment compétents pour rédiger des textes de longueur suffisante tout en étant compréhensibles et analysables par des lecteurs externes (les annotateurs en l'occurrence). Il est clair que cette « sur-représentation » des niveaux intermédiaires ne pose fondamentalement pas de problèmes dans le cadre de la présente étude, surtout quand on sait que les études d'analyses d'erreurs classiques et les travaux sur les erreurs des FLE sont à l'heure actuelle réalisés majoritairement sur ce type d'apprenants.

Niveau	% de textes
A1	8,79
A2	19,44
B1	21,52
B2	20,97
C1	20,81
C2	8,47

(Tableau 3) répartition des textes dans le corpus FEI
Par niveaux du CECR

Pour ce qui est de la répartition des L1 dans le corpus, les deux langues les plus représentées sont l'anglais et l'arabe, mais il faut aussi prendre en compte la représentation du français comme langue « usuelle ». D'autres langues sont également légèrement surreprésentées, ces dernières sont fort heureusement, assez différentes les unes des autres avec des langues romanes comme le portugais/l'espagnol, slaves comme le Russe ou encore berbère comme le kabyle. Ces langues plus représentées constituent ensemble la moitié des textes du corpus, l'autre moitié est un mélange d'autres langues à proportion variables, mais bien moindre. Encore une fois ici, la chance est d'avoir le corpus le plus général et représentatif possible afin d'ouvrir la porte par la suite à des études et des analyses plus spécifiques (liées par exemple à des besoins pédagogiques).

Langue	% de textes
Anglais	12,51
Arabe	12,89
Français	12,73
Portugais	4,63
Espagnol	4,63
Kabyle	4,55
Russe	4,46
Autres	43,6

(Tableau 4) répartition des textes dans le corpus FEI
Par langues « usuelle », L1.

Le corpus ainsi réparti a par la suite été divisé en Batch (sous-parties) de manière aléatoire, la plus représentative possible. Ces sous-parties sont nécessaires pour la bonne gestion de la campagne d'annotation. Le docteur Rodrigo Wilkens s'est chargé de diviser les textes en documents contenant 10 textes chacun et de répartir ces documents entre les annotateurs. C'est ainsi qu'à peu près deux fois 100 textes (dix documents de 10 +1 texte et neuf documents de 10) ont été attribués aux différentes phases d'entraînement des annotateurs et donc au Gold standard. Puis dans la première phase de la campagne, 770 textes (77 documents de 10 textes) ont été attribués aux quatre annotateurs ayant continué le projet. Ce qui fait pour la première phase d'annotation un total maximum de 3080 textes, soit la moitié du corpus.

Étant donné la vitesse d'annotation des annotateurs qui tourne autour de 200 textes maximums par mois, seuls 2350 textes (Gold standard compris) ont pu être annotés à ce stade : juin 2023. Comme mentionné dans les chapitres précédents, la campagne d'annotation est un outil redoutablement utile par la précision et la qualité des annotations humaines, mais atrocement lent et peu performant. Il a donc fallu faire un choix et diminuer les espérances quant à la taille du corpus d'entraînement et de test des modèles. Ces choix seront évidemment discutés plus loin dans le chapitre au regard des résultats desdits modèles.

Dans ce chapitre, je ferai donc référence à différents jeux de textes (Batch de documents) selon les besoins et les stades d'avancement du projet. En premier lieu, le Gold standard qui fait référence aux 191 textes annotés par tous les annotateurs et par moi-même et qui sert majoritairement à l'évaluation de la campagne. Ensuite, le deuxième Batch composé des 191 textes annotés par mes soins du Gold standard agrémenté de 220 textes par annotatrices (annotés lors de la première phase d'annotation solo de la campagne, de mars à avril 2023). Ce Batch 2 est donc composé d'approximativement 1070

textes $((220 \times 4) + 19)$. Enfin, le dernier Batch, celui servant à l'entraînement des modèles et à leur évaluation, est composé des 1070 textes du Batch 2 augmenté de deux autres phases d'annotation. D'abord la deuxième phase d'annotation solo, 100 textes supplémentaires par annotatrices qui a pris fin début mai 2023, puis la troisième phase, encore en cours. Cette dernière phase étant encore en cours, j'ai évidemment décidé d'en extraire à ce jour tous les documents exploitables et annotés pour élever le total de textes aux 2350 mentionnés plus haut. Ce qui nous amène à un total de documents annoté par quatre annotateurs tous issus d'études en langues et lettres (communication, traduction ou littérature) donc aidés du guide totalement apte à annoter les erreurs d'apprenants. La répartition des niveaux de langues (CECR) par textes est visible en annexe (Annexe 18).

4.2 Résultats de la campagne d'annotation.

La première salve d'analyse et de résultats présentée ici sera celle qui concerne la ressource et sa création. D'abord l'analyse du corpus et de la répartition des erreurs selon les textes qui permettra de mettre en lumière certaines caractéristiques du corpus et d'évaluer de manière préliminaire la disparité entre les annotateurs. Ensuite, il sera temps d'analyser précisément les performances et la qualité de la campagne d'annotation en calculant de diverses manières l'accord entre les annotateurs et le Gold standard. Ces deux aspects sont évidemment cruciaux pour assurer la qualité de la ressource d'un point de vue scientifique (utilité, failles, exploitabilité) et également d'un point de vue pédagogique, la confiance des résultats étant un point de résistance des enseignants envers l'ALAO. En outre, ces analyses sont un passage obligé pour assurer les meilleurs résultats possibles lors de l'évaluation des modèles. Ils permettront d'analyser ces résultats avec un regard informé et d'apporter des pistes de solution.

4.2.1 Analyse du corpus.

Sur les 6500 textes annotés décrits plus haut dans la section, 220 textes par annotateurs ont été échantillonnés en plus des 191 documents du Gold standard afin de réaliser une analyse plus approfondie des erreurs dans le corpus. Cette analyse purement linguistique permet d'établir la viabilité de la ressource présentée ici, et ce, par rapport aux autres travaux dans le domaine. L'analyse d'erreurs réalisée est plutôt sommaire et ne repose que sur des statistiques descriptives (fréquence des balises en l'occurrence). Dans ce cas-ci, les balises par token sont une variable qualitative nominale où chaque balise serait une catégorie de cette variable catégorielle. C'est en ce sens que je vais les analyser à plusieurs niveaux. De ces 1071 textes,

j'ai échantillonné 100 textes (20 par annotateurs + 20 du Gold standard) afin de faire une première analyse globale des erreurs trouvées dans les textes. Ensuite, il est intéressant de comparer les résultats par annotateurs. Pour y arriver, les 200 textes restants par annotateurs et 180 textes restants pour le Gold standard seront analysés. Cette analyse me permettra entre autres de comparer les performances de chaque annotateur et d'évaluer aussi le degré de subjectivité de la tâche. L'analyse statistique présentée a été entièrement réalisée à l'aide du logiciel « IBM SPSS Statistics (Statistical Package for the Social Sciences) » qui est un logiciel largement utilisé pour l'analyse statistique et la manipulation de données. Enfin, de ces analyses factuelles, je me permettrai d'extrapoler d'éventuelles explications linguistiques et de regarder de plus près des catégories linguistiques pour placer la ressource dans la lignée des travaux d'analyses d'erreurs comme décrits dans la première section de ce projet

a. Analyse générale d'un échantillon.

Le tableau (Tableau 5), présenté ici, représente les fréquences des balises dans l'échantillon représentatif de 100 documents. Ces documents comptabilisent un total de 13486 tokens dont 80,2 % sont corrects (balise « _ ») c'est-à-dire 10812 tokens ne contenant aucune erreur ni grammaticale ni orthographique. Les balises manquantes entre le tableau et la typologie ne sont simplement pas répertoriées dans l'échantillon.

	Balise			
	Fréquence	Pourcentage	Pourcentage valide	Pourcentage cumulé
Valide	637	4,7	4,7	4,7
_	10812	80,2	80,2	84,9
*	5	,0	,0	84,9
ERR	24	,2	,2	85,1
L-CASSE	86	,6	,6	85,7
L-DIA	315	2,3	2,3	88,1
L-GRAPHIE	317	2,4	2,4	90,4
L-MOT-I	217	1,6	1,6	92,0
L-MOT-N	45	,3	,3	92,4
M-CMP	61	,5	,5	92,8
M-EUP	56	,4	,4	93,2
M-F-GENRE	106	,8	,8	94,0
M-F-NOMBRE	90	,7	,7	94,7
M-F-PRONOM	20	,1	,1	94,8
M-V-AUX	3	,0	,0	94,9
M-V-CONJ	85	,6	,6	95,5
M-V-MODE	63	,5	,5	96,0
M-V-PRON	16	,1	,1	96,1
M-V-TEMPS	17	,1	,1	96,2
S-COH	5	,0	,0	96,2
S-MAN	150	1,1	1,1	97,4
S-ORD	64	,5	,5	97,8
S-PCT	242	1,8	1,8	99,6
S-REP	3	,0	,0	99,7
S-SUP	47	,3	,3	100,0
Total	13486	100,0	100,0	

(Tableau 5) sortie SPSS, tableau des fréquences des balises trouvées par token.

Hors balises correctes, on constate que les balises L-DIA et L-GRAPHIE (erreurs de diacritique et de graphie/de forme) sont les plus fréquentes dans le corpus avec 11,8 % des balises d'erreurs pour L-DIA et 11,85 % pour L-GRAPHIE. La balise des erreurs de ponctuation score aussi une fréquence relativement élevée avec 9 % des balises pour la balise S-PCT. Sans m'avancer dans une explication linguistique, l'explication principale pour les trois catégories les plus représentées semble pencher vers le fait qu'elles englobent différents cas de figures et différents types d'erreurs. Là où d'autres ne sont utilisés que dans des cas très restreints. En revanche, les types d'erreurs les moins fréquents sont les erreurs concernant les auxiliaires (M-V-AUX) ainsi que la balise représentant les répétitions erronées (S-REP) et la balise de cohérence (articulation logique du texte, S-COH). Ici encore, une première piste d'explication réside dans le fait que ces balises touchent à des erreurs très spécifiques n'existant que dans des situations bien précises, par conséquent moins fréquentes.

b. Analyse des annotations par annotateur et comparaison.

La question est de savoir maintenant si d'une part les échantillons par annotateurs sont équivalents, si chacun des échantillons sont représentatifs des mêmes phénomènes, en d'autres termes si le corpus au global est annoté avec suffisamment de robustesse pour être comparé aux autres études d'analyse d'erreurs d'apprenants. Mais elle sera aussi d'évaluer si la tâche ne contiendrait pas une part de subjectivité. En effet, nous avons établi que la norme et que les erreurs étaient des éléments constitutifs du français écrit bien complexe à délimiter. Ces contours flous rendent donc la tâche d'annotation elle-même floue et subjective. Aussi précis que soit le guide, certains annotateurs auront peut-être une sensibilité moindre au concept d'erreur d'orthographe. D'autre part, il faut tenir compte du fait que l'échantillonnage des textes par annotateurs peut lui aussi influencer les balises et leurs fréquences. C'est pourquoi les tableaux présentés par la suite (Tableaux 6–10) représentent la répartition des balises par token, pour chacun des annotateurs distinctivement et pour le Gold standard (annoté par mes soins). Ces résultats seront, ensuite, discutés et comparés à l'échantillon analysé plus tôt (point a.).

Balise				
	Fréquence	Pourcentage	Pourcentage valide	Pourcentage cumulé
Valide	793	4,0	4,0	4,0
—	16460	82,8	82,8	86,8
*	7	,0	,0	86,8
ERR	10	,1	,1	86,8
L-CASSE	145	,7	,7	87,6
L-DIA	422	2,1	2,1	89,7
L-GRAPHIE	406	2,0	2,0	91,7
L-MOT-I	256	1,3	1,3	93,0
L-MOT-N	79	,4	,4	93,4
M-CMP	141	,7	,7	94,1
M-EUP	34	,2	,2	94,3
M-F-GENRE	142	,7	,7	95,0
M-F-NOMBRE	104	,5	,5	95,5
M-F-PRONOM	24	,1	,1	95,7
M-V-AUX	4	,0	,0	95,7
M-V-CONJ	115	,6	,6	96,2
M-V-MODE	58	,3	,3	96,5
M-V-TEMPS	60	,3	,3	96,8
M-V-VOIX	3	,0	,0	96,9
S-COH	16	,1	,1	96,9
S-MAN	212	1,1	1,1	98,0
S-ORD	57	,3	,3	98,3
S-PCT	242	1,2	1,2	99,5
S-REP	4	,0	,0	99,5
S-SUP	94	,5	,5	100,0
Total	19888	100,0	100,0	

(Tableau 6) sortie SPSS, tableau des fréquences des balises trouvées par token pour les 180 textes du Gold standard.

Balise				
	Fréquence	Pourcentage	Pourcentage valide	Pourcentage cumulé
Valide	750	4,1	4,1	4,1
—	14368	78,0	78,0	82,1
*	1	,0	,0	82,1
ERR	58	,3	,3	82,4
L-CASSE	173	,9	,9	83,4
L-DIA	450	2,4	2,4	85,8
L-GRAPHIE	437	2,4	2,4	88,2
L-MOT-I	302	1,6	1,6	89,8
L-MOT-N	66	,4	,4	90,2
M-CMP	119	,6	,6	90,8
M-EUP	38	,2	,2	91,0
M-F-GENRE	165	,9	,9	91,9
M-F-NOMBRE	142	,8	,8	92,7
M-F-PRONOM	28	,2	,2	92,8
M-V-AUX	8	,0	,0	92,9
M-V-CONJ	108	,6	,6	93,5
M-V-MODE	79	,4	,4	93,9
M-V-PRON	13	,1	,1	94,0
M-V-TEMPS	57	,3	,3	94,3
M-V-VOIX	2	,0	,0	94,3
S-COH	25	,1	,1	94,4
S-MAN	289	1,6	1,6	96,0
S-ORD	110	,6	,6	96,6
S-PCT	498	2,7	2,7	99,3
S-REP	23	,1	,1	99,4
S-SUP	105	,6	,6	100,0
Total	18414	100,0	100,0	

(Tableau 7) sortie SPSS, tableau des fréquences des balises trouvées par token pour les 200 textes Annotateur 1.

Balise				
	Fréquence	Pourcentage	Pourcentage valide	Pourcentage cumulé
Valide	1239	4,7	4,7	4,7
—	22176	84,4	84,4	89,1
*	1	,0	,0	89,1
ERR	25	,1	,1	89,2
L-CASSE	119	,5	,5	89,7
L-DIA	546	2,1	2,1	91,8
L-GRAPHIE	423	1,6	1,6	93,4
L-MOT-I	223	,8	,8	94,2
L-MOT-N	33	,1	,1	94,4
M-CMP	97	,4	,4	94,7
M-EUP	61	,2	,2	95,0
M-F-GENRE	160	,6	,6	95,6
M-F-NOMBRE	134	,5	,5	96,1
M-F-PRONOM	21	,1	,1	96,2
M-V-AUX	9	,0	,0	96,2
M-V-CONJ	184	,7	,7	96,9
M-V-MODE	103	,4	,4	97,3
M-V-PRON	23	,1	,1	97,4
M-V-TEMPS	61	,2	,2	97,6
S-COH	38	,1	,1	97,7
S-MAN	152	,6	,6	98,3
S-ORD	66	,3	,3	98,6
S-PCT	314	1,2	1,2	99,8
S-REP	9	,0	,0	99,8
S-SUP	51	,2	,2	100,0
Total	26268	100,0	100,0	

(Tableau 8) sortie SPSS, tableau des fréquences des balises trouvées par token pour les 200 textes Annotateur 2.

Balise				
	Fréquence	Pourcentage	Pourcentage valide	Pourcentage cumulé
Valide	1401	4,9	4,9	4,9
—	22893	80,2	80,2	85,1
*	2	,0	,0	85,1
ERR	30	,1	,1	85,2
L-CASSE	149	,5	,5	85,8
L-DIA	702	2,5	2,5	88,2
L-GRAPHIE	453	1,6	1,6	89,8
L-MOT-I	367	1,3	1,3	91,1
L-MOT-N	83	,3	,3	91,4
M-CMP	116	,4	,4	91,8
M-EUP	199	,7	,7	92,5
M-F-GENRE	178	,6	,6	93,1
M-F-NOMBRE	244	,9	,9	94,0
M-F-PRONOM	60	,2	,2	94,2
M-V-AUX	20	,1	,1	94,3
M-V-CONJ	177	,6	,6	94,9
M-V-MODE	185	,6	,6	95,5
M-V-PRON	33	,1	,1	95,6
M-V-TEMPS	66	,2	,2	95,9
M-V-VOIX	7	,0	,0	95,9
S-COH	5	,0	,0	95,9
S-MAN	266	,9	,9	96,8
S-ORD	82	,3	,3	97,1
S-PCT	678	2,4	2,4	99,5
S-REP	35	,1	,1	99,6
S-SUP	105	,4	,4	100,0
Total	28536	100,0	100,0	

(Tableau 9) sortie SPSS, tableau des fréquences des balises trouvées par token pour les 200 textes Annotateur 3

Balise				
	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1517	4,7	4,7	4,7
—	26167	81,5	81,5	86,2
*	8	,0	,0	86,3
L-CASSE	73	,2	,2	86,5
L-DIA	625	1,9	1,9	88,4
L-GRAPHIE	599	1,9	1,9	90,3
L-MOT-I	528	1,6	1,6	92,0
L-MOT-N	39	,1	,1	92,1
M-CMP	172	,5	,5	92,6
M-EUP	200	,6	,6	93,2
M-F-GENRE	199	,6	,6	93,9
M-F-NOMBRE	283	,9	,9	94,7
M-F-PRONOM	53	,2	,2	94,9
M-V-AUX	8	,0	,0	94,9
M-V-CONJ	177	,6	,6	95,5
M-V-MODE	111	,3	,3	95,8
M-V-PRON	25	,1	,1	95,9
M-V-TEMPS	55	,2	,2	96,1
M-V-VOIX	7	,0	,0	96,1
S-COH	12	,0	,0	96,1
S-MAN	381	1,2	1,2	97,3
S-ORD	87	,3	,3	97,6
S-PCT	581	1,8	1,8	99,4
S-REP	24	,1	,1	99,5
S-SUP	168	,5	,5	100,0
Total	32099	100,0	100,0	

(Tableau 10) sortie SPSS, tableau des fréquences des balises trouvées par token pour les 200 textes Annotateur 4.

Globalement, les balises les plus utilisées sont, comme pour l'échantillon général, celles concernant les erreurs de forme/graphie et de signes diacritiques avec une moyenne (hors mots corrects) de 11,92 % pour la balise L-DIA :

Gold standard : 11,1 %

Annotateur 1 : 12,3 %

Annotateur 2 : 13,3%

Annotateur 3 : 12%

Annotateur 4 : 10,5 %

Et une moyenne (hors mots corrects) de 10,2 % pour la balise L-GRAPHIE :

Gold standard : 11,1 %

Annotateur 1 : 11,8 %

Annotateur 2 : 10 %

Annotateur 3 : 8 %

Annotateur 4 : 10 %

La balise des erreurs de ponctuation connaît plus de variabilité entre les annotateurs. En effet on remarque que chez les annotateurs 1 et 2 les erreurs de ponctuation ne concernent que 7,7%

des erreurs alors qu'on atteint des scores de 12 % et de 12,3 % pour l'annotateur 3 et pour le Gold standard (Tableaux 6 et 9). Néanmoins, ce type d'erreurs reste supérieur à la moyenne des autres erreurs. Cette variabilité n'est pas une fin en soi, car elle est probablement due à la sensibilité qu'ont les annotateurs à voir et à comprendre les « règles » normes grammaticales qui entourent ce point précis de l'orthographe française. En effet, beaucoup de natifs ignorent, tout bonnement, l'existence de règles régissant, l'emploi de certains signes de ponctuation. L'annotateur 3 et moi-même sommes vraisemblablement plus aptes à comprendre et à repérer ce type d'erreur bien spécifique.

Les erreurs les moins représentées dans ces échantillons sont en somme relativement similaires à celles trouvées dans l'échantillon global. Encore une fois, ce sont les erreurs concernant les auxiliaires qui sont les moins fréquentes, tout comme les erreurs de répétitions et de cohérence. La balise M-V-VOIX est par ailleurs complètement absente des échantillons de l'annotateur 2 ainsi que de l'échantillon global (Tableaux 7 et 5). Ce qui n'est pas étonnant, l'utilisation de la voix passive est relativement rare en langue française, et ce, d'autant plus pour des apprenants de niveaux faibles.

Pour ce qui est du pourcentage de token correct dans chaque échantillon, il est également équivalent au pourcentage de l'échantillon global. L'annotateur ayant le plus haut taux de token non annoté (donc considéré comme correct) est l'annotateur 2 (Tableau 8) avec 84,4 %. Or le Gold standard n'enregistre que 78 % de formes correctes. Une différence de plus de 6 % est certes considérable, mais peut être expliquée par la nature des textes dans l'échantillon. En effet, les textes de niveaux forts (B2/C1) contiennent à priori moins d'erreurs que ceux de niveau faible. Aussi, cela peut s'expliquer par une subjectivité de la tâche et une interprétation plus ou moins libre du guide d'annotation (de la part de l'annotateur 2) qui entraîne une sensibilité moins accrue au concept même d'erreurs.

Il est cependant intéressant de noter que le nombre de tokens par échantillon de même taille est différent.

Annotateur 1 : 19888 Tokens pour 200 documents

Annotateur 2 : 26268 Tokens pour 200 documents

Annotateur 3 : 28526 Tokens pour 200 documents

Annotateur 4 : 32099 Tokens pour 200 documents

En somme, du point de vue de la répartition des balises dans les échantillons de chaque annotateur, malgré quelques variations minimes, on retrouve une cohérence et une harmonisation des valeurs. Cette harmonie permet d'assurer une représentativité certaine du corpus, ce dernier pourra donc être considéré comme valable pour discuter les erreurs des

apprenants. Il sera évidemment nécessaire de comparer cette intuition de stabilité aux résultats des calculs accord inter-annotateurs obtenus lors de la confection du Gold standard. À noter que cet accord a été réalisé par étapes avant que les annotateurs soient capables d'utiliser pleinement le guide et d'identifier correctement les erreurs de la typologie. Avant de réaliser cette comparaison, je vais m'atteler à analyser plus en profondeur la répartition des erreurs dans les textes d'apprenants. Pour introduire le corpus créé dans la lignée des études déjà produites à ce sujet et pour en vérifier la teneur.

c. Analyse et extrapolation des résultats par balises et par catégories linguistiques.

Une fois les erreurs classées par catégorie linguistique (lexique, syntaxe et morphologie), il est aisé d'établir les erreurs les plus fréquentes dans nos échantillons de textes d'apprenants. De plus, si on considère l'échantillon comme représentatif du corpus, il est possible de généraliser ces découvertes aux apprenants du corpus entier. Ce travail de généralisation et d'analyse d'erreurs a évidemment déjà été réalisé par le passé. Et dans l'optique de rendre la présente analyse la plus représentative possible, il est nécessaire de confronter ces découvertes à celles des travaux précédents, dans le but, évidemment, de trouver de possibles explications pédaogo-linguistiques aux erreurs du corpus. Par catégorie, les erreurs les plus fréquentes sont :

Erreurs lexicales : L-GRAPHIE, L-DIA et L-MOT-I/N

Erreurs morphologiques : M-F-NOMBRE/GRENRE, M-V-CONJ/MODE, M-CMP

Erreurs syntaxiques : S-PCT, S-MAN

Comme point de comparaison, le corpus ayant servi à tester les algorithmes de détection et de diagnostic d'erreurs du projet FreeText (L'haire et Vandeventer Faltin, 2003 :33), par exemple, est annoté avec une typologie d'erreur similaire à celle utilisée dans ce projet-ci, bien que plus simpliste. Tout comme dans notre corpus, les erreurs les plus retrouvées sont celle relevant du choix de mots (appelées dans le projet FreeText erreurs d'homophonie) ainsi que les erreurs de flexions en genre notamment. Parmi les erreurs les moins représentées dans ce corpus, on compte les erreurs d'auxiliaire et de voix verbales (introuvables dans le corpus FreeText) encore une fois, comme dans l'échantillon de notre corpus.

Selon Delen Karaagaç (2012 :177), les erreurs des apprenants auraient pour origine deux hypothèses dont la plus fréquente serait les erreurs par analogie, voyez ici une généralisation de règle ou de coutumes connues par l'apprenant. Cela se traduirait alors par une grosse fréquence d'erreurs de choix de mot (L-MOT-I) ou de flexions erronées (qu'elles soient verbales : M-V-CONJ, ou non : M-F-GENRE/NOMBRE). Ou encore une propension à violer

les règles de complémentations (S-CMP) ou à omettre des mots nécessaires à la construction d'une phrase grammaticale (S-MAN). Cette hypothèse corrobore entièrement les résultats quantitatifs des fréquences d'erreurs dans le corpus de texte.

Néanmoins, il faut reconnaître que pour réaliser une analyse poussée d'erreurs sur base de l'échantillon présenté ici, il est nécessaire d'avoir de plus amples informations sur le niveau des apprenants (Annexe 18) et leur langue première (indisponible à ce stade). En effet, la justification des fréquences rapportées réside probablement dans l'influence de la langue et dans la difficulté de certaines règles grammaticales et orthographiques. Et ce pour la simple et bonne raison que certains niveaux, sur-représentés ou sous-représentés, influencent la présence ou non de formes plus ou moins complexes liées au parcours d'apprentissage et aux paliers des niveaux du CECR. Néanmoins, l'analyse des erreurs n'étant pas le sujet principal du projet, je me réserve le droit de ne pas poursuivre cette analyse plus amplement afin de laisser place à d'autres résultats. Cette analyse sommaire est suffisante pour démontrer la robustesse du corpus (une fois couplée aux calculs d'accord inter-annotateurs).

L'étude d'H. E. Alsadag (2021) est représentative du fait que la L1 influence aussi considérablement la répartition des erreurs dans un corpus. Ici, tous les apprenants étant nativement arabophones avec un niveau équivalent en français, la répartition des erreurs se fait suivant les mêmes catégories linguistiques que celles présentées dans le chapitre précédent (syntaxe, lexique et morphologie). Là où les apprenants arabophones ont 65 % de leurs erreurs dans la catégorie des erreurs syntaxiques et seulement 20 % d'erreurs lexicales. La répartition des erreurs dans notre échantillon est plutôt de l'ordre de 36 % d'erreurs lexicales contre 19% seulement d'erreurs syntaxiques. Cela peut évidemment s'expliquer par le fait que les apprenants de la présente étude sont d'horizons linguistiques différents et donc les faiblesses des uns sont compensées quantitativement par les points forts des autres.

Malgré cette analyse plutôt sommaire, certaines intuitions au sujet des annotateurs et de la précision de la campagne d'annotation menée se sont dégagées. Il est maintenant temps de comparer ces intuitions avec les résultats quantitatifs de la campagne elle-même. En effet, pour rendre la ressource viable, il faut vérifier la qualité de l'annotation plus amplement que par une analyse qualitative des erreurs détectées dans des échantillons différents. La section suivante va discuter de la précision avec laquelle les annotateurs ont réalisé leur tâche au regard de mon annotation (étant linguiste et créatrice du guide, je fais office de Gold standard).

4.2.2 Évaluation de l'annotation.

Comme je l'ai expliqué dans la section méthodologie, la création du Gold Standard et donc les premières phases de la campagne d'annotation ont eu lieu en trois étapes. Les quatre premières annotatrices ont d'abord eu accès à 51 textes pour exercer leur aptitude à annoter et leur compréhension du guide. Après une vérification minutieuse de ma part (vérification qui a permis de constituer le Batch 1 du Gold Standard), j'ai identifié les failles de leurs annotations. Une réunion de clarification plus tard, elles se sont lancées dans l'annotation de 50 textes (Batch 2) puis de 90 textes (Batch 3). Ce sont ces 191 textes vérifiés/annotés par mes soins qui constituent le Gold standard global. Pour évaluer l'efficacité de la campagne d'annotation et extrapoler les performances de ces premières annotatrices aux annotateurs qui se sont ajoutés à la tâche (par faute de temps, ces derniers ne sont pas discutés dans la présente étude), je vais dans un premier temps analyser l'accord global entre les 5 annotatrices et le Gold standard (Tableaux 11 - 13). Puis j'analyserai les performances de chaque annotatrice par rapport au Gold (Tableaux 14 - 16). Enfin, j'évaluerai si les intuitions de la section précédente sur les performances des annotateurs sont fondées ou non. Cette analyse se fera par le biais d'une formule d'accord inter-annotateur robuste : l'alpha de Krippendorff. Le choix s'est porté sur cette mesure, car elle tient compte des différentes catégories d'erreurs, corrige les effets du hasard, et peut identifier les biais systématiques entre les annotateurs. Étant donné que les annotations d'erreurs peuvent varier en complexité et en interprétation, l'alpha de Krippendorff offre une approche robuste pour quantifier l'accord global tout en prenant en compte les nuances inhérentes à cette tâche multi-classe.

a. Accord global.

Pour avoir une vision à 360° des performances de la campagne d'annotation, j'ai traité les données des 3 Batch du Gold Standard de trois manières différentes. Dans un premier temps j'ai conservé les 23 balises de la typologie (22 + la balise pour les mots corrects). Dans un deuxième temps, j'ai décidé de rendre les balises binaires (1 si le mot est annoté, 0 si le mot est correct). Puis dans un troisième temps, j'ai divisé les balises en 4 grandes catégories (erreurs lexicales, syntaxiques et morphologiques + une catégorie pour la balise ERR) et une cinquième pour les mots non annotés évidemment. L'accord inter-annotateur (via l'alpha de Krippendorff) est alors calculé selon les trois traitements différents : 23 balises (Tableau 11), binaire (Tableau 12), 5 balises (Tableau 13). Cette division me permettra d'identifier quelles sont les principales difficultés de la tâche.

Voici les résultats pour la version avec 23 balises :

Global	Krippendorff's Alpha hors Gold	Krippendorff's Alpha avec Gold
Batch 1 (51 textes)	0.40	0.44
Batch 2 (50 textes)	0.60	0.62
Batch 3 (90 textes)	0.58	0.61

(Tableau 11) résultat du calcul d'accord (Krippendorff's Alpha) globaux entre les 5 annotateurs [et le Gold standard] Annotation avec 23 balises d'erreurs.

Pour tous, j'ai calculé l'accord avec une formule de l'Alpha de Krippendorff (Annexe 12). On constate une nette amélioration entre un score de 40 % entre les annotatrices lors de la première phase de 51 textes et des scores avoisinants les 60 % pour les deux phases suivantes. La stagnation entre ces deux dernières phases s'explique par le fait qu'elles ont été organisées dans un laps de temps très court sans réel feedback de ma part aux annotateurs entre les deux phases. On peut donc imaginer qu'avec l'expérience les annotateurs ont encore amélioré leurs performances. De plus, la seconde colonne qui représente le calcul de l'accord entre les cinq annotateurs et le Gold standard on constate que cet accord est nettement plus haut que l'accord entre les annotatrices seulement. L'hypothèse la plus probable est que certaines annotatrices se rapprochent plus du Gold standard que d'autres à ce stade.

Voici les résultats pour la version binaire :

Binaire	Krippendorff's Alpha hors Gold	Krippendorff's Alpha avec Gold
Batch 1	0.37	0.41
Batch 2	0.57	0.59
Batch 3	0.56	0.59

(Tableau 12) résultat du calcul d'accord (Krippendorff's Alpha) globaux entre les 5 annotateurs [et le Gold standard] Annotation avec 2 balises (correct, incorrect).

Ici, et cela devrait se confirmer avec l'analyse approfondie de chaque annotateur, les performances sont moins bonnes. Ce qui s'explique probablement par le fait que la localisation de l'erreur est une tâche complexe pour des annotateurs non-experts de la langue. Par ailleurs, l'amélioration entre le Batch 1 et les Batches 2/3 se confirme quand même avec encore une fois

des scores avoisinant un score de 60% entre les annotatrices et le Gold lors de l'annotation des 90 derniers documents.

Enfin, voici les résultats pour la version à cinq classes :

Classes	Krippendorff's Alpha hors Gold	Krippendorff's Alpha avec Gold
Batch 1	0.39	0.43
Batch 2	0.59	0.61
Batch 3	0.57	0.61

(Tableau 13) résultat du calcul d'accord (Krippendorff's Alpha) globaux entre les 5 annotateurs [et le Gold standard]
Annotation avec 5 balises d'erreurs (catégories linguistiques).

Nettement plus proches des premiers résultats, cette version (Tableau 13) confirme que la tâche la plus complexe semble bien être la détection des erreurs et le placement des balises plutôt que l'utilisation de ces dernières. Nous arrivons néanmoins pour les deux dernières phases, après clarification donc avec des annotateurs plus à l'aise avec la tâche, à des scores se rapprochant des 60 %. Pour une tâche qui prête à ce point à débat avec des balises nombreuses et complexes, il est évident que ces résultats sont à ce stade satisfaisants.

b. Accord entre les annotateurs et le Gold Standard.

Ici encore, j'ai utilisé les trois versions des trois Batches d'annotations afin d'affiner l'analyse des résultats par annotateurs. De plus, j'ai calculé le pourcentage d'accords entre chacun des annotateurs et le Gold Standard grâce au tableau de contingence (Annexe 13) (Tableaux 14 – 16). Cet ajout me permettra de confirmer ou non mes intuitions de la section précédente et par la même occasion éventuellement prédire l'évolution des performances des annotateurs au fur et à mesure des annotations. Toujours, pour garder une cohérence avec les scores précédents, le choix du calcul se porte sur l'alpha de Krippendorff, bien que celui-ci ne soit pas le plus utilisé dans le cas de calculs deux à deux.

Résultats de la version globale :

	BATCH 1		BATCH 2		BATCH 3	
	Krippendorff's Alpha	%	Krippendorff's Alpha	%	Krippendorff's Alpha	%
Annotateur 1	0.57	90.48	0.66	90.67	0.74	94.4
Annotateur 2	0.56	90.72	0.67	91.1	0.66	93.2
Annotateur 3	0.60	91.0	0.65	90.19	0.68	93.0
Annotateur 4	0.35	89.0	0.65	90.15	0.64	92.3
Annotateur 5	0.47	85.9	0.68	89.7	0.63	92.1

(Tableau 14) résultat du calcul d'accord (Krippendorff's Alpha) deux à deux entre les 5 annotateurs et le Gold standard
Annotation avec 23 balises d'erreurs.

Pour cette version encore, on constate une nette amélioration entre le premier Batch et les deux suivants. Les scores anormalement bas de l'annotateur 4 au premier Batch de 51 textes s'expliquent par le fait que cette dernière n'arrivait pas à manipuler correctement la plateforme d'annotation. Ce qui a engendré plus de la moitié des 51 documents non annotés par ses soins, d'où les maximums 35 % de score entre ses annotations et le Gold. L'annotateur ayant la plus nette amélioration est l'annotateur 1 avec un gain de plus de 15 % de score entre le Batch 1 et le dernier jeu de 90 documents. Pour les autres, on reste dans une fourchette plus qu'acceptable entre 63 % et 68 % entre les annotateurs et le Gold Standard.

Résultats de la version binaire :

	BTACH 1		BATCH 2		BATCH 3	
	Krippendorff's Alpha	%	Krippendorff's Alpha	%	Krippendorff's Alpha	%
Annotateur 1	0.55	90.48	0.64	90.67	0.72	94.4
Annotateur 2	0.54	90.72	0.65	91.1	0.64	93.2
Annotateur 3	0.58	91.0	0.63	90.19	0.66	93.0
Annotateur 4	0.33	89.0	0.62	90.15	0.62	92.3
Annotateur 5	0.43	85.9	0.65	89.7	0.61	92.1

(Tableau 15) résultat du calcul d'accord (Krippendorff's Alpha) deux à deux entre les 5 annotateurs et le Gold standard Annotation avec 2 balises (correct, incorrect).

Dans cette version-ci, comme on le constatait dans l'accord global, il y a à nouveau une nette régression des résultats. On perd, par exemple, pour l'annotateur 1 2% pour l'accord maximum. Cela confirme donc la complexité de la tâche d'identification des erreurs et du placement des balises.

Résultats pour la version avec cinq classes :

	BATCH 1		BATCH 2		BATCH 3	
	Krippendorff's Alpha	%	Krippendorff's Alpha	%	Krippendorff's Alpha	%
Annotateur 1	0.57	90.48	0.66	90.67	0.73	94.4
Annotateur 2	0.56	90.72	0.67	91.1	0.66	93.2
Annotateur 3	0.60	91.0	0.65	90.19	0.68	93.0
Annotateur 4	0.35	89.0	0.64	90.15	0.64	92.3
Annotateur 5	0.46	85.9	0.67	89.7	0.63	92.1

(Tableau 16) résultat du calcul d'accord (Krippendorff's Alpha) deux à deux entre les 5 annotateurs et le Gold standard Annotation avec 5 balises d'erreurs (catégories linguistiques).

Au risque de me répéter, on remarque que malgré les disparités des scores de la première phase d'annotation (51 documents) il apparaît que chacune des annotatrices effleure à terme les 70% de score avec le Gold standard (Tableau 16, batch 2 et 3). Même pour celles ayant semblé beaucoup de mal aussi bien avec les balises/le guide et la plateforme d'annotation, la pratique et l'entraînement permet d'obtenir des résultats tout à fait corrects. De plus, si on jette un œil aux pourcentages d'accords, ils correspondent totalement aux pourcentages de token annotés « corrects » autrement dit non annotés analysés précédemment. Cette proximité dans les résultats entre les données du Gold standard et les données annotées ensuite par les annotatrices seules me permet d'avancer confiante dans la suite des annotations. En effet, il est aisé d'en déduire que l'amélioration que les annotateurs ont connu entre la première phase et les deux suivantes (due sans nul doute à une adaptation à la plateforme et une meilleure maîtrise de la typologie) n'aura de cesse de se confirmer au fil de leur pratique de l'annotation.

Ces constats sont plus qu'encourageants, car ils permettent de confirmer l'utilité et l'efficacité de telles campagnes même si ces dernières sont chronophages, difficiles à mettre en place et onéreuses. Cela est d'autant plus encourageant que la tâche d'annotation d'erreurs, combinée à une typologie complète à visée pédagogique et informative précise, nécessite beaucoup de connaissances et d'expertise. De plus, la qualité des annotations de la campagne est un soulagement avant de passer à l'étape de modélisation étant donné que ce sont ces annotations et donc cette ressource qui va permettre d'entraîner les modèles. Et bien entendu, ce sont ces modèles une fois entraînés et satisfaisants qui seront l'avenir de l'annotation d'erreurs d'apprenants FLE, qui permettront d'éviter d'avoir recours à de telles campagnes pour la création de ressources ou d'outils pédagogiques.

4.3 Résultats de la campagne de modélisation.

4.3.1 Résultats dans la littérature, « baseline » préliminaire.

Comme décrit dans le chapitre 3 : méthodologie, les projets de détection d’erreurs pour des « learner corpora » anglophones sont le point de départ de ce projet-ci. L’enjeu principal est évidemment de les adapter au français d’une part, puis d’adapter ces modèles francophones à une annotation multi-classes (autre que de la détection binaire). Pour traiter efficacement les résultats des modèles testés dans le cadre de ce mémoire, il est important de revenir sur les résultats de ces études antérieures et de les considérer en quelque sorte comme «baseline » ou comme point de comparaison avec les résultats des modèles francophones. De plus, avec les études plus récentes menées dans la détection d’erreur pour le français, il est nécessaire de reprendre ici les éléments de performance principaux pour faciliter la discussion des résultats du modèle et de la réelle base line, testés lors de ce projet.

Les éléments sélectionnés sont plusieurs projets qui intègrent des modèles basiques Bi-LSTM et BERT afin de se rapprocher au maximum des implémentations francophones présentées dans le chapitre 3. Ce sont par ailleurs ces mêmes projets qui ont guidé la création de nos modèles. D’autre part, un élément crucial dans le choix de ces études a été le corpus d’entraînement de ces dernières (pour l’anglais du moins). Chacune d’entre elles sont entraînées, entre autres, sur le FCE dataset, une base de données qui se rapproche fortement du type de textes présents dans notre propre ressource. Ce sont donc les résultats concernant ce corpus d’entraînement et de test (FCE dataset) que je présenterai, toujours en ce qui concerne les études anglaises.

Aussi, chaque étude présentée ici offre des résultats uniquement sur la détection d’erreurs, c’est-à-dire la classification binaire. C’est d’ailleurs une des raisons qui m’ont poussée à réaliser une baseline propre au projet permettant d’avoir un véritable ordre de grandeur pour analyser les résultats du BERT.

Chacun des résultats présentés sont directement tirés des articles et sont, si besoin, augmentés du calcul (réalisé par mes soins) de la mesure F1 non présente dans les différents articles, mais utile d’un point de vue comparaison. La F1 est donc calculée selon la formule :
$$[F\text{-mesure} = 2 * (\text{Précision} * \text{Rappel}) / (\text{Précision} + \text{Rappel})].$$

Pour les résultats du modèle Bi-LSTM, ceux-ci sont basés sur le modèle implémenté par Rei et Yannakoudakis en 2016 (Tableau 17, ligne 2). Le Bi-LSTM présenté est un Bi-LSTM simple

pour la détection d'erreur (et non le Deep Bi-LSTM). Une autre implémentation de ce Bi-LSTM présenté en 2017 par Rei (Tableau 17, ligne 3) sera aussi présentée comme baseline étant donné que cette implémentation aussi a servi de base pour l'implémentation du Bi-LSTM adapté au français.

	P	R	Mesure F0,5	Mesure F1
Rei et al. (2016)	46,1	28,5	46,0	35,27
Rei (2017)	55,38	25,34	44,56	34,48

(Tableau 17) Résultats, précision, rappel, F1 et F0,5 des modèles Bi-LSTM de 2016 et 2017 sur le FCE dataset.

Pour ce qui est du modèle BERT, le principal projet ayant influencé la création du modèle testé ici est celui de Kaneko et Komachi, 2019 (Tableau 18) qui proposent comme baseline prometteuse la version de BERT « BERT_base » pré-entraînée puis finetunée sur le FCE dataset. Ce modèle BERT est celui qui se rapproche le plus de celui implémenté depuis Hugging face pour l'annotation francophone « CAMEMBERT_base ».

	P	R	Mesure F0,5	Mesure F1
Kaneko et al. (2019)	69,80	37,37	59,47	48,6

(Tableau 18) Résultats, précision, rappel, F1 et F0,5 des modèles BERT de 2019 sur le FCE dataset.

Ces résultats sont évidemment donnés à titre informatif afin d'avoir un ordre de grandeur dans lequel s'insérer. Tout comme la baseline SVM du projet, ces données servent uniquement à expliquer et analyser les résultats du modèle BERT sous d'autres prismes. Il est important de pointer le fait que les corpus, bien que similaires, sont extrêmement différents (langue, taille, composition) et que les méthodes d'annotation dudit corpus le sont tout autant (balises moins complexes, seuls deux annotateurs). Le choix de présenter ici les modèles Bi-LSTM se repose principalement sur le fait que les architectures de deep-learning diffèrent complètement de l'apprentissage de surface type SVM. Y ajouter les résultats d'autres modèles d'apprentissage profond sur le même corpus permet encore une fois d'ajouter un prisme de comparaison.

4.3.2 Résultats pour le français.

a. Base line :

Pour avoir un élément de comparaison adéquat, nous avons décidé d'implémenter un algorithme d'apprentissage de surface qui n'est autre qu'une machine à vecteurs de supports (Tableau 19), très réputés il y a encore quelques années (avant l'avènement du deep-learning). Les résultats une fois l'entraînement effectué sur le corpus de données binaires permettront de situer le niveau 0 à améliorer par le modèle testé par la suite. Le corpus d'entraînement et de test, comme je l'ai décrit plus tôt, est donc identique à celui pour l'entraînement et le test du BERT pour la détection d'erreurs dont les résultats seront présentés plus loin.

	P	R	Mesure F0,5	Mesure F1
SVM binaire	79,61	35,37	60,64	48,95

(Tableau 19) Résultats, précision, rappel et F-mesure
Pour le modèle SVM entraîné à la classification binaire.

Il est évident que les résultats pour la détection binaire, à l'image de la littérature, seront éventuellement inférieurs que ceux fournis par un modèle Transformers aussi basique que celui présenté ici. Néanmoins, ces résultats seront nécessaires lors de la discussion et de l'extrapolation vers la classification multi-classe ainsi que lors de la réflexion autour des limitations du modèle BERT.

b. BERT et classification multi-classe :

Pour ce qui est de la classification grâce à Bert, comme dit dans la méthodologie, nous avons procédé en plusieurs étapes. Les résultats clés de ces étapes sont donc sommairement présentés ici et seront discutés plus en profondeur dans la section suivante. Dans un premier temps nous nous penchons sur les résultats de la classification binaire (correct, incorrect) plus communément appelé détection d'erreurs. Cette tâche sera le fondement de nos discussions étant donné que c'est celle qui a été la plus éprouvée dans la littérature. Nous aurons donc beaucoup de points de comparaisons possibles. Dans un second temps il faudra tester les implémentations des classifieurs BERT multi-classes (5 et 24 balises d'erreurs) afin d'évaluer la robustesse des modèles Transformers pour des tâches plus complexes que la détection d'erreurs.

Les prédictions sur le jeu de données de tests pour la détection d'erreurs ont été réalisées après différentes epoch d'entraînement afin de toujours identifier le score le plus haut et donc l'entraînement le plus efficace (Tableau 20). Cette étape fait évidemment partie de l'optimisation du modèle afin de minimiser la perte sur les données de test et d'obtenir les meilleurs résultats possibles).

Epochs	Mesure F1
1	0.778433437424544
2	0.7964773920600285
3	0.7831644804487858
4	0.7873016512851222
5	0.8060265862368937
6	0.7840476214844887
7	0.7859029404039605
8	0.7959454904882273
9	0.7959454904882273

(Tableau 20) évolution de la mesure F1, tâche de classification Binaire avec CamemBert base.

Contrairement aux expérimentations pour l'optimisation du Learning rate (section 3.3, Fig.5), la neuvième epoch n'est pas la plus performante sur le jeu de validation, mais il s'agit bien de la cinquième comme notre intuition initiale l'indiquait. Pour tous les prochains résultats, j'afficherai ceux après cinq epoch et ceux après neuf epoch pour plus de précision et pour ne rater aucun bon résultat. En ce qui concerne la classification binaire (Tableau 21), on outrepassa de beaucoup les résultats du SVM (30% de plus pour le BERT) grâce à l'apprentissage profond.

	Classes	Précision	Rappel	Mesure F1
5 epoch	0	89	85	87
	1	49	59	53
9 epoch	0	90	83	86
	1	47	61	53

(Tableau 21) résultats après 5 et 9 epoch de la classification binaire Avec pour le corpus de test 34436 instance de la classe 0 (correct) Et 8532 instances de la classe 1 (incorrect).

Grâce à ces mesures, on calcule donc aisément, après 5 epoch (le plus performant) une mesure de précision globale de 81% et une mesure de rappel globale de 80% [calculées à partir de la moyenne pondérée des scores par classes afin de minimiser l'impact du nombre d'instances]. Ce qui est tout à fait compétitif, mais nous y reviendrons dans la section suivante.

Le cœur de l'étude menée ici est évidemment le diagnostic et l'annotation plus fine des erreurs dans les textes FLE. Pour ce faire, nous avons envisagé deux approches, la première est une classification en 5 grandes catégories, celle-ci est la plus prometteuse aux vu de résultats de la classification binaire et de sa proximité en termes de nombre de classes. La seconde est la classification multi-classe avec toutes les balises du corpus, cette dernière est très expérimentale.

Dans l'ordre du Tableau 22), la première classe (0) correspond toujours à la balise « correcte », la deuxième (1) concerne la balise « ERR » qui correspond aux séquences incompréhensibles. Ensuite, la troisième (2) englobe les erreurs « Lexicales », la quatrième (3) les erreurs « Morphologiques » et, enfin, la cinquième (4) les erreurs « Syntaxiques ». La présentation des résultats se fera donc avec des mesures de précision, rappel et F1 par balise afin de faciliter l'analyse de ces derniers et d'exploiter les failles du modèle. De plus, le tableau 22 reprend également le nombre d'erreurs de chaque classe présentes dans le corpus de test (ce nombre est globalement représentatif de la répartition des erreurs dans les corpus de validation et d'entraînement, cf. 4.2.1).

	Classes	Précision	Rappel	Mesure F1	Support
5 epoch	0	89	66	76	34440
	1	0	0	0	28
	2	46	66	54	4041
	3	9	38	14	2480
	4	8	3	4	1879
9 epoch	0	90	53	67	34440
	1	0	0	0	28
	2	46	68	45	4041
	3	9	51	15	2480
	4	8	5	6	1879

(Tableau 22) résultats par balises après 5 et 9 epoch
De la classification multi-classe (5 balises),
Mesures de précision, de rappel et de F1.

De plus, ces mesures m'ont permis de calculer (par le biais d'une moyenne pondérée) les performances du modèle pour la classification en 5 catégories. Les résultats sont d'ailleurs assez encourageants, mais nous y reviendrons dans la section suivante. Le modèle score au global des mesures de F1, de précision et de rappel :

F1= 72 % après 5 epoch, 71 % après 9 epoch,

P = 71 % après 5 epoch, 72 % après 9 epoch,

R = 68 % après 5 epoch, 72 % après 9 epoch.

Pour en finir avec la présentation des résultats de l'annotation automatique par apprentissage profond, il me reste à présenter les résultats les plus expérimentaux. Il s'agit évidemment de la classification multi-classe à 24 balises, soit une par type d'erreurs. Ces résultats sont bien sûr les moins impressionnants pour de nombreuses raisons qui seront abordées dans les deux sections suivantes. Néanmoins, il est intéressant pour l'avenir de visualiser les performances du modèle par balise et en parallèle de voir le nombre d'instances de chaque balise dans le corpus de test (Tableau 23). Les résultats présentés ici ne sont que ceux après 5 epoch, les résultats après 9 epoch sont visibles en annexe (Annexe 17)

		Précision	Rappel	Mesure F1	Support
5 epoch	0	88	2	3	34440
	1	0	0	0	28
	2	15	4	6	375
	3	3	73	6	112
	4	29	8	12	1533
	5	17	75	28	1246
	6	2	0	1	782
	7	2	17	3	105
	8	3	93	5	253
	9	1	1	1	374
	10	5	51	10	361
	11	3	7	5	451
	12	0	50	1	30
	13	0	0	0	344
	14	0	1	1	360
	15	0	0	0	72
	16	1	13	1	115
	17	0	0	0	8
	18	1	4	1	51
	19	4	1	1	593
	20	0	0	0	210
	21	3	26	5	881
	22	1	23	1	53
	23	4	3	3	191

(Tableau 23) résultats de la classification multi-classe
à 24 balises après 5 epoch,
mesures de précision, rappel et F1.

Comme les résultats par balise l'indiquent, les mesures d'évaluation globales sont, il est clair, moins impressionnantes que les précédentes. Grâce à la moyenne pondérée des mesures par balises, on obtient après 5 epoch : précision = 14,5%, rappel = 2% et F1 = 3,6%.

4.3.3 Discussion.

Quand on met en relation les résultats de la classification binaire avec BERT sur le corpus anglais du FCE dataset (Kaneko et al, 2019) et ceux réalisés grâce à la version francophone de BERT que nous avons réalisée ici, on constate une nette progression des résultats avec une mesure F1 qui supprime celle de l'anglais de plus de 30%. Idem pour la baseline réalisée sur notre corpus francophone, le SVM est, comme on l'imaginait, surpassé par les performances de l'architecture Transformers.

Ces résultats presque incroyables de la classification binaire sont évidemment très encourageants pour les travaux en français, mais doivent tout de même être explicités et analysés afin de rendre compte de leur robustesse.

Pour une tâche moins similaire, le projet d'annotation d'erreurs de coordination mené en 2023, entre autres par Laura Noreskal, peut aussi servir de point de comparaison intéressant quant aux performances d'une architecture BERT sur de la classification binaire en français. Cette comparaison est toutefois inexacte via plusieurs facteurs, la nature et la taille du corpus, le type d'erreurs analysé, mais aussi le niveau d'annotation (token, fragments) ainsi que l'annotation humaine (création des corpus).

Néanmoins, avec les informations fournies dans le papier il est aisé de calculer (de la même manière que pour nos résultats) la mesure F1 de la tâche de classification effectuée avec CamemBert (de loin le plus performant). Cette mesure avoisine les 78% sur un corpus déséquilibré contenant plus d'instances erronées que d'instances correctes. Pour la classification binaire avec CamemBert, le déséquilibre du corpus (en faveur d'une classe ou de l'autre) semble donc être un facteur favorisant les résultats.

En effet, les résultats moins probants pour les tâches anglophones pourraient donc venir du fait que le FCE corpus soit équilibré. Nous verrons par la suite que ce déséquilibre ne sera pas toujours un facteur favorisant la réussite de la classification.

Un autre résultat prometteur est celui de la classification en cinq classes, c'est-à-dire la détection des erreurs en fonction de la catégorie d'erreurs linguistiques auxquelles elles appartiennent. Proche, par le nombre de classes, de la classification binaire, on obtient de ce fait des résultats similaires pour certaines catégories. En effet, dans les deux cas, le classifié n'a aucun mal à reconnaître la classe prédominante, c'est-à-dire les tokens estampillés « correct ». Ce phénomène est explicable statistiquement par le ratio mot correctement orthographié-mot mal orthographié que le corpus de test (tout comme les autres corpus) contient.

Cette explication fonctionne aussi pour le résultat décevant de la détection de la classe 1 (les mots portant la balise « ERR »). Cette balise n'est présente que 28 fois sur les quelque 43 000 tokens du corpus de test. Si l'on revient d'ailleurs à l'analyse des erreurs dans le corpus (section 4.2.1) cette classe ne représente en moyenne que 0,2 % des erreurs du corpus. N'ayant pas suffisamment d'éléments pour affiner l'annotation de cette balise, le modèle, aussi performant soit-il, tendra toujours vers une mauvaise détection de ce type d'erreur.

Cette réflexion se confirme aussi avec les autres résultats de la classification en cinq catégories, les catégories les moins représentées dans le corpus sont moins bien classifiées par le modèle. Encore une fois dans la section d'analyse du corpus (4.2.1), les erreurs lexicales étant plus fréquentes chez les apprenants, avec pour les textes Gold standard 8 % des tokens portant cette balise. C'est-à-dire qu'approximativement 36 % des erreurs contenues dans le Gold standard sont des erreurs lexicales contre seulement 18 % qui sont des erreurs morphologiques.

Même si le déséquilibre de fréquences des erreurs dans le corpus est logique et attendu d'un point de vue linguistique, on s'imagine facilement qu'un apprenant commettra plus d'erreurs d'un type plutôt qu'un autre. Il représente un problème à surmonter d'un point de vue technique, et ce, d'autant plus si on augmente encore le nombre de catégories.

L'hypothèse du déséquilibre comme facteur limitant est alors vérifiée avec la classification multi-label selon les 24 types d'erreurs. Il est clair que ces résultats sont très peu probants et qu'une augmentation des instances d'erreurs pourrait les améliorer (en témoigne le nombre d'instances support de chaque classe). De plus, la classification en 24 types d'erreurs étant très fine et très précise, il était évident que des modèles basiques comme ceux appliqués aux données binaires ne seraient pas assez robustes. Toujours chez Noreksal et al. (2023) on retrouve des résultats aussi décevants au moment de passer à une classification multi-classe. Bien que dans ce projet, le nombre de classes n'augmente pas aussi considérablement, CamemBert base semble bel et bien avoir du mal à gérer ce genre de classification fine. Au niveau de ses performances, notre modèle est par exemple incapable de reconnaître les erreurs de voix verbale (17), ce qui peut être expliqué par l'analyse du corpus (4.2.1), mais aussi les erreurs concernant les verbes pronominaux (15) et les erreurs de conjugaison (13). Par ailleurs, il a moins de mal à reconnaître et à classer les erreurs de graphie et de forme (5).

Néanmoins, à ce stade, la seule hypothèse du nombre d'instances d'erreurs disponibles dans le corpus est trop faible pour expliquer un tel manque de résultats. Et malheureusement, les architectures Transformers, ou plus largement l'apprentissage profond, ont comme grand défaut cet effet « boîte noire » qui rend difficile l'explication de certains comportements, problématiques ou non. Pour tenter d'expliquer les résultats produits dans nos

expérimentations, mais aussi d'ouvrir la voie vers de possibles améliorations, dans la section suivante, je vais aborder différentes pistes. Ces pistes sont à elles seules de possibles solutions à creuser et à exploiter et donc de possibles failles du modèle testé ici.

4.4 Limitations et place pour la suite.

Dernière section de ce chapitre, cette section sera consacrée à l'exploration des possibles limitations qui pourraient avoir contribué négativement aux résultats de la classification automatique et plus largement aux facteurs ayant entraîné des difficultés inutiles dans ce projet. Ces limitations sont divisées en trois pistes distinctes et seront soutenues autant que faire se peut par des études et des articles scientifiques. En fin, cette section se clôturera par des perspectives d'amélioration future et par un rapide tour d'horizon des autres approches possibles au problème de la classification des erreurs d'orthographe.

4.4.1 Première piste : la typologie.

La typologie d'erreurs des apprenants du FLE, utilisée ici, a été initialement créée dans un but de fidélité linguistique et d'exploitabilité pédagogique/ scientifique. L'intérêt premier était donc d'avoir une typologie de confiance, argumentée et justifiée, pour qu'elle puisse rendre un corpus utilisable par les chercheurs en linguistique et par les enseignants du FLE. Le second but était de concilier cet objectif avec les avantages considérables qu'apportent la TAL et l'ELAO. Je pense donc que cette typologie et les choix méthodologiques qui en ont découlé sont possiblement des facteurs limitant les résultats en termes de classification automatique.

En premier lieu, le choix de la campagne d'annotation couplé à la subjectivité de la tâche et à la complexité de la classification est à remettre en cause. Les résultats des accords inter-annotateurs en témoignent, l'homogénéité des résultats n'est pas garantie à 100 %. Il conviendrait alors pour assurer aux modèles des corpus d'entrées plus fiables (avec des classifications plus unanimes sur lesquelles se baser) d'assurer moins de subjectivité lors de la tâche d'annotation manuelle. Cela pourrait se faire soit en minimisant le nombre d'annotateurs pour permettre moins de variabilité, bien que cette solution apporte aussi son lot de problèmes, soit en explicitant davantage la typologie (par le guide) ou en formant mieux les annotateurs à la tâche, soit encore en simplifiant la typologie sans trahir évidemment l'envie d'exactitude linguistique de cette dernière.

En deuxième lieu, il est bon de revenir sur la complexité de la typologie. Cette dernière contient d’abord beaucoup de classes précises et ne se rapportant qu’à certains contextes grammaticaux très précis. Ensuite, nous avons fait le choix méthodologique de permettre de superposer les balises, afin de coller à la réalité là où les apprenants commettent plusieurs erreurs sur un même mot. En enfin, la classification se fait au niveau du token, or lors de l’annotation manuelle, certaines erreurs englobent des constructions contenant plus d’un token, ce qui rend encore une fois la classification complexe. Le problème de la superposition sera abordé plus en profondeur dans la suite de la section. Par ailleurs, plus le nombre de classes est élevé, plus il est complexe pour le modèle d’en distinguer les différences, les paramètres à prendre en compte par le modèle pour les distinguer étant multiples et entremêlés.

En troisième lieu, cette complexité de la typologie couplée à des textes réels des apprenants FLE ont donné lieu à deux problèmes de taille : le déséquilibre et le manque de représentation de certaines classes. Le déséquilibre sera abordé plus amplement dans le point suivant. Mais il est clair que la typologie précise que nous avons créée ne permet pas de représenter de manière homogène chacune des classes. En effet, si une classe a très peu d'exemples d'apprentissage, le modèle peut avoir du mal à apprendre des représentations précises pour cette classe. Cela peut entraîner des prédictions moins précises ou une tendance au sur apprentissage sur les classes majoritaires qui ont plus d'exemples d'apprentissage.

4.4.2 Deuxième piste : la taille du corpus.

Dans un article publié en 2020 au sujet de sa création CamemBert, Martin L. (et al., 2020B) avance différentes pistes qui pourraient favoriser ou défavoriser les performances de ce modèle. La première est, bien évidemment, la taille des données disponibles pour le français. Les données en français étant rares et la création de corpus conséquente, coûteuse et lente, la raison la plus évidente de résultats insatisfaisants est évidemment le nombre de données consacrées à l’entraînement, la validation et le test du modèle (même si celui-ci est pré entraîné sur de nombreuses données éclectiques comme c’est le cas pour BERT en général).

Augmenter le nombre de données pourrait donc permettre simplement d’améliorer les résultats. Mais cela pourrait aussi remédier aux problèmes d’hétérogénéité entre les classes et de déséquilibre flagrant. Car, il est vrai, le problème avec un déséquilibre de classe montre est que le modèle peut être biaisé vers les classes majoritaires, parce qu’il est incité à prédire avec précision ces classes pour maximiser sa performance globale. Cela peut entraîner des difficultés pour le modèle à généraliser et à prédire correctement les classes minoritaires qui ont moins

d'exemples d'apprentissage. Ce qui pourrait être le cas pour les résultats prometteurs de la classification binaire et à cinq classes.

4.4.3 Troisième piste : les modèles.

Outre les données en elles-même, les résultats de la classification en 23 classes peuvent être dues à une utilisation inadaptée de l'architecture Transformers. Il serait intéressant de repenser, par exemple, le fine-tuning du modèle avec des hyper paramètres plus adaptés à la classification multi-classe en testant d'autres Learning rate, epoch ou en changeant la taille du lot.

Au regard de ce qui se fait pour l'anglais, il peut être bénéfique à l'image de ce que Kaneko et Komachi ont fait en 2019 en modifiant l'architecture même de BERT pour coller à leur étude. Ou alors, comme le suggère Rei et Yannakoudakis en 2017, il pourrait être avantageux de d'ajouter d'autres informations afin de nourrir le modèle et de pallier le problème de différenciation (dimensionnalité et similarité entre les classes). Entre autres informations, il serait bon d'explorer l'ajout d'un étiquetage morpho-syntaxique (POS) ou d'informations sur la langue première de chacun des extraits de texte ou encore le niveau de langue des apprenants. Dans leur papier, ils proposent aussi d'ajouter des informations sur les relations grammaticales pour nourrir encore plus le modèle et lui permettre de contrebalancer la disparité entre les classes.

Pour la disparité toujours, il serait utile de tester d'autres méthodes d'équilibrage des poids accordés à chaque classe, en fonction de leur fréquence, mais aussi éventuellement en fonction de leur importance d'un point de vue pédagogique (erreurs les plus rédhibitoires selon le niveau de langue souhaité par exemple).

Enfin, on pourrait, pour la classification complète, prendre le problème sous un autre prisme et se tourner vers d'autres architectures, plus prometteuses encore que BERT. En effet, l'effervescence autour des méga modèles de langues tels que GPT (3 et plus) atteint aussi la sphère de la correction et de la détection des erreurs d'orthographe. Bon nombre de chercheurs se lancent alors dans l'analyse des performances de telles tâches (détection, annotation ou correction) lorsqu'elles sont demandées à Chat GPT ou à d'autres modèles plus éprouvés. C'est d'ailleurs le cas de Wu et al. qui en 2023 comparent les algorithmes du géant de la correction grammaticale « Grammarly » et son nouveau concurrent Chat GPT. Bien que leur analyse ne traite pas directement de notre sujet, elle permet d'ouvrir les portes à de nouvelles expérimentations afin de maximiser les performances de nos modèles et d'atteindre notre but : avoir une ressource annotée efficacement en erreurs et avoir des outils exploitables pédagogiquement grâce à l'ALAO/ELAO.

Chapitre 5 : Conclusion.

5. Conclusion :

La présente étude avait comme premier la création d'une ressource de textes d'apprenants du FLE annotée en erreurs d'orthographe et de grammaire. Cette ressource se voulait exploitable à la fois pour les études en linguistique et pour les enseignants, dans un contexte pédagogique. Un objectif annexe était également de concilier le monde de la pédagogie avec celui du traitement automatique du langage, en développant des modèles capables de détecter et de classer ces erreurs selon une typologie établie. Le but ultime était d'ouvrir la voie vers des outils pédagogiques ELAO adaptés aux apprenants du français. Le tout sans galvauder la rigueur linguistique nécessaire pour garder la confiance du monde de la pédagogie et des sciences du langage, souvent réticent à avoir recours à de nouvelles technologies réputées pour leur effet « boîte noire » et leur manque de fiabilité. Cela étant dit, le point d'honneur a donc été d'expliquer et de justifier au maximum les choix et partis pris lors de l'élaboration du projet et évidemment, comme tout projet de TAL, de tenter d'atteindre les résultats les plus satisfaisants et les plus proches de la capacité humaine à classer les erreurs. En parallèle de ces objectifs, de nombreuses hypothèses ont été explorées. Au niveau de l'annotation du corpus, l'hypothèse principale était de la mener par le biais d'une campagne d'annotation manuelle, et ce, au regard des difficultés que cette méthode implique et des failles des annotateurs humains. Cette hypothèse se basait majoritairement sur le fait qu'une typologie d'erreurs d'apprenant est, par définition, un set de balise complexe et que les méthodes automatiques nécessitent malheureusement encore un grand nombre de textes annotés (apprentissage supervisé). Au niveau de la modélisation et de l'annotation automatique, l'hypothèse principale était de mettre à l'épreuve les architectures Transformers éprouvées dans la littérature pour la détection d'erreur. Cette mise à l'épreuve consistait alors à confirmer leur robustesse en passant d'une classification binaire à une classification multi-classe de plus en plus complexe.

Le parcours de création de la ressource annotée est passé par plusieurs étapes cruciales. Tout d'abord, la mise en place d'une typologie des erreurs a été entreprise, visant à la rendre fiable autant que possible, afin d'assurer son exploitation pertinente. Cette étape a donc donné lieu à une analyse minutieuse des études ultérieures concernant les erreurs d'orthographe et de grammaire des apprenants du français. Elle a aussi donné vie à une réflexion purement linguistique sur les concepts-mêmes d'erreurs et de norme orthographique, nécessaires pour justifier et soutenir les différents chemins pris par ce projet. Ensuite, une campagne d'annotation a été menée, mobilisant plus de dix annotateurs au total sur une période de plus de six mois. Cette étape, après moultes formations et explications, et la création d'un guide d'annotation, a permis de créer la ressource annotée selon la typologie des erreurs et les balises préalablement définies. Balises qui ont été étudiées au maximum pour coller à la réalité des erreurs et être exploitables d'un point de vue technique. La ressource ainsi

créée a permis de nourrir les modèles décrits ci-après, en étant déclinée en trois versions : la première destinée à la détection d'erreurs (classification binaire), la deuxième reprenant une classification des erreurs en cinq grandes classes (classification externe), et la dernière reprenant les 23 balises de la typologie (objectif final). La ressource qui en a résulté est donc un corpus de 2350 textes annotés par 5 annotateurs différents (moi-même et quatre annotateurs non spécialistes de la langue). Cette ressource se divise elle-même en trois versions, la première réservée à la détection d'erreurs est annotée de manière binaire, la deuxième est annotée en cinq grandes catégories d'erreurs et la dernière est annotée en 23 types d'erreurs présents dans la typologie.

La phase de modélisation a été une étape essentielle pour atteindre nos objectifs. Nous avons commencé par une approche de classification binaire avec une baseline basée sur un algorithme (modèle ML) d'apprentissage de surface, le SVM, puis nous avons exploré l'apprentissage profond. Cette phase a eu lieu en utilisant CamemBERT base, la version francophone du modèle de langue entraîné par Google du nom de BERT. Cette démarche visait à rivaliser avec les projets existants dans la littérature en anglais, en termes de compétitivité, ce surtout pour la détection d'erreurs.

Les résultats obtenus lors de la campagne d'annotation ont été satisfaisants (en termes d'accord inter-annotateur), fournissant une première vision de la répartition des erreurs dans le corpus étudié. Et une ressource de qualité satisfaisante pour passer à la phase de modélisation, bien que, nous y reviendrons, la ressource nécessite encore des améliorations structurelles. Ensuite, les résultats encourageants obtenus avec BERT pour la détection d'erreurs (classification binaire) et l'annotation des erreurs en 5 classes linguistiques démontrent une certaine robustesse du modèle à passer de la détection à la classification des erreurs d'apprenants FLE [F1 = 72 % après 5 epoch d'entraînement]. Ces résultats encourageants sont d'ailleurs bien au-dessus de la lignée de travaux en détections d'erreurs réalisés par le passé en anglais, ce qui nous fait dire que certaines limitations, citées plus loin, ont pu mener à des situations gênantes comme du sur apprentissage. Cependant, il convient aussi de reconnaître que le modèle BERT présente des limitations pour la classification en 23 classes, pour plusieurs raisons possibles. Tout d'abord, la complexité et l'hétérogénéité de la typologie peuvent avoir un impact sur les performances du modèle. En effet, les balises du corpus ne sont pas équilibrées, avec un nombre insuffisant d'éléments pour certaines balises et un excès pour d'autres. La taille globale du corpus, en raison, entre autres, de la lenteur de la campagne d'annotation, peut également avoir joué un rôle dans les résultats obtenus pour cette classification plus fine des erreurs. Enfin, il est possible que le modèle BERT ne soit pas adapté ou optimisé pour classer efficacement les 23 classes d'erreurs difficilement différenciables. C'est-à-dire que pour toutes ces raisons, le

modèle semble incapable de distinguer efficacement des différentes classes et donc de classer les erreurs.

Pour de futures améliorations, il serait recommandé d'enrichir le corpus, de simplifier ou d'optimiser la typologie, et de réduire la subjectivité de la tâche d'annotation pour obtenir un corpus plus homogène, avec un meilleur accord inter-annotateur. Il serait également bénéfique d'optimiser le modèle BERT en jouant avec les hyperparamètres ou en ajoutant des informations supplémentaires telles que le POS, le niveau de langue ou les liens grammaticaux. Par ailleurs, l'exploration de nouveaux modèles plus adaptés que BERT, tels que les super modèles de langues comme GPT-3, qui suscitent actuellement un grand intérêt dans le domaine de la TAL, pourrait également être envisagée.

En conclusion, cette étude a permis de créer une ressource de textes d'apprenants du FLE annotée en erreurs d'orthographe et de grammaire, exploitable pour les études en linguistique et dans un contexte pédagogique. Les résultats obtenus ont été encourageants, bien que des améliorations soient nécessaires pour parvenir à une classification plus précise en 23 classes d'erreurs. Ces perspectives d'améliorations ouvrent des pistes intéressantes pour de futurs travaux de recherche dans le domaine de l'ELAO adaptée aux apprenants du français.

Partie II. Bibliographie et Annexes.

Bibliographie :

Antoniadis, G., 2010. « De l'apport pertinent du TAL pour les systèmes d'ALAO : l'exemple du projet MIRTO », in : *2ème Congrès Mondial de Linguistique Française*, n.150, pp.2211-2223.

Artstein, R., Poesio, M., 2008. « Survey Article: Inter-Coder Agreement for Computational Linguistics », in : *Computational Linguistics*, vol. 34(4), pp. 555-596.

Arrivé, M., 1993. « Reforme l'orthographe ? », col. *Linguistique nouvelle*, Presse universitaire de France, Paris.

Bechichi, R., 2017. « Analyse des erreurs interférentielles dans les productions écrites de FLE », *mémoire de master*, Bouzid, D. (Dir.), Université Larbi Ben M'Hidi.

Bell, S., Yannakoudakis, H., Rei, M., 2019. « Context Is Key : Grammatical Error Detection with Contextual Word Representations », in : *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp.103-115.

Belkebir, R., Habash, N., 2021. "Automatic Error Type Annotation for Arabic », in : *Proceedings of the 25th Conference on Computational Natural Language Learning*, pp.596-606.

Blanche-Benveniste, C., Chervel, A., 1978. « L'orthographe (Nouvelle édition augmentée d'une post-face) », F. Maspero, Paris.

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

Bosc-Miné, C., 2014. « Caractéristiques et fonctions des feed-back dans les apprentissages », in : *L'Année psychologique*, vol.114(2), pp.315-53.

Bourray, M., 2016. « L'erreur dans l'apprentissage de la production écrite en classe de FLE: repérage, traitement et remédiation », in : *FRANCISOLA*, vol.1(2), pp.163-174.

Bryant, C., et al., 2017. « Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction », in : *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp.793-805.

Bryant, C., et al., 2022. « Grammatical Error Correction : A Survey of the State of the Art », in : [arXiv:2211.05166v2](https://arxiv.org/abs/2211.05166v2).

Buisson-Buellet, I., Mout, T., Vernet, S., 2012. « Une réforme de l'orthographe ? Quels positionnements ? », in : Dister, A., Moreau, M-L. (dir). « Réforme de l'orthographe française Craintes, attentes et réactions des citoyens », *Revue de sociolinguistique en ligne*, n.19, pp. 37-51.

Burt, M., et al., 1982. « Language two », Oxford University Press, Oxford.

Câslaru, M. D., 2016, « Système d'annotation du corpus d'apprenants roumains de FLE », in : JHSS : *linguistics, stylistics and translation Studies*, vol.7(2), pp. 101-112.

Catach, N., 1973a. « Que faut-il entendre par système graphique du français ? », in : *Langue française*, n.20. L'orthographe, pp. 30-44.

Catach, N., 1973b. « notion actuelle d'histoire de l'orthographe », in : *Langue française*, n.20. L'orthographe, pp. 11-18.

Chanier, T., 1992. « Perspectives de l'apport de l'EIAO dans l'apprentissage des langues : modélisation de l'apprenant et diagnostic d'erreurs. », in : *Revue de liaison de la recherche en Informatique Cognitive des Organisations*, vol.3(4), pp.25-34.

Chaurand, J., Grosse, A., 2006. « Histoire de la langue et histoire de l'orthographe, entretien recueilli par Honvault-Durocq R. », in : Honvault-Durocq, R. (dir). « L'orthographe en question », col. *Dylalang*, Publication des Université de Rouan et du Havre, Mont-Saint-Aignant, pp. 25-47.

Chervel, A., 1973. « La grammaire traditionnelle et l'orthographe », in : *Langue française*, n.20. L'orthographe, pp. 86-96.

Chodorow, M., et al., 2012. « Problems in Evaluating Grammatical Error Detection Systems », in : *Proceedings of COLING 2012*, pp. 611–628.

Cibois, P., 1991. « Enquêtes d'opinion sur l'orthographe », in : Catach, N., Petitjean, L., Tournier, M. (dir). « Orthographe et société », *Mots*, n.28, pp. 86-98.

Cutumisu, M., Lu, C., 2021. « Integrating Deep Learning into An Automated Feedback Generation System for Automated Essay Scoring », in : *Proceedings of The 14th International Conference on Educational Data Mining (EDM21)*, International Educational Data Mining society, pp.573-579.

Dagneaux, E., Denness, S., Granger, S., 1998. « Computer-Aided Error Analysis », in : *System*, vol.26(2), pp.163-174.

David, J., 2006. « L'orthographe du français et son apprentissage, historique et perspectives », in : Honvault-Durocq, R. (dir). « L'orthographe en question », col. *Dylalang*, Publication des Université de Rouan et du Havre, Mont-Saint-Aignant, pp.169-190.

Delbar, V., et al., 2016. « L'anti-correcteur : outil d'évaluation positive de l'orthographe et de la grammaire », in : *Actes de la conférence conjointe JEP-TALN-RECITAL 2016*. vol.2, pp.333–41.

Delen Karaagaç, N., 2006. « Définition et utilisation de la notion de « faute » et « d'erreur » dans l'enseignement de la langue française », *Ç.Ü. Sosyal Bilimler Enstitüsü Dergisi*, vol.15(1), pp. 125-132.

Delen Karaagaç, N., 2012. « Des erreurs linguistiques et des effets de contexte : analyse des différents types d'erreurs commises par les étudiants turcs », in : *Uluslararası Sosyal Aratırmalar Dergisi The Journal of International Social Research*, vol.5(23), pp. 175-184.

Devlin, J., et al., 2018. « 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*, vol.1, pp. 4171-4186.

Emirkanian, L., et al., 1989. « La correction des erreurs d'orthographe d'usage dans un analyseur morpho-syntaxique de français », in : *Langue française*, n.83, pp. 106-122.

Farid, G., 2012. « La « nouvelle orthographe », 21 ans plus tard », in : *SHS Web of Conferences*, n.1, pp. 2261-2424.

Fayol, M., Foulon, J-M., Pacton, S., 2005. « L'apprentissage de l'orthographe lexicale », in : *Rééducation orthophonique*, n.222, pp. 47-68.

Felice, M., Bryant, C., Briscoe, T., 2016. « Automatic Extraction of Learner Errors in ESL Sentences Using Linguistically Enhanced Alignments », in : *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp.825-835.

Fontanelle, T., 2006. « Les nouveaux outils de correction linguistique de Microsoft », in : *Actes de la 13ème conférence sur le Traitement Automatique des Langues Naturelles*, pp.5-20.

François, T., Wilkens, R., 2023. « FEI report 2: Mar/2022 – April/2023, Appendix B : Corpus ML », pp. 20-45

Gardin, B., 2006a. « Sur quelques composantes sociolinguistiques de l'orthographe française », in : Honvault-Durocq, R. (dir). « L'orthographe en question », col. *Dylalang*, Publication des Université de Rouan et du Havre, Mont-Saint-Aignant, pp.110-114.

Gardin, B., 2006b. « Sur quelques (dys)fonctions de l'orthographe », in : Honvault-Durocq, R. (dir). « L'orthographe en question », col. *Dylalang*, Publication des Université de Rouan et du Havre, Mont-Saint-Aignant, pp. 121-122.

Granger, S., 2003. « Error-tagged Learner Corpora and CALL : A Promising Synergy », in : *CALICO Journal*, vol.20(3), pp.465-480.

Granger, S., Gilquin, G., Meunier, F., 2015. « Learner corpus design and methodology » in : Granger, S., Gilquin, G., Meunier, F. (Eds.), *The Cambridge Handbook of Learner Corpus Research*, Cambridge: Cambridge University Press, pp. 7-182

Granger, S., et al., 2022. « The Louvain error tagging manual version 2.0 », CECL & UCLouvain.

Grouin, C., 2022. « Impact du français inclusif sur les outils du TAL », in : *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles*, pp. 126-135.

Halsted, I., 1975. « Putting error in its place », in : *Journal of basic writing*, vol.1(1), pp. 72-86.

Hanawa, K. Et al., 2021. « Exploring methods for generating Feedback comments for writing learning », in : *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp.9719–9730

H E Alsadag, A., 2021. « Analyse linguistique des erreurs courantes en FLE chez les étudiants à l'université d'Aljufra en Libye », in : *journal of Language Studies*, vol.4(3), pp. 176-184.

Honvault-Durocq, R., 2006. « Les rectifications de l'orthographe et la situation dans les dictionnaires », in : Honvault-Durocq, R. (dir). « L'orthographe en question », col. *Dylalang*, Publication des Université de Rouan et du Havre, Mont-Saint-Aignant, pp. 225-245.

James, C., 1998. « Errors in language learning and use : exploring error analysis », Longman, New-York.

Jaffré, J-P.,1992. « Didactiques de l'orthographe. », Hachette Éducation, Paris.

Jaffré, J-P., 2005. « L'orthographe du français, une exception ? », in : *Le français aujourd'hui*, vol.1(148), pp. 23-31.

Jung, E., 1973. « Causes des fautes d'orthographe », in : *Langue française*, n.20. L'orthographe, pp. 97-110.

Kaneko, M., Sakaizawa, Y., Komachi, M., 2017. « Grammatical Error Detection Using Error- and Grammaticality-Specific Word Embeddings », in : *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, vol.1, pp.40-48.

Kaneko, M., Komachi, M., 2019. « Multi-Head Multi-Layer Attention to Deep Language Representations for Grammatical Error Detection », in : *Computación y Sistemas*, vol.23.

Katoozian, K., 2013. « Comparaison de types d'erreurs orthographiques en FLM et FLE : une revue d'articles », in : *JéTou, 16-17 mai*, pp. 64-69.

Klie, J.-C., et. al, 2018. « The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation » in : *Proceedings of System Demonstrations of the 27th International Conference on Computational Linguistics*.

Korre, K., Chatzipanagiotou, M., Pavlopoulos, J., 2021. « ELERRANT : Automatic Grammatical Error Type Classification for Greek », in : *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pp.708-717.

L'haire, S., Vandeventer Faltin, A., 2003. « Diagnostic d'erreurs dans le projet FreeText », in : *Alsic. Apprentissage des Langues et Systèmes d'Information et de Communication*, vol.6(2), pp.21-37.

La Russa, F., 2021. « Le feedback correctif à l'écrit : techniques, élaboration, acquisition. », *thèse de doctorat*, CAUSA, M. (Dir), Université Bordeaux Montaigne.

Lüdeling, A., et al., 2005. « Multi-Level Error Annotation in Learner Corpora », in : *Proceedings of corpus linguistics, n.1*, pp.14-17.

Lüdeling, A., Hirschmann H., 2015. « Error Annotation Systems », in : Granger, S., Gilquin, G., Meunier, F. (Ed), « *The Cambridge Handbook of Learner Corpus Research* », Cambridge university press, Cambridge, pp.135-158.

Mars, A., 2016. « Le TAL au service des enseignants des langues : mise en oeuvre d'une plate-forme pour l'enseignement du français et de l'arabe, langues étrangères. », *thèse de doctorat*, Antoniadis, G. (dir), Université Grenoble Alpes.

Martin, L., et al., 2020. « CamemBERT: a Tasty French Language Model », in : *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, En ligne*, pp. 7203-7219.

Martin, L., et al., 2020B. « Les modèles de langue contextuels Camembert pour le français : impact de la taille et de l'hétérogénéité des données d'entraînement », in : *JEP-TALN-RECITAL 2020 - 33ème*

Journées d'Études sur la Parole, 27ème Conférence sur le Traitement Automatique des Langues Naturelles, pp. 54-65.

Mitton, R., 1996. « English Spelling and the Computer », *Studies in Language and Linguistics*. Longman, London ; New York.

Mizumoto, T., et al., 2012. « The Effect of Learner Corpus Size in Grammatical Error Correction of ESL Writings », in : *Proceedings of COLING 2012 : Posters*, pp.863-72.

Nicholls D., 2003. « The Cambridge Learner Corpus - error coding and analysis for lexicography and ELT », in : *Proceedings of the Corpus Linguistics 2003 conference*, n.16, pp.572-581.

Noreskal, L., et al., 2023. « Détecter une erreur dans les phrases coordonnées au sein des rédactions universitaires », article présenté dans le cadre de la conférence : *Coria-tlan-2023 : CORIA TALN RECITAL 2023 (du 5 au 9 juin 2023)*.

Pennington, J., Socher, R., Manning, C., 2014. « GloVe : global vector for word representation », in : *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532-1543.

Póltorak, E., 2011. « Question du Feedback en didactique des langues étrangères : quelques réflexions théoriques », in : *Glottodidactica. An International Journal of Applied Linguistics*, n.38, pp.31-42.

Rahmatian, R., 2005. « L'erreur, un facteur de dynamisme dans les processus d'apprentissage des langues étrangères », in : *Plume*, n.2, pp. 105-123.

Ravestein, J., Sensevy, G., 1993. « Statuts de l'erreur dans la relation didactique », *Grand N*, IREM Université J. Fourier, Grenoble.

Rei, M., 2017. « Semi-supervised Multitask Learning for Sequence Labeling », in : *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol.1, pp. 2121-2130.

Rei, M., Yannakoudakis, H., 2016. « Compositional Sequence Labeling Models for Error Detection in Learner Writing », in : *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol.1, pp.1181-1191.

Rei, M., Yannakoudakis, H., 2017. "Auxiliary Objectives for Neural Error Detection Models », in : *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pp.33-43.

Reuter, Y., 1984. « Pour une autre pratique de l'erreur », in : *Pratiques : linguistique, littérature, didactique*, n.44, l'évaluation, pp. 117-126.

Reuter, Y., 2003. « Penser l'erreur à l'école : De l'erreur au dysfonctionnement », Presses universitaires du Septentrion, Villeneuve d'Ascq.

Richards, J., 1970. « A non-contrastive approach to error analysis », Prepared for the TESOL Convention, San Francisco.

Rothe, S., et al., 2021. « A Simple Recipe for Multilingual Grammatical Error Correction'. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, vol.2, pp.702-707.

Rozovskaya, A., Roth, D., 2010. « Annotating ESL Errors : Challenges and Rewards », in: *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pp.28-36.

Sautot, J-P., 2001. « Apprentissage de l'orthographe et insécurité linguistique face à l'écrit. », in : Colletta J-M., Tcherkassof A. (Ed.). « Émotion, interaction et développement », Universités Stendhal et Mendès France, Grenoble, pp. 193-199.

Satake, Y., 2020. « How Error Types Affect the Accuracy of L2 Error Correction with Corpus Use », in : *Journal of Second Language Writing*, vol.50, 100757.

Sautot, J-P., 2020. « Typologie d'erreurs d'orthographe Pour quoi faire ? », in : *négociations orthographiques*, halshs-02913518.

Sebba, M., 2006. « Spelling and society », Cambridge university Press, New-York.

Simon, J-P., 2006. « Perception des rectifications orthographiques par de futurs enseignants », in: Honvault-Durocq, R. (Dir). « L'orthographe en question », col. *Dylalang*, Publication des Université de Rouan et du Havre, Mont-Saint-Aignant, pp.191-2015.

Strujić, L-P., 2021. « Les erreurs d'orthographe dans l'écriture à la main et sur l'ordinateur chez les apprenants de FLE (A2) », *mémoire de master*, Orešković Dvorski L. (Dir), Université de Zagreb.

Tetreault, J., Filatova, E., Chodorow, M., 2010. « Rethinking Grammatical Error Annotation and Evaluation with the Amazon Mechanical Turk », in : *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 45-48.

Tetreault, J., Chodorow, M., Madnani, N., 2014. « Bucking the Trend : Improved Evaluation and Annotation Practices for ESL Error Detection Systems », in: *Language Resources and Evaluation vol.48(1)*, pp. 5-31.

Viémon, M., 2021. « Les erreurs fréquentes des étudiants universitaires hispanophones de FLE à l'écrit », in : *Anales de Filología Francesa, n.29*, pp.769-789.

Voutilainen, A., 1999. « A Short History of Tagging », in : van Halteren, H. (Ed.), « Syntactic Wordclass Tagging », Springer, Netherlands, pp. 9-21.

Wagner, J., Foster, J., van Genabith, J., 2007, « A Comparative Evaluation of Deep and Shallow Approaches to the Automatic Detection of Common Grammatical Errors", in : *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp.112-121.

Wolfarth, C., et al., 2017. « Apports du TAL à la constitution et à l'exploitation d'un corpus scolaire au travers du développement d'un outil d'annotation orthographique », in : *Corpus [en ligne]*, n.16.

Xiang, Y., Wang, X., Han, W., Hong, Q., 2015. « Chinese Grammatical Error Diagnosis Using Ensemble Learning », in : *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pp.99-104.

Yannakoudakis, H., Briscoe, T., Medlock, B., 2011. « A New Dataset and Method for Automatically Grading ESOL Texts », in : *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp.180-189.

Zhao, J., et al., 2017. « N-Gram Model for Chinese Grammatical Error Diagnosis », in : *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, pp.39-44.

Annexes et codes :

Annexe 1. Rectification des anomalies dans la « nouvelle orthographe » (Farid, 2012:2059)

Description grossière des réformes de l'orthographe instiguées en 1990 et abordées dans la section 2.1.2.

«

Anomalies rectifiées :

a) Les mots anciennement en -illier ou illière, où le i qui suit la consonne ne s'entend pas, s'écrivent maintenant en -iller ou -illère : joailler, marguiller, quincailer, serpillère.

b) Les termes en -ol(l)e, anciennement avec double l, s'écrivent maintenant avec un seul l (comme on écrit déjà bestiole, camisole, profiterole... : barcarole, corolaire, corole, mariole. Exceptions : on continuera à écrire, avec double l, folle, molle, colle et ses composés (décoller, recoller).

c) Les verbes -ot(t)er, anciennement avec double t, s'écrivent maintenant avec un seul t, de même que leurs dérivés : ballotage, ballotement, balloter, cachoterie, cachotier, friser, frisotis, grelotement, greloter, mangeoter.

d) Les termes suivants, dont la consonne doublait par erreur après un e muet, ne doublent plus maintenant leur consonne : dentelier (au lieu de dentellier) lunetier (au lieu de lunettier), prunelier (au lieu de prunellier), interpeler (au lieu de interpellier). Remarque : les consonnes restent doubles dans dentelle, lunette, prunelle, puisque le e de gauche n'est pas muet mais se prononce [è].

e) Des anomalies diverses ont été rectifiées : appâts (au lieu de appas); assoir, rassoir, sursoir; bizut (à cause des graphies bizuter, bizutage [au lieu de bizuth]); bonhomie (au lieu de bonhomie mais comme bonhomme) boursoufflement (au lieu de boursoufflement); boursouffler; boursoufflure; cahutte (au lieu de cahute); charriot (comme charrue); combattif (comme battre); combattivité (comme battre); cuisseau; déciller (comme cil); douçâtre; exéma (au lieu de eczéma); imbécilité (comme imbécile); innommé (comme nommé); levreau; nénufar; ognon (au lieu de oignon); pagaille (au lieu de pagaie, paye); persiflage (au lieu de persiflage); persiffler (comme siffler); relai (comme balai); saccharine (au lieu de saccharine) et ses dérivés; ventail (comme vent).

« De 1694 à 1975, l'orthographe française a subi au moins sept phases d'ajustements, et les Rectifications orthographiques de 1990 sont la dernière étape, du moins jusqu'à aujourd'hui. »

».

Annexe 2. « Typologie d'erreurs de Nina Catach, 1986 » (Sautot, 2020:5)

Typologie proposée par Nina Catach s'inscrivant dans son idée de « plurisystème » pour appréhender le manque de correspondance entre les phonèmes et les graphèmes en français. Typologie basée sur les erreurs des natifs du français, elle est abordée dans les points 2.1.1 et 2.1.4.

L'orthographe française : Traité théorique et pratique page 288
GRILLE TYPOLOGIQUE DES ERREURS D'ORTHOGRAPHE

Catégories d'erreurs	Remarques	Exemples
Erreurs extragraphiques		
Erreurs à dominante calligraphique	Ajout ou absence de jambage, lettres mal formées, etc.	*mid (nid)
Reconnaissance et coupure des mots	Peut se retrouver dans toutes les catégories suivantes	Le *lévier (l'évier) Un *navion (un avion)
Erreurs à dominante extragraphique (en particulier phonétique) L'écrit est erroné parce que l'oral est erroné	- Omission ou adjonction de phonèmes - Confusion de consonnes - Confusion de voyelles	*maintenant (maintenant) *suchoter (chuchoter) *crocodile (crocodile) *moner (mener)
Erreurs graphiques (oral – juste – écrit erroné)		
Erreurs à dominante phonogrammique (règles fondamentales de transcription et de position)	- N'altérant pas la valeur phonique - Altérant la valeur phonique	*binette (binette) *pingoin (pingouin) *guorille (gorille) *merite (mérite) *briler (briller) *recu (reçu)
Erreurs à dominante morphogrammique		
Morphogrammes grammaticaux	- Confusion de nature, de catégorie, de genre, de nombre, de forme verbale - Omission ou adjonction erronée d'accords étroits - Omission ou adjonction erronée d'accords larges	*chevaus (chevaux) *les rue (les rues) *ils chantes (chantent) *tu achète (achètes) *Ceux que les enfants ont vu (vus)
Morphogrammes lexicaux	- Marques du radical - Marques préfixes / suffixes	*canart (canard) *anterremant (enterrement) *annui (ennui)
Erreurs à dominante logogrammique		
Logogrammes lexicaux	- Confusion entre les homophones lexicaux	j'ai pris du *vain (vin)
Logogrammes grammaticaux	- Confusion entre les homophones grammaticaux	Ils *ce (se) sont dit *c'est livres (ces)
Erreurs à dominante idéogrammique	- Majuscules - Ponctuation - Apostrophe - Trait d'union	La *france (la France) *les, amis (les amis) *leau (l'eau) *peut être (peut-être)

(Les subdivisions et les exemples ne sont pas limitatifs. Les erreurs sont précédées d'un astérisque, les notations corrects sont mis entre parenthèses)

Annexe 3. « Typologie d’erreurs d’apprenant roumain du français » (Câslaru, 2016)

Adaptation de la typologie d’erreurs des apprenants de l’anglais proposée par Sylvianne Granger en 2003 (Annexe 4) pour les apprenants du français. Cette typologie s’articule sur trois niveaux, celui présenté ici et deux autres concernant les classes grammaticales et le caractère inter-langagier des erreurs (emprunt à la langue native et interférences). Cette typologie est décrite dans le point 2.1.4 et sert de point de départ à la création de la typologie présentée dans la section méthodologie de ce travail.

Lexique	Morphologie	Syntaxe
Erreurs de diacritiques (superflues ou omises)	Erreurs d’auxiliaires	Mauvais choix de préposition/ conjonctions (structuration de l’idée)
Erreurs de graphies et de forme (y inclus homophonie)	Erreurs de classe grammaticale	Omission d’un mot
Mots non standard (inexistant en français)	Erreurs de conjugaison (terminaison inexistante)	Mot superflu (répétition/redondance)
Mots inappropriés (existant mais mal utilisé)	Euphonie (absence d’élision ou de contraction)	Ordre des mots
	Erreurs de flexion (déclinaison des cas)	
	Confusion de genre	
	Erreurs dans le choix du mode verbal	
	Erreurs de nombres	
	Temps inapproprié	
	confusion des voix verbales	

Annexe 4. « Tagset pour l’annotation des erreurs ESL » (Granger, 2003 :468)

Typologie des erreurs des apprenants de l’anglais (ESL) agrémenté de balises en deux niveaux, le domaine d’erreur et le type d’erreur. Cette typologie est abordée dans la section 2.1.4 et sert de canevas à l’élaboration du guide d’annotation et à la création des balises proposées dans la section méthodologie.

Error Domains and Categories

Error Domains		Error Categories	
<F>	Form	<AGL> <MAJ> <DIA> <HOM> <GRA>	Agglutination Upper/lower case Diacritics Homonymy Other spelling errors
<M>	Morphology	<MDP> <MDS> <MFL> <MFC> <MCO>	Derivation-prefixation Derivation-suffixation Inflection Inflection-confusion Compounding
<G>	Grammar	<CLA> <AUX> <GEN> <MOD> <NBR> <PER> <TPS> <VOI> <EUF>	Class Auxiliary Gender Mode Number Person Tense Voice Euphony
<L>	Lexis	<SIG> <CPA> <CPD> <CPV> <CPN> <FIG>	Meaning Adjective complementation Adverb complementation Verb complementation Noun complementation Prefab
<X>	Syntax	<ORD> <MAN> <RED> <COH>	Word order Word missing Word redundant Cohesion
<R>	Register	<RLE> <RSY>	Lexis Syntax
<Y>	Style	<CLR> <LOU>	Unclear Heavy
<Q>	Punctuation	<CON> <TRO> <OUB>	Punctuation confusion Punctuation redundant Punctuation missing
<Z>	Typo		

Annexe 5. « Catégories d’erreurs du système de détection ERRANT » (Bryant et al, 2017 :3)

Typologie proposée dans l’optique d’améliorer les systèmes de correction automatique (données annexes) et de faciliter leur évaluation. Elle se base sur une approche par règles pour classifier les erreurs commises par des natifs de l’anglais.

Code	Meaning	Description / Example
ADJ	Adjective	<i>big</i> → <i>wide</i>
ADJ:FORM	Adjective Form	Comparative or superlative adjective errors. <i>goodest</i> → <i>best</i> , <i>bigger</i> → <i>biggest</i> , <i>more easy</i> → <i>easier</i>
ADV	Adverb	<i>speedily</i> → <i>quickly</i>
CONJ	Conjunction	<i>and</i> → <i>but</i>
CONTR	Contraction	<i>n’t</i> → <i>not</i>
DET	Determiner	<i>the</i> → <i>a</i>
MORPH	Morphology	Tokens have the same lemma but nothing else in common. <i>quick (adj)</i> → <i>quickly (adv)</i>
NOUN	Noun	<i>person</i> → <i>people</i>
NOUN:INFL	Noun Inflection	Count-mass noun errors. <i>informations</i> → <i>information</i>
NOUN:NUM	Noun Number	<i>cat</i> → <i>cats</i>
NOUN:POSS	Noun Possessive	<i>friends</i> → <i>friend’s</i>
ORTH	Orthography	Case and/or whitespace errors. <i>Bestfriend</i> → <i>best friend</i>
OTHER	Other	Errors that do not fall into any other category (e.g. paraphrasing). <i>at his best</i> → <i>well</i> , <i>job</i> → <i>professional</i>
PART	Particle	<i>(look) in</i> → <i>(look) at</i>
PREP	Preposition	<i>of</i> → <i>at</i>
PRON	Pronoun	<i>ours</i> → <i>ourselves</i>
PUNCT	Punctuation	<i>!</i> → <i>.</i>
SPELL	Spelling	<i>genectic</i> → <i>genetic</i> , <i>color</i> → <i>colour</i>
UNK	Unknown	The annotator detected an error but was unable to correct it.
VERB	Verb	<i>ambulate</i> → <i>walk</i>
VERB:FORM	Verb Form	Infinitives (with or without “to”), gerunds (-ing) and participles. <i>to eat</i> → <i>eating</i> , <i>dancing</i> → <i>danced</i>
VERB:INFL	Verb Inflection	Misapplication of tense morphology. <i>getted</i> → <i>got</i> , <i>fliped</i> → <i>flipped</i>
VERB:SVA	Subject-Verb Agreement	<i>(He) have</i> → <i>(He) has</i>
VERB:TENSE	Verb Tense	Includes inflectional and periphrastic tense, modal verbs and passivization. <i>eats</i> → <i>ate</i> , <i>eats</i> → <i>has eaten</i> , <i>eats</i> → <i>can eat</i> , <i>eats</i> → <i>was eaten</i>
WO	Word Order	<i>only can</i> → <i>can only</i>

Annexe 6. Typologie des erreurs, sous deux versions : version 1. Une version archaïque du guide d'annotation présentée en novembre 2022 (stage au CENTAL), version 2. tableau récapitulatif du type d'erreurs et des catégories. Attention, la typologie présentée dans la version 1 est une version antérieure et moins complète que celle présentée dans la version 2 et dans le guide.

Version 1 :

Typologie d'erreurs:

2. <L> Erreurs lexicales:

1.1 <DIA> Erreurs de diacritiques,

Toutes les erreurs qui concernent les signes diacritiques:

Utilisation des accents

**ocean / océan* (oubli)

**fôret / forêt* (mauvais placement)

**nécessaire / nécessaire* (ajout)

**français / français* (s'applique aussi pour la cédille)

Utilisation du trait d'union (agglutination de mots composés ou division de mots simples)

**portebagage / porte-bagage* (agglutination)

**centre ville / centre-ville* (oublis)

**contre-action / contraction* (division)

Utilisation de l'apostrophe

**d y / d'y*

1.2 <CA> Erreurs de casse,

Cette catégorie concerne les erreurs de majuscules et de minuscules (en début de phrase ou les nom propres)

**paris / Paris* (oubli)

1.3 <GRA> Erreurs de graphie et de forme,

Cette catégorie concerne les erreurs d'orthographe/ d'écriture:

Les coquilles

**ps / pas*

**qunad / quand*

L'ajout ou l'omission d'une lettre

**recomençai / recommençai* (omission)

**perssonnages / personnages* (ajout)

Le choix de l'homophone

!! Si l'homophone concerne la forme d'un verbe, se référer aux erreurs verbales (au plus proche du POS)

**Il ce déshabille/ il se déshabille*

L'écriture en phonétique (mot reconnaissable)

**labirint / labyrinthe*

Inversion de l'ordre des lettres

**écomonie / économie*

1.4 <MOT> Erreurs dans le choix du mot,

Cette catégorie englobe une série de sous catégories à annoter différemment, elles concernent toutes le choix du mot

1.4.1 <MOT-N> mot non standard:

Mot inexistant en français (emprunt L1 ou mot non reconnaissable)

**window / fenêtre* (emprunt anglais)

1.4.2 <MOT-I> choix de mot inapproprié:

Mot existant en français (bien orthographié) mais utilisé erronément

Le mot peut être incompris sémantiquement ou confondu avec un mot proche. Cela comprend aussi les erreurs sémantiques de co-occurrence et de collocations.

**table / tableau*

**pêche / poisson*

**cours religieux / cours de religion* (erreur de co-occurrence)
une économie **beaucoup mieux / une économie meilleure*

3. <MOR> Erreurs morphologiques:

2.1 <V> Erreurs du verbe,

Cette catégorie concerne toutes les erreurs de formation et d'utilisation du verbe

2.1.1 <V-T> Utilisation des temps:

Erreurs dans le choix du temps de conjugaison, y compris la concordance des temps (s'applique aussi à la concordance des temps)

*Un sondage qui *était publié / [...] a été publié*

*Si [V à l'imparfait] *j'aurai / Si [V à l'imparfait]] j'aurais* (confusion entre conditionnel présent et futur simple, utilisation du temps erronée)

2.1.2 <V-A> Erreurs d'auxiliaires:

Erreurs résultantes du mauvais choix de l'auxiliaire dans la construction d'un temps composé (correct) [si l'auxiliaire est correct mais mal conjugué/ orthographier, utiliser le tag correspondant au type de fautes]

*Je *m'avais très bien amusé / je m'étais très bien amusé*

2.1.3 <V-M> Utilisation du mode verbal:

Erreur du choix de mode (infinitif, indicatif ou subjonctif)

*Bien qu'ils *sont / bien qu'ils soient* (indicatif - subjonctif)

**T'invité / t'inviter* (indicatif - infinitif)

2.1.4 <V-V> Utilisation de la voix verbale:

Aussi bien les erreurs concernant la construction de la voix passive que celles concernant une confusion entre passif et actif

2.1.5 <V-C> Erreurs de conjugaison:

Toutes erreurs commises en rapport avec la terminaison verbale, mauvaise conjugaison (accord sujet-verbe ou verbe-code)

Ou encore si la terminaison est manquante

*Elle a *réussit / elle a réussi* (considéré comme un mauvais choix de terminaison et non comme une erreur de temps, car la construction et l'utilisation du temps est correcte)

*Ils *mangeait / ils mangeaient*

!! Quand l'erreur concerne le mode verbal se référer aux erreurs de mode

2.1.6 <V-P> Erreurs concernant les verbes pronominaux:

Mauvais choix dans le pronom, erreur dans l'élision, omission du pronom

*il *se joue / il joue* (pas un verbe pronominal)

*Il *me aime / il m'aime*

!! Pas les erreurs d'homophonie (se/ce)

2.2 <CP> Violations des modes de complémentation grammaticale des mots,

Valence verbale, nominale, adjectivale et adverbale: mauvais choix de prépositions pour introduire le complément (valide aussi pour la mauvaise formulation du complément en général)

**libre à choisir / libre de choisir* (complémentation adjectivale)

**se transformer dans / se transformer en* (complémentation verbale)

!! Englober le verbe dans la balise

2.3 <EUP> Erreurs d'euphonie,

Elision, liaison, contraction (problèmes phonétiques majoritairement)

**cette objet / cet objet*

**ce aspect / cet aspect*

**de le / du*

**je entre / j'entre*

2.4 <F> Erreurs de flexion (hors verbe),

Comme la flexion du verbe est déjà traitée, cette catégorie traite des erreurs de flexion pour les autres classes grammaticales

2.4.1 <F-G> Flexion en genre:

Ceci comprend l'accord en genre des adjectifs et déterminants avec le nom, mise au féminin des noms et des pronoms (hors verbes pronominaux) gestion des relations anaphoriques

**la vélo / le vélo*

**la robe bleu / la robe bleue*

**l'homme [...] elle / l'homme [...] il* (gestion des relations anaphoriques)

**acteur / actrice* (mauvaise gestion de la formation du genre)

2.4.2 <F-N> Flexion en nombre:

Ceci comprend l'accord en nombre des adjectifs et déterminants avec le nom, mise au pluriel des noms et des pronoms (hors verbes pronominaux) gestion des relations anaphoriques

**les vélo / les vélos*

**les petit enfants / les petits enfants*

**les hommes [...] il / les hommes [...] ils* (gestion des relations anaphoriques)

**travails / travaux* (mauvaise gestion des règles de formation du pluriel)

2.5 <D> Erreurs de dérivation,

Mauvaise utilisation des règles d'affixation, gestion erronée des suffixes et des préfixes pour dériver un mot (génération de nouveau mot, résultant ou non en un mot existant)

**malcontentement / mécontentement* (préfixe)

**évolutionner / évoluer* (suffixe)

4. <SYN> Erreurs syntaxiques:

3.1 <COH> Erreurs de cohérence,

Mauvais choix de connecteurs logiques (conjonctions, prépositions ...)

Ou Omission des connecteurs logiques

*ils sont petits *or fort / ils sont petits mais forts*

3.2 <MAN> Mot manquant

Omission d'un mot peu importe sa classe grammaticale

**j'ai pas / je n'ai pas* (construction tolérée à l'oral)

3.3 <SUP> mot superflu,

Tout mot inutilement rajouté dans la syntaxe (n'impactant pas une syntaxe déjà correcte)

*Qui mélange *entre la modernité et le classique / Qui mélange la modernité et le classique*

Cas complexe à discuter : « j'irai de nouveau avons-y à la montagne »

3.3 <REP> Mot répété,

Coquille ou duplication du sujet (mauvaise gestion du pronom et de son anaphore, incorrecte à l'écrit justifié par la valence verbale)

**le domaine social *et économique et politique / le domaine social , économique et politique*

**l'homme il va / l'homme va ou il va*

3.4 <ORD> Erreur dans l'ordre des mots,

Formant ainsi une construction grammaticalement inacceptable ou encore, construction grammaticalement acceptable mais dont le sens n'est pas transmis de manière effective.

Phrases « bizarres »

*Je peux *m'amuser bien / bien m'amuser*

3.5 <PCT> Ponctuation,

Erronée, manquantes ou répétée (coquilles ou mauvaise gestion de la ponctuation)

*Ce sont les différences *, qui sont présentées / ce sont les différences qui sont présentées*

Note:

Si la correction engendre une faute dans le reste du texte, ne pas corriger cette faute.

Si deux fautes différentes de même nature, annoter deux fois la même balise ?

Version 2 :

LEXIQUE	MORPHOLOGIE	SYNTAXE
Erreurs de casse	Erreurs Verbale (Auxiliaire)	Erreurs de cohérence
Erreurs de diacritique	Erreurs Verbale (temps)	Erreurs dans l'ordre des mot
Erreurs de graphie	Erreurs Verbale (conjugaison)	Mot manquant
Mot non-standard	Erreurs Verbale (mode)	Mot superflu
Mot inapproprié	Erreurs Verbale (voix)	Erreurs de répétition
	Erreurs Verbale (verbe pronominaux)	Erreurs de ponctuation
	Erreurs de complémentation	
	Erreurs d'euphonie	
	Erreurs de flexion (pronom)	
	Erreurs de flexion (genre)	
	Erreurs de flexion (nombre)	

Annexe 7. Guide final, version définitive du guide présentée aux annotateurs pour réaliser la tâche d'annotation manuelle. Les dernières modifications ont été réalisées lors de la création du Gold standard (février 2023).

Tableau récapitulatif :

TAG	DESCRIPTION	
L-CASSE	Erreur de majuscule (manquante ou en trop)	1.2
L-DIA	Erreur de signes diacritiques (accents, cédille, apostrophe, ...)	1.1
L-GRAPHIE	Erreur de forme ou de graphie (coquilles, ...)	1.3
L-MOT-N	Erreur de choix de mot, non standard en français	1.4.1
L-MOT-I	Erreur de choix de mot, inapproprié dans le contexte	1.4.2
M-V-AUX	Erreur dans l'utilisation de l'auxiliaire	2.1.2
M-V-CONJ	Erreur de conjugaison (terminaison flexionnelle)	2.1.5
M-V-MODE	Erreur dans le mode verbal	2.1.3
M-V-PRON	Erreur pour le verbe pronominal	2.1.6
M-V-TEMPS	Erreur dans l'utilisation des temps verbaux	2.1.1
M-V-VOIX	Erreur d'utilisation de la voix verbale	2.1.4
M-CMP	Erreur de complémentation (verbal, nominal, adjectival, ... (prépositions)	2.2
M-EUP	Erreur d'euphonie (liaison)	2.3
M-F-PRONOM	Erreur de flexion du pronom	2.4.1
M-F-GENRE	Erreur de flexion (non-verbale) en genre	2.4.2
M-F-NOMBRE	Erreur de flexion (non-verbale) en nombre	2.4.3
S-COH	Erreur de cohérence (conjonctions)	3.1
S-MAN	Mot manquant	3.2
S-SUP	Mot superflu	3.3
S-REP	Répétition et redondance	3.4
S-ORD	Ordre des mots	3.5
S-PCT	Erreur de ponctuation	3.6
ERR	Construction incompréhensible	4

Guide d'annotation :

La tâche qui vous est demandée est d'annoter les erreurs d'orthographe et de grammaire présentes dans des textes d'apprenants du français (FLE). Pour ce faire, vous aurez accès aux textes des apprenants dans une interface d'annotation (INCEpTION) qui vous permettra d'annoter chaque erreur d'un ou plusieurs tags correspondants. Le guide que vous vous apprêtez à lire contient l'explication de tous les types d'erreurs que vous rencontrerez ainsi que les tags et la manière d'annoter. À ce guide est associé un guide d'utilisation de l'outil INCEpTION.

Principes fondamentaux :

- a. Tous les tags sont cumulables si plusieurs erreurs coexistent dans le même mot/constituent.
- a. L'unité d'annotation est le TOKEN, cette unité est donc indivisible et devra être annotée dans son entièreté. Il conviendra donc d'annoter les mots en entier. Pour les erreurs impliquant une suite de TOKEN, il convient d'annoter tous les mots impliqués dans la construction erronée.

Par exemple:



- b. Si la correction engendre des fautes dans la suite du texte, ces fautes ne sont pas prises en compte.

Par exemple: *les voiture sont une très bonne *moyenne de locomotion.*

Le mot *moyenne* est ici confondu avec le mot *moyen* et sera annoté de la balise

<L-MOT-I>, les accords en genre étaient corrects (accord en féminin avec le mot *moyenne*) ils ne devront pas être annoté malgré le changement de mot.

- c. Si un même mot comporte deux fautes de même nature, il convient d'annoter deux fois la même balise (une par faute). Attention tout de même, quand le mot est écrit phonétiquement (plusieurs erreurs individuelles peuvent apparaître).
- d. Toujours privilégier le choix d'une balise plus précise que le cumul des tags.
- e. De manière générale, les erreurs de cohésion, de discours, de lecture de consignes** ne sont pas des erreurs à annoter (trop subjectif et ne concernent pas la maîtrise des règles de la langue écrite).
- f. Ce qui ne « sonne » pas natif, mais n'est pas faux, est correct. Tant que grammaticalement et orthographiquement la phrase est correcte il ne faut rien annoter, même si cette dernière semble alambiquée ou peu appropriée.

- g. Dans le cas d'erreurs impliquant la nouvelle orthographe, il convient de tenir compte des deux orthographes et de les considérer comme correctes même si elles sont présentes simultanément dans un document.
- h. Dans le cas où plusieurs interprétations et donc plusieurs corrections sont possibles, privilégiez toujours celle qui aura le moins d'impact sur le texte original (évitez au maximum de supprimer des mots qui ont du sens par exemple).

**et nous prendre de photos*

Évitez dans cet exemple de supprimer le nous et de modifier le de en « des » pour créer « *et prendre des photos* »

Mais optez plutôt pour changer uniquement le « de » en « en » (<M-CMP>) pour créer « *et nous prendre en photos* » beaucoup plus proche de la phrase originale

S-MAN et M-CMP nous prendre de photos.

**Dans en de notre voyage*

Ici aussi privilégiez l'interprétation « *pendant un de nos voyages* » qui reste plus fidèle à la construction d'origine.

M-CMP dans L-MOT-I en M-F-NOMBRE de M-F-NOMBRE notre L-GRAPHIE voage

- i. Attention, évitez la paresse lors de vos annotations, même si plusieurs mots **côtes à côtes** ont la même erreur, annotez-les séparément (cf. Le point b). Très peu de cas nécessitent que vous annotiez des constructions de plusieurs mots, en règles générales annotez mot-à-mot.

Les informations d'annotations sont en orange dans ce guide.

- j. Notez que les constructions emphatiques sont acceptables en français, encore une fois privilégiez les erreurs de ponctuations dans ces cas plutôt que la suppression de l'élément qui forme l'emphase.

**toi tu nous as manqué*

À la rigueur annotez le manque de la virgule entre « toi » et « tu » mais la construction est correcte.

**je voulais inviter vous et vos enfants*

Ici aussi, à la rigueur indiquez un « vous » manquant être « voulais » et « inviter » mais la construction est correcte.

**J'espere que vous êtes bien, toi, tes parents et toute la famille*

Ici encore une construction emphatique qui n'est pas une erreur ni à l'oral ni à l'écrit.

- k. Pour les noms propres toujours vérifier s'il y a une erreur de casse <L-CASSE>, pour les erreurs de graphie <L-GRAPHIE> annotez-les uniquement si le contexte vous indique pour sûr que le mot est erronément écrit. Vous n'êtes jamais à l'abri de tomber sur un nom de ville que vous ne connaissez pas même si ce dernier est proche d'un nom que vous connaissez.

La typologie est divisée en quatre grandes catégories d'erreurs :

5. les erreurs Lexicales (par exemple, graphie et choix des mots),
6. les erreurs de morphologie (par exemple, flexion, dérivations, accords),
7. Les erreurs syntaxiques (par exemple, erreurs au niveau de la phrase, ponctuation),
8. Les constructions incompréhensibles (par exemple, erreurs de dernier recours pour des phrases incompréhensibles).

En amont de ce guide, vous trouvez un tableau récapitulatif des tags ainsi qu'une brève description, ce tableau sera aussi disponible dans l'interface d'annotation.

1. <L> Erreurs lexicales :

1.1 <DIA> Erreurs de diacritiques,

Toutes les erreurs qui concernent les signes diacritiques sont à classer dans cette catégorie.

NE PAS CONFONDRE, signe diacritique (trait d'union, accent, apostrophe) et signe de ponctuation (parenthèses, points, guillemets).

Utilisation des accents.

Annoter uniquement le mot qui comporte l'erreur d'accent.

**ocean / océan* (omission)

- Y compris, l'absence d'accents sur les majuscules -

**fôret / forêt* (mauvais placement)

**nécessaire / nécessaire* (ajout)

**français / français* (s'applique aussi pour la cédille)

Utilisation de l'apostrophe.

**d y / d'y*

Toujours annoter les deux mots et l'espace qui composent la construction erronée.

Attention ne concerne pas les erreurs dans la rédaction des phénomènes d'élision et de liaison qui sont annotés <M-EUP>.

Utilisation du trait d'union (agglutination de mots composés ou division de mots simples) .

Attention, lorsqu'un espace indésirable est placé entre le trait d'union et le mot, cette erreur est aussi considérée comme une erreur de diacritique.

Toujours annoter les deux mots, l'espace et le tiret qui composent la construction erronée.

**portebagage / porte-bagage* (agglutination)

**centre ville / centre-ville* (omission)

**contre-action / contraction* (division)

**porte- bagage/ porte-bagage*

1.2 <CASSE> Erreurs de casse,

Cette catégorie concerne les erreurs de majuscules et de minuscules (en début de phrase ou les noms propres).

Annoter uniquement le mot qui compte l'erreur de casse.

**paris / Paris* (omission)

1.3 <GRAPHIE> Erreurs de graphie et de forme,

Annoter uniquement le mot qui compte l'erreur de graphie.

Attention, la confusion entre à et a / ou et où est considérée comme une erreur de diacritique.

Les coquilles.

**ps / pas*

**qunad / quand* (inversion de l'ordre des lettres)

**écomonie / économie* (idem)

L'ajout ou l'omission d'une lettre.

**recomençait / recommençait* (omission)

**perssonnages / personnages* (ajout)

L'écriture en phonétique.

Cette catégorie concerne uniquement les mots qui sont phonétiquement reconnaissables (Tips: prononcez les mots pour lesquels vous avez un doute, si le son est reconnaissable, annoter <L-GRA> si vous ne reconnaissez aucun mot du français annotez <L-MOT-N>.

**labirint / labyrinthe*

1.4 <MOT> Erreurs dans le choix du mot,

1.4.1 <MOT-N> mot non standard :

Annoter uniquement le mot non-standard (s'il s'agit d'une expression lexicale, annoter l'expression).

Mot inexistant en français (emprunt L1 ou mot non reconnaissable).

La catégorie fonctionne aussi pour les néologismes inutilisés en français, ou quand le mot est tellement déformé qu'il en devient incompréhensible.

**window / fenêtre* (emprunt anglais)

Dans le doute entre erreur de graphie ou mot non standard, n'utilisez cette balise que si vous avez une intuition forte à propos d'un emprunt à une autre langue.

Erreurs de dérivation.

Mauvaise utilisation des règles d'affixation, gestion erronée des suffixes et des préfixes pour dériver un mot (génération de nouveau mot, résultant ou non en un mot existant)

**malcontentement / mécontentement* (préfixe)

**évolutionner / évoluer* (suffixe)

Attention, cette catégorie concerne aussi les expressions lexicales inexistantes en français (empruntées à une autre langue).

Dans ce cas annoter tous les mots qui composent l'expression erronée.

1.4.2 <MOT-I> choix de mot inapproprié :

Mot existant en français mais utilisé erronément

Le mot peut être incompris sémantiquement ou confondu avec un mot proche. Cela comprend aussi les erreurs sémantiques de cooccurrence et de collocation^s.

**table / tableau*

**pêche / poisson*

**cours religieux / cours de religion* (erreur de cooccurrence)

une économie **beaucoup mieux* / une meilleure économie

Pour les erreurs de cooccurrence: annoter tous les mots qui composent la construction erronée.

Avec **vu sur / avec vue sur* (confusion entre le verbe voir et le substantif)

Concerne aussi les erreurs de confusion (ex. les sons *au* et *ou*).

*Je *voudrais bien aller / je voudrais bien y aller*

*L'homme *qui est devenu / l'homme qu'il est devenu*

Les erreurs d'utilisation d'expression lexicales sont à classer dans cette catégorie également.

Pour les erreurs d'expressions lexicales: annoter tous les mots qui composent l'expression erronée.

*J'aimerais que ce soit fait *dans le plus proche possible / *le plus rapidement possible*

Attention cette catégorie ne concerne pas les ajouts de mots superflus annotés <S-SUP>, elle ne concerne que les constructions grammaticales correctes dont un mot ou une expression lexicale est mal choisi.

Attention, si le mot inapproprié est mal orthographié, il sera annoté doublement <L-MOT-I> pour le choix de mot et <L-GRA> pour la graphie.

Cette catégorie englobe aussi les erreurs de choix de l'homophone comme le son/sont

2. <M> Erreurs morphologiques :

2.1 <V> Erreurs du verbe,

2.1.1 <V-TEMPS> Utilisation des temps :

Erreurs dans le choix du temps de conjugaison, y compris la concordance des temps.

Annotez uniquement le verbe qui comporte l'erreur de temps.

*Un sondage qui *était publié / [...] a été publié*

*Si [V à l'imparfait] *j'aurai / Si [V à l'imparfait]] j'aurais* (confusion entre conditionnel présent et futur simple, utilisation du temps erronée)

2.1.2 <V-AUX> Erreurs d'auxiliaires :

Erreurs résultantes du mauvais choix de l'auxiliaire dans la construction d'un temps composé (correct) [si l'auxiliaire est correct, mais mal conjugué/ orthographié, utiliser le tag correspondant au type de fautes].

N'utilisez pas cette balise si l'auxiliaire est manquant, utilisez plutôt <S-MAN>.

Annotez uniquement l'auxiliaire, pas la forme verbale en entier.

*Je *m'avais très bien amusé / je m'étais très bien amusé*

2.1.3 <V-MODE> Utilisation du mode verbal :

Erreur du choix de mode (infinitif, indicatif ou subjonctif)

Annotez uniquement le verbe qui comporte l'erreur de mode.

*Bien qu'ils *sont / bien qu'ils soient* (indicatif > subjonctif)

**T'invité / t'inviter* (indicatif > infinitif)

2.1.4 <V-VOIX> Utilisation de la voix verbale :

Aussi bien les erreurs concernant la construction de la voix passive que celles concernant une confusion entre passif et actif.

**Le policier a été arrêté le voleur / le voleur a été arrêté par le policier (ou le policier a arrêté le voleur)* (confusion entre les objet/sujet des voix passives et actives)

Annoter uniquement le verbe.

*le piéton a été renversé *de la voiture / le piéton a été renversé par la voiture*

Annoter uniquement la préposition erronée.

2.1.5 <V-CONJ> Erreurs de conjugaison :

Toutes erreurs commises en rapport avec la terminaison verbale, mauvaise conjugaison (accord sujet-verbe ou verbe-COD)

Cette catégorie s'applique aussi si la terminaison est manquante.

Annotez uniquement le verbe qui comporte l'erreur de conjugaison.

*Elle a *réussit / elle a réussi*

(considéré comme un mauvais choix de terminaison et non comme une erreur de temps, car la construction et l'utilisation du temps est correcte)

*Ils *mangeait / ils mangeaient*

Attention quand l'erreur concerne le mode verbal se référer aux erreurs de mode <M-V-MODE>.

2.1.6 <V-PRON> Erreurs concernant les verbes pronominaux :

Mauvais choix dans le pronom, erreur dans l'élision, omission du pronom, mauvaise gestion des relations anaphoriques (quand le pronom ne correspond pas au nom auquel il se rapporte).

Annoter aussi bien le verbe que le pronom erroné, ou uniquement le verbe si le pronom est manquant.

Attention ceci comprend les erreurs de choix de mot se/ce par exemple.

*il *se joue / il joue* (pas un verbe pronominal)

*Il *me aime / il m'aime*

*C'est toi que j'aime, je *l'aime / c'est toi que j'aime, je t'aime* (relation anaphorique)

Attention, cette catégorie est TOUJOURS à envisager en priorité sur des erreurs de <L-GRAPHIE> ou des erreurs de <F-PRONOM>. Pour ces dernières vérifiez toujours qu'il ne s'agisse pas d'un verbe pronominal avant de l'envisager.

2.2 <CMP> Violations des modes de complémentation grammaticale des mots,

Valence verbale, nominale, adjectivale et adverbiale : mauvais choix de prépositions pour introduire le complément (valide aussi pour la mauvaise formulation du complément en général), tous les types de compléments rentrent dans cette catégorie (y compris les compléments plus larges comme les compléments circonstanciels).

**libre à choisir / libre de choisir* (complémentation adjectival)

**se transformer dans / se transformer en* (complémentation verbale)

*Il va à Paris *dans le train / Il va à Paris en train*

Annoter uniquement la partie du complément problématique (en règle générale, la préposition).

2.3 <EUP> Erreurs d'euphonie,

Élision, liaison, contraction (problèmes phonétiques majoritairement)

Annoter uniquement les mots qui composent la construction euphonique erronée ET NON toute la construction.

**ce aspect / cet aspect*

Ici annoter le « de », l'espace et le « le »

**de le / du*

**je entre / j'entre*

Attention, ne pas annoter les erreurs d'utilisation de l'apostrophe si le phénomène d'euphonie est correct (privilégier l'erreur de diacritique <L-DIA>)

Ici annoter le « d », l'espace et le « y »

**d y / d'y* (à annoter <L-DIA> et non <M-EUP>)

Attention, si il y a des erreurs de découpage de mot lorsqu'on fait une liaison, il faut annoter les deux mots séparément:

**la partement*

La = erreur de genre + erreur d'euphonie <M-F-GENRE> <M-EUP>

Partement = mot non-standard <L-M-N>

**un navion*

Un = mot correct

Navion = erreur de graphie <L-GRAPHIE>

2.4 <F> Erreurs de flexion (hors verbe),

Comme la flexion du verbe est déjà traitée, cette catégorie traite des erreurs de flexion pour les autres classes grammaticales

2.4.1 <F-PRONOM> Flexion du pronom:

Ceci comprend les cas où la flexion d'un pronom serait erronée, aussi bien la déclinaison de celui-ci (exemples 1) ou la personne à laquelle ils renvoient (exemples 2). Cette catégorie concerne aussi bien les groupes nominaux pronominalisés que les pronoms relatifs introduisant une proposition subordonnée.

Attention, veuillez dans les cas où le pronom se trouve avant le verbe à bien différencier une erreur dans un verbe pronominal <M-V-PRON> et les cas où le pronom renvoie à un objet indirect qui aurait été placé avant le verbe.

Veuillez à annoter uniquement le pronom qui contient l'erreur.

Exemples 1:

*Ce *que va se passer / ce qui va se passer*

*Je *toi le donne / je te le donne* (je donne « le » à « toi », pas de verbe pronominal)

Exemples 2:

*Quand tu seras là, je veux *vous raconter / quand tu seras là, je veux te raconter*

*Ta mère et toi, [...] *tu allez au cinéma demain / vous allez au cinéma demain* (faire attention au contexte, il est clair ici que le verbe est correctement conjugué et c'est donc le choix du pronom qui pose problème)

Attention à toujours prendre en compte le contexte avant d'annoter ce genre d'erreurs.

Il te lave —> selon de contexte peut être une erreur dans le choix du pronom, une erreur de verbe pronominal ou une construction correcte.

2.4.2 <F-GENRE> Flexion en genre :

Ceci comprend l'accord en genre des adjectifs et déterminants avec le nom, mise au féminin des noms et des pronoms (hors verbes pronominaux) gestion des relations anaphoriques (quand le pronom ne correspond pas au nom auquel il se rapporte).

Annoter uniquement le mot mal accordé, si plusieurs mots ont la même erreur, annotez-les séparément.

**la vélo / le vélo*

**la robe bleu / la robe bleue*

**l'homme [...] elle / l'homme [...] il* (gestion des relations anaphoriques)

**acteurs / actrices* (mauvaise gestion de la formation du genre)

Attention quand l'erreur de choix du mot influence les accords et flexions dans la phrase, annoter en priorité le choix du mot et laisser les flexions telles quelles

2.4.3 <F-NOMBRE> Flexion en nombre :

Ceci comprend l'accord en nombre des adjectifs et déterminants avec le nom, mise au pluriel des noms et des pronoms (hors verbes pronominaux) gestion des relations anaphoriques (quand le pronom ne correspond pas au nom auquel il se rapporte).

Annoter uniquement le mot mal accordé, si plusieurs mots ont la même erreur, annotez-les séparément.

**les vélo / les vélos*

**les petit enfants / les petits enfants*

**les hommes [...] il / les hommes [...] ils* (gestion des relations anaphoriques)

**travails / travaux* (mauvaise gestion des règles de formation du pluriel)

2. <S> Erreurs syntaxiques :

3.1 <COH> Erreurs de cohérence,

Mauvais choix ou omission de connecteurs logiques c'est à dire, dans une majorité des cas, les conjonctions de subordination et de coordination.

Annoter uniquement la conjonction erronée.

*ils sont petits *or fort / ils sont petits mais forts*

**comme tu étais venu tu n'aurais pas raté la fête / si tu étais venu [...]*

Attention, la répétition de « et » lors d'une énumération qu'il convient d'annoter <S-REP>.

Attention, certaines prépositions peuvent y ressembler mais veuillez toujours à vérifier si l'erreur dans le choix de la préposition ne constitue pas une erreur de complémentation <M-COMP>.

3.2 <MAN> Mot manquant

Omission d'un mot (ou d'un groupe de mots) peu importe sa classe grammaticale.

Indiquer le mot manquant sur le mot qui doit normalement le précéder.

**j'ai pas / je n'ai pas* (construction tolérée à l'oral)

3.3 <SUP> mot superflu,

Tout mot (ou construction) inutilement rajouté dans la syntaxe (n'impactant pas une syntaxe déjà correcte). La suppression de cet élément doit être supportée par des évidences langagières (sémantique ou syntaxique).

Annoter uniquement le mot superflu, si plusieurs mots sont superflus côte à côte et ne forment pas une construction à part entière, les annoter séparément. S'il s'agit d'une construction, annotez la construction d'un bloc.

*Qui mélange *entre la modernité et le classique / Qui mélange la modernité et le classique
j'irai de nouveau *allons-y à la montagne / j'irai de nouveau à la montagne*
(grammaticalement incorrecte d'ajouter un nouveau verbe dont le sujet diffère du premier)

3.4 <REP> Mot répété,

Coquille, répétition ou duplication du sujet (mauvaise gestion du pronom et de son anaphore, incorrecte à l'écrit, justifiée par la valence verbale).

Annoter uniquement le mot répété.

**le domaine social *et économique et politique / le domaine social , économique et politique
l'homme il va / l'homme va ou il va

3.5 <ORD> Erreur dans l'ordre des mots,

Formant ainsi une construction grammaticalement inacceptable ou encore, construction grammaticalement acceptable, mais dont le sens n'est pas transmis de manière effective.

Les phrases grammaticalement correctes, mais dont l'ordre des constituants est tellement éloigné de la construction native que le sens se perd.

*Je peux *m'amuser bien / bien m'amuser*

Attention, quand la phrase est correcte, mais simplement le style paraît trop lourd ou le registre pas approprié, la phrase reste considérée comme correcte.

3.6 <PCT> Ponctuation,

Erronée, manquante ou répétée (coquilles ou mauvaise gestion de la ponctuation)

Lorsque la ponctuation finale est manquante, indiquez la faute sur le dernier mot de la phrase. Pour le reste, toute ponctuation manquante est indiquée sur le mot qui doit normalement la précéder. Si la ponctuation est PRÉSENTE, annotez la ponctuation.

Ce sont les différences, qui sont présentées / ce sont les différences qui sont présentées*

Attention, lorsqu'un espace est placé entre le mot et la virgule, le point = une erreur

Lorsque l'espace est oublié avant et/ou après les deux points, le point d'interrogation et le point d'exclamation = une erreur⁸

J'aime les chats, les chiens et les poules / j'aime les chats, les chiens et les poules*

Il m'a dit: « vas-y » / il m'a dit : « vas-y »*

3. <ERR> Les constructions incompréhensibles :

Cette catégorie n'est à utiliser que si la construction est complètement obscure. Seulement quand aucun sens de la construction ne peut être interprété tant la grammaire, le choix des mots et l'orthographe sont défaillants.

TOUJOURS privilégier la correction élément par élément pour en tirer du sens avant d'employer ce tag.

Annoter toute la partie incompréhensible.

CAS PROBLÉMATIQUES:

- **le chateau est très proche de Paris, à une heure si on prend le train, à coté de Lille*

Ici, l'erreur réside dans le placement des virgules et non dans l'ordre des mots.

- **me confirmez votre(s) présence(s)*

Annotez-les « (s) » superflus comme des mots superflus <S-SUP>.

- L'accord du participe passé avec le sujet « on » se fera TOUJOURS au singulier, même si le sens du mot « on » est pluriel.⁹

8 « On ne ménage pas d'espace *avant* le point, la virgule, les points de suspension (voir N. B. ci-dessous), la parenthèse fermante, le crochet fermant ; — *après* la parenthèse ouvrante, le crochet ouvrant.

Il y a une espace *avant* les guillemets ouvrants, la parenthèse ouvrante, le crochet ouvrant, le tiret ; — *après* le point, la virgule, le point-virgule, le double point, les points d'interrogation et d'exclamation, les points de suspension, les guillemets fermants, la parenthèse fermante, le crochet fermant, le tiret. Il y a une espace, mais non sécable (c'est-à-dire que cette espace ne peut coïncider avec le passage à la ligne) *avant* le point-virgule, le double point, les points d'interrogation et d'exclamation, les guillemets fermants, — *après* les guillemets ouvrants. — Autrement dit : une ligne ne peut commencer par un point-virgule, un double point, un point d'interrogation ou d'exclamation, des guillemets fermants, une parenthèse fermante, — ni se terminer par des guillemets ouvrants. »

GREVISSE, M. et GOOSSE, A. 2008. « le bon usage, Grammaire française 14e édition », Éditions De Boeck Université, Bruxelles, p. 122.

9 « le verbe étant toujours au singulier avec *on*. »

Annexe 8 à 11. Codes de traitement des données une fois extraites de l'interface d'annotation INCEpTION. Le format d'extraction des données étant le format propre à la plate-forme, il fallait traiter les données pour correspondre au format CONLL nécessaires pour toutes les autres étapes du projet.

Annexe 8. Marche à suivre standard pour gérer la *tokenization* [26] et les formats des balises : étapes 1, 3 et 4 du formatage (l'étape 2 étant simplement la concaténation des documents).

```
#####
# script : traitement_1.py
# auteur : Angela Kasparian
# date : janvier/fevrier 2023
#####
# Ce script import les documents annotés sous le format WebAnno tsv V3, et tokenize
# plus finement la colonne "token", il crée donc un enouvelle colonne qui contient
# la nouvelle tokenisation et met à jour les lignes, indices et balises pour chancun
# de nouveaux tokens.
#####

import string
from nltk.tokenize import word_tokenize
import csv
from io import StringIO

with open('/Users/akaspri/Desktop/docs complets/batch 3/curation_83.tsv', 'r', encoding= 'UTF-8') as f:
    with open('/Users/akaspri/Desktop/docs complets/batch 3/curation_83_1.tsv', 'w', encoding= 'UTF-8') as l:
        w = csv.writer(l, delimiter='\t')
        index2indexes = {}
        lst_lines = []
        for ln in f:
            line = ln.strip().split('\t')
            if len(line)>2:
                word = line[2].replace(" ", " ").replace("'", " ").replace("-", " ")
                word = word.replace("-", " - ")
                word = word.replace(".", ". ")
                word = word.replace(", ", ", ")
                if "<br" in word:
                    index1A = int(line[1].split("-")[0])
                    index1B = int(line[1].split("-")[1])
                    word = word.replace("<br", " ")
                    index1B = index1A + len(word)
                    line[1] = str(index1A) + "-" + str(index1B)
                if ">" in word:
                    index1A = int(line[1].split("-")[0])
                    new = word_tokenize(word, language='french')
                    index0 = line[0]
                    index1A = int(line[1].split("-")[0])
                    index1B = int(line[1].split("-")[1])

                    for aux in new:
                        index2indexes_index = str(index1A) + "-" + str(index1B)
                        index1B = index1A + len(aux)
                        index2indexes_index = str(index1A) + "-" + str(index1B)
                        index2indexes[index2indexes_index] = len(lst_lines)

                        line_aux = [index0] + [index2indexes_index] + [line[2]] + [line[3]] + [aux]

                        lst_lines.append(line_aux)

                    if index2indexes_index not in index2indexes:
                        print("line",line)
                        print("orig","\n".join([" ".join(v) for v in lst_lines[max(0,len(lst_lines)-5):]]))

                    index_lst = index2indexes[index2indexes_index]

                    if lst_lines[index_lst][3] == " ":
                        lst_lines[index_lst][3] = line[3]
                    else:
                        for v in lst_lines[index_lst][3].split(";"):
                            if v == line[3]:
                                break
                            else:
                                lst_lines[index_lst][3] += ";" + line[3]

                    index1A = index1B
                    index2indexes_index = str(index1A) + "-" + str(index1B)

            for line in lst_lines:
                if "." not in line[0]:
                    w.writerow(line)

print('done')
```

```

=====
# script : traitement_3.py
# auteur : Angela Kasparian
# date : janvier/fevrier 2023
=====
# Ce script prend les documents tokenizés corectement par annotateurs et se charge
# de nettoyer l'affichage des balises format : balise [n]| balise [n] pour avoir
# un format facilement manipulable format de sortie : balise;balise
=====

import string
from nltk.tokenize import word_tokenize
import csv

with open('/Users/akaspari/Desktop/docs complets/batch 2/final.tsv', 'r', encoding= 'UTF-8') as f:
    with open('/Users/akaspari/Desktop/docs complets/batch 2/TRAIN.tsv', 'w', encoding= 'UTF-8') as l:
        w= csv.writer(l, delimiter='\\t')
        lst_line=[]
        for ln in f:
            line = ln.strip().split('\\t')
            word = line[3].replace("[", "")
            for i in string.digits:
                word= word.replace(i, "")
            word= word.replace("]", "")
            word= word.replace("|", ";")

            line_aux = [line[0]] + [line[1]] + [line[2]] + [word] + [line[4]]
            lst_line.append(line_aux)
        for line in lst_line:
            w.writerow(line)
print("done")

```

```

=====
# script : traitement_4.py
# auteur : Angela Kasparian
# date : janvier/fevrier 2023
=====
# ce script récupère le fichier de l'étape précédente et clone les lignes contenant
# des balises doubles (balise;balise) [et plus] afin de séparer les balises pour
# ne gerer qu'une balise à la fois. Le token sur lequel la balise est blacée reste
# inchangé.
=====

import string
import csv

with open('/Users/akaspari/Desktop/docs complets/batch 2/TRAIN.tsv', 'r', encoding= 'UTF-8') as a:
    with open('/Users/akaspari/Desktop/docs complets/batch 2/TRAIN_brut.tsv', 'w', encoding= 'UTF-8') as a2:
        ade= csv.reader(a, delimiter='\\t', quotechar='')
        ade2= csv.writer(a2, delimiter='\\t')
        lst_ade=[]
        for line in ade:
            lst_ade.append(line)
        for i in range(len(lst_ade)):
            lst_labels=[]
            line_ade= lst_ade[i]
            label_ade= line_ade[3].split(";")
            for lab1 in label_ade:
                lst_labels.append(lab1)
            for lb in lst_labels:
                line_new= [line_ade[0]] + [line_ade[1]] + [line_ade[2]] + [line_ade[3]] + [lb] + [line_ade[4]]
                ade2.writerow(line_new)
print('done')

```

Annexe 9. Extrait de l'étape 5, gestion des balises, transformation des balises brutes (alphabétiques) en balises numériques (23 balises).

```
#####  
# script : traitement_5.py  
# auteur : Angela Kasparian  
# date : janvier/fevrier 2023  
#####  
# ce script récupère le fichier de l'étape précédente et transforme les balises  
# en balises numérique afin de les traiter pour l'accord et dans les modèles ML  
# Attention cette étape est facultative au besoin d'avoir des balises numérique ou  
# non.  
#####  
  
import string  
from nltk.tokenize import word_tokenize  
import csv  
  
with open('/Users/akaspari/Desktop/docs complets/batch 2/TRAIN_brut.tsv', 'r') as f:  
    with open('/Users/akaspari/Desktop/docs complets/batch 2/TRAIN_nombres.tsv', 'w') as l:  
        r = csv.reader(f, delimiter='\t', quotechar='')  
        w = csv.writer(l, delimiter='\t')  
        for line in r:  
            tag = line[4]  
            if tag == '_':  
                line[4]='0'  
                w.writerow(line)  
            elif tag == 'ERR':  
                line[4]='1'  
                w.writerow(line)  
            elif tag == 'L-CASSE':  
                line[4]='2'  
                w.writerow(line)  
            elif tag == 'M-F-PRONOM':  
                line[4]='3'  
                w.writerow(line)  
            elif tag == 'L-DIA':  
                line[4]='4'  
                w.writerow(line)  
            elif tag == 'L-GRAPHIE':  
                line[4]='5'  
                w.writerow(line)  
            elif tag == 'L-MOT-I':  
                line[4]='6'  
                w.writerow(line)  
            elif tag == 'L-MOT-N':  
                line[4]='7'  
                w.writerow(line)
```

Annexe 10. Extrait de l'étape 5 bis, gestion des balises brutes en balises numériques (5 balises) pour les catégories linguistiques.

```
#####
# script : traitement_5_classes.py
# auteur : Angela Kasparian
# date : janvier/fevrier 2023
#####
# ce script récupère le fichier de l'étape précédente et transforme les balises
# en balises numérique afin de les traiter pour l'accord et dans les modèles ML
# Attention cette étape est facultative au besoin d'avoir des balises numérique ou
# non.
# Il ne crée pas des balises pour chacune des erreurs mais bien 5 balises pour les
# classes linguistiques.
#####

import csv

with open('/Users/akaspari/Desktop/docs complets/batch 2/TRAIN_brut.tsv', 'r') as f:
    with open('/Users/akaspari/Desktop/docs complets/batch 2/TRAIN_classes.tsv', 'w') as l:
        r = csv.reader(f, delimiter='\t', quotechar='')
        w = csv.writer(l, delimiter='\t')
        for line in r:
            tag = line[4]
            if tag == '_':
                line[4]='0'
                w.writerow(line)
            elif tag == 'ERR':
                line[4]='1'
                w.writerow(line)
            elif tag == 'L-CASSE':
                line[4]='2'
                w.writerow(line)
            elif tag == 'M-F-PRONOM':
                line[4]='3'
                w.writerow(line)
            elif tag == 'L-DIA':
                line[4]='2'
                w.writerow(line)
            elif tag == 'L-GRAPHIE':
                line[4]='2'
                w.writerow(line)
            elif tag == 'L-MOT-I':
                line[4]='2'
                w.writerow(line)
            elif tag == 'L-MOT-N':
                line[4]='2'
                w.writerow(line)
```

Annexe 11. Étape 6, une fois les documents traités au format CONLL (suppression des colonnes superflues) format : [Token] \t [Label]. Gestion des balises pour obtenir un format de détection d'erreurs, c'est-à-dire balises binaires 0 et 1.

```
#=====
# script : traitement_6.py
# auteur : Angela Kasparian
# date : janvier/fevrier 2023
#=====
# Ce script prend les documents par annotateurs avec des balises numériques une fois
# nettoyé (format du doc : [token] \t [labels]) et transforme les balises numériques
# en balises binaires 0 = correct, 1 = incorrect. |
#=====

import csv

with open('/Users/akaspari/Desktop/docs complets/batch 1/val_b1.tsv', 'r') as f:
    with open('/Users/akaspari/Desktop/docs complets/batch 1/val_truc.tsv', 'w') as l:
        r = csv.reader(f, delimiter='\t', quotechar='')
        w = csv.writer(l, delimiter='\t')
        for line in r:
            tag=line[1]
            if line[1] != '0':
                line[1] = '1'
                w.writerow(line)
            else:
                w.writerow(line)
```

Annexe 12. Codes python pour le calcul d'accord global entre les 5 annotateurs et le Gold standard via la formule de l'alpha de krippendorff (module alpha du package Krippendorff « based on Thomas Grill implémentation » source : <https://pypi.org/project/krippendorff/> consulté le 19 mai 2023.)

```
1 #=====
2 # script : accord_global.py
3 # auteur : Angela Kasparian
4 # date : samedi 19 mai 2023
5 #=====
6 # ce script permet sur base d'un fichier CSV contenant les tokens puis une colonne
7 # par annotateur (y compris le GOLD) de calculer 2 accords grace à la formule de
8 # l'alpha de Krippendorff.
9 # résultat 1 : l'accord entre les 5 annotateurs ET le GOLD
10 # résultat 2 : l'accord entre les 5 annotateurs seulement
11 #=====
12 import pandas as pd
13 from krippendorff import alpha
14
15 # Chargement du fichier CSV
16 df = pd.read_csv('/Users/akaspari/Desktop/cours/stage/batch_gold/calcul_3.csv', sep=';', header=None, names=['token',
17                                                         'balise_gold',
18                                                         'balise_annotation1',
19                                                         'balise_annotation2',
20                                                         'balise_annotation3',
21                                                         'balise_annotation4', 'balise_annotation5'])
22
23 # Sélection des colonnes d'annotations
24 annotation_columns_global = ['balise_gold', 'balise_annotation1', 'balise_annotation2', 'balise_annotation3', 'balise_annotation4', 'balise_annotation5']
25 annotations1 = df[annotation_columns_global].values.T.tolist()
26
27 # Calcul de l'accord entre les annotateurs
28 alpha_value1 = alpha(annotations1, level_of_measurement='nominal')
29
30 # Affichage des résultats
31 print("Krippendorff's alpha with Gold:", alpha_value1)
32
33 # Sélection des colonnes d'annotations
34 annotation_columns = ['balise_annotation1', 'balise_annotation2', 'balise_annotation3', 'balise_annotation4', 'balise_annotation5']
35 annotations2 = df[annotation_columns].values.T.tolist()
36
37 # Calcul de l'accord entre les annotateurs
38 alpha_value2 = alpha(annotations2, level_of_measurement='nominal')
39
40 # Affichage des résultats
41 print("Krippendorff's alpha without Gold:", alpha_value2)
42
```

Annexe 13. Codes python permettant de calculer l'accord entre chacun des annotateurs et le Gold standard, mais aussi de calculer le pourcentage d'accord entre chacun des annotateurs et le Gold Standard. (module alpha du package Krippendorff « based on Thomas Grill implémentation » *source* : <https://pypi.org/project/krippendorff/> consulté le 19 mai 2023 et module confusion_matrix du package sklearn.metrics pour calculer le pourcentage d'accord.)

```
1 #=====
2 # script : code_gold.py
3 # auteur : Angela Kasparian
4 # date : samedi 19 mai 2023
5 #=====
6 # ce script permet sur base d'un fichier CSV contenant les token puis une colonne
7 # par annotateur (y compris le GOLD) de calculer en changeant l'annotateur concerné
8 # de calculer et l'accord entre cet annotateur et le GOLD mais aussi le pourcentage
9 # d'accord entre les deux.
10 # il retourne aussi la matrice de contingence entre l'annotateur choisi et le GOLD.
11 #=====
12
13
14 import pandas as pd
15 import numpy as np
16 from sklearn.metrics import confusion_matrix
17 from krippendorff import alpha
18
19 # Chargement du fichier CSV
20 df = pd.read_csv('/Users/akaspari/Desktop/cours/stage/batch_gold/calcul_3.csv', sep=';', header=None, names=['token',
21 'balise_gold',
22 'balise_annotation1',
23 'balise_annotation2',
24 'balise_annotation3',
25 'balise_annotation4', 'balise_annotation5'])
26
27 # Calcul de l'accord entre les deux fichiers d'annotation
28 data = [list(df['balise_gold']), list(df['balise_annotation1'])]
29 alpha_value = alpha(data, level_of_measurement='nominal')
30
31
32 # Calcul de la matrice de contingence
33 matrice_contingence = confusion_matrix(df['balise_annotation1'], df['balise_gold'])
34
35
36
37 # Affichage des résultats
38 print('Krippendorff\'s alpha :', alpha_value)
39 print('Matrice de contingence :')
40 print(matrice_contingence)
41 print('Pourcentage d\'accord :', np.sum(np.diag(matrice_contingence)) / np.sum(matrice_contingence) * 100)
42
```

Annexe 14. Code Python de la création, entraînement et évaluation du modèle baseline pour la classification binaire : SVM (machine à vecteur de support). Le code fait appel au module GridSearchCV pour l'optimisation des paramètres et les métriques d'évaluation proposée par la librairie sklearn. Il utilise aussi les embeddings (plongement lexicaux) disponibles via Gensim.

```

1  #####
2  # script : SVM_binaire.py
3  # auteur : Angela Kasparian
4  # date : Juin 2023
5  #####
6  # Ce script permet la création, l'entraînement et l'évaluation d'un modèle de
7  # classification par apprentissage de surface : un modèle à vecteur de support "SVM"
8  # L'optimisation du modèle se fait par le module GridSearchCV.
9  #####
10
11 import pandas as pd
12 import numpy as np
13 from sklearn.svm import SVC
14 from sklearn.model_selection import GridSearchCV
15 from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score
16 from sklearn.preprocessing import LabelEncoder
17 from sklearn.utils import resample
18 from gensim.models import FastText
19
20
21
22 # Charger les données du jeu de données
23 train_path = '/Users/akaspari/Desktop/cours/docs complets/TRAIN SVM/SVM_TRAIN_binaire.tsv'
24 test_path = '/Users/akaspari/Desktop/cours/docs complets/TEST/FORMAT_TEST_binaire.tsv'
25
26 train_df = pd.read_csv(train_path, delimiter='\t', names=['token', 'label'])
27 test_df = pd.read_csv(test_path, delimiter='\t', names=['token', 'label'])
28
29 train_texts = train_df['token']
30 train_labels = train_df['label']
31 test_texts = test_df['token']
32 test_labels = test_df['label']
33
34 # Encoder les étiquettes en entiers
35 label_encoder = LabelEncoder()
36 train_labels = label_encoder.fit_transform(train_labels)
37 test_labels = label_encoder.transform(test_labels)
38
39 # Créer le modèle FastText
40 embedding_size = 200 # Dimension de l'espace d'embedding
41 window_size = 5 # Taille de la fenêtre du contexte
42 min_count = 0 # Nombre minimal d'occurrences pour un mot
43
44 fasttext_model = FastText(vector_size=embedding_size, window=window_size, min_count=min_count)
45 fasttext_model.build_vocab(corpus_iterable=train_texts)
46 fasttext_model.train(corpus_iterable=train_texts, total_examples=len(train_texts), epochs=10)
47
48
49 # Obtenir les embeddings de texte pour les données d'entraînement et de test
50 train_embeddings = np.array([fasttext_model.wv[token] for token in train_texts])
51 test_embeddings = np.array([fasttext_model.wv[token] for token in test_texts])
52
53 print("embedding done")
54
55
56 # optimisations des paramètres
57 param_grid = {
58     'C': [0.1, 1, 10],
59     'kernel': ['linear', 'rbf'],
60     'gamma': [0.01, 0.1, 1]
61 }
62
63 # Créer le modèle SVM
64 basic_svm = SVC()
65
66 # Créer l'objet GridSearchCV
67 grid_search = GridSearchCV(estimator=basic_svm, param_grid=param_grid, scoring='accuracy', cv=3)
68
69 # Effectuer la recherche sur grille
70 grid_search.fit(train_embeddings, train_labels)
71
72 # Obtenir les meilleurs hyperparamètres et le meilleur score
73 best_params = grid_search.best_params_
74 best_score = grid_search.best_score_
75
76 print(best_params, best_score)
77
78 # Entraînement et prédiction pour le jeu de données
79 svm = SVC(**best_params, class_weight='balanced')
80 print("start training")
81 svm.fit(train_embeddings, train_labels)
82 predictions = svm.predict(test_embeddings)
83
84 # Calcul des métriques
85 accuracy = accuracy_score(test_labels, predictions)
86 f1 = f1_score(test_labels, predictions, average='weighted')
87 precision = precision_score(test_labels, predictions, average='weighted')
88 recall = recall_score(test_labels, predictions, average='weighted')

```

Annexe 15. code Python de la création et de l'entraînement du modèle BERT (CamemBERT) pour la classification des erreurs selon les 3 types de jeu de données, le type binaire (correct 0, incorrect 1), à 5 classes (les catégories linguistiques) et les toutes les erreurs de la typologie. Ce code a été supervisé par Rodrigo Wilkens.

```
1  #####
2  # script : bert_french_nclassesV5.py
3  # auteur : Rodrigo Wilkens (et Angela Kasparian)
4  # date : juin 2023
5  #####
6  # Ce script entraine un modèle BERT (CamemBert) pour la classification d'erreurs (soit
7  # binaire, soit à 5 classes, soit à 23 classes)
8  #####
9
10 ##### modele BERT pour le français #####
11 ##### entraînement à n classes, à faire tourner avec TOUT les fichiers (une fois avec) TRAIN_nombres, TEST_nombre, DEV_nobres (24 classes)
12 ##### ET les fichiers (une fois avec) TRAIN_classes, TEST_classes, et DEV_classes (5 classes)
13 ##### ET les fichiers (une fois avec) TRAIN_binaire, TEST_binaire et DEV_binaire #####
14
15
16 import argparse
17 import pandas as pd
18 import numpy as np
19 import tensorflow as tf
20 from transformers import TFCamembertForSequenceClassification, CamembertTokenizer
21 from sklearn.preprocessing import LabelEncoder
22 from sklearn.metrics import f1_score
23 import glob
24 import os
25 from pathlib import Path
26 import tensorflow as tf
27 from tensorflow.python.client import device_lib
28 from sklearn.utils import class_weight
29 from tensorflow.keras.callbacks import TensorBoard
30 from sklearn import metrics
31
32 # Parseur d'arguments
33 parser = argparse.ArgumentParser()
34 parser.add_argument('--learning_rate', type=float, help='Learning rate')
35 parser.add_argument('--epoch', type=int, help='epoch')
36
37 args = parser.parse_args()
38
39 def set_gpu():
40     os.environ["CUDA_DEVICE_ORDER"] = "PCI_BUS_ID" # see issue #152
41     gpus = tf.config.list_physical_devices('GPU')
42     print("*"*80)
43
```

```

44 # Restrict TensorFlow to only use the first GPU
45 pu_device = None
46 try:
47     print(device_lib.list_local_devices())
48     if gpus:
49         print("Num GPUs Available: ", len(gpus))
50         tf.config.set_visible_devices(gpus[0], 'GPU')
51         logical_gpus = tf.config.list_logical_devices('GPU')
52         print(len(gpus), "Physical GPUs,", len(logical_gpus), "Logical GPU")
53         pu_device = '/GPU:0'
54     else:
55         print("no gpu", gpus)
56 except:
57     print("No GPU found")
58 print("*"*80)
59 return pu_device
60
61 class DummyCheckpoint(tf.keras.callbacks.Callback):
62     def __init__(self, directory, delete_old=True):
63         super().__init__()
64         if not os.path.exists(directory):
65             os.mkdir(directory)
66         self.directory = directory
67     def on_epoch_end(self, epoch, logs=None):
68         self.model.save_weights(Path(self.directory, str(epoch) + ""))
69
70     def on_train_end(self, logs=None):
71         self.model.save_weights(Path(self.directory, "final" + ""))
72
73 pu_device = set_gpu()
74
75 # Chemins d'accès aux fichiers de données
76 train_path = '/CECI/trsf/users/w/i/wilkens/angela/FORMAT_TRAIN_binaire.tsv'
77 dev_path = '/CECI/trsf/users/w/i/wilkens/angela/FORMAT_DEV_binaire.tsv'
78 test_path = '/CECI/trsf/users/w/i/wilkens/angela/FORMAT_TEST_binaire.tsv'
79
80 epochs = args.epochs
81
82 print("train_path", train_path)
83 print("dev_path", dev_path)
84 print("test_path", test_path)
85 print("epochs", epochs)
86 print("learning_rate", args.learning_rate)

```

```

88 # Charger les données d'entraînement
89 train_df = pd.read_csv(train_path, delimiter='\t', names=['token', 'label'])
90 train_labels = train_df['label']
91 train_texts = train_df['token']
92
93 # Charger les données de validation
94 dev_df = pd.read_csv(dev_path, delimiter='\t', names=['token', 'label'])
95 dev_labels = dev_df['label']
96 dev_texts = dev_df['token']
97
98 # Charger les données de test
99 test_df = pd.read_csv(test_path, delimiter='\t', names=['token', 'label'])
100 test_labels = test_df['label']
101 test_texts = test_df['token']
102
103
104 with tf.device(pu_device):
105     # Encoder les étiquettes en entiers
106     label_encoder = LabelEncoder()
107     train_labels = label_encoder.fit_transform(train_labels)
108     dev_labels = label_encoder.transform(dev_labels)
109     test_labels = label_encoder.transform(test_labels)
110
111     # Charger le tokenizer et encoder les textes en vecteurs d'entiers
112     tokenizer = CamembertTokenizer.from_pretrained('camembert-base', do_lower_case=True)
113     train_encodings = tokenizer(train_texts.tolist(), truncation=True, padding=True)
114     dev_encodings = tokenizer(dev_texts.tolist(), truncation=True, padding=True)
115     test_encodings = tokenizer(test_texts.tolist(), truncation=True, padding=True)
116
117     # Créer les jeux de données TensorFlow
118     train_dataset = tf.data.Dataset.from_tensor_slices((dict(train_encodings), train_labels)).shuffle(len(train_df)).batch(32)
119     dev_dataset = tf.data.Dataset.from_tensor_slices((dict(dev_encodings), dev_labels)).batch(32)
120     test_dataset = tf.data.Dataset.from_tensor_slices((dict(test_encodings), test_labels)).batch(len(test_df))
121
122     # calculer le poids des classes
123     class_weights = class_weight.compute_class_weight('balanced', np.unique(train_labels), train_labels)
124     class_weights = dict(enumerate(class_weights))
125
126     # Charger le modèle Camembert pré-entraîné pour la classification de séquences
127     model = TFCamembertForSequenceClassification.from_pretrained('camembert-base', num_labels=len(label_encoder.classes_))
128

```

```

129 # Compiler le modèle
130 optimizer = tf.keras.optimizers.Adam(learning_rate=args.learning_rate)
131 model.compile(optimizer=optimizer, loss=model.compute_loss, metrics=['accuracy'])
132
133 # Create some useful log tools
134 # tsboard = TensorBoard(log_dir="logs_TensorBoard", write_graph=True, histogram_freq=1, embeddings_freq=1, profile_batch="1,5")
135 csvlog = tf.keras.callbacks.CSVLogger("logs.csv", separator="\t", append=False)
136 # deval = DummyEval(directory = None, dataset=dev_dataset, dataset_name="DevDS", cls=np.unique(train_labels), data_columns=None)
137 # detrain = DummyEval(directory = None, dataset=train_dataset, dataset_name="TrainDS", cls=np.unique(train_labels), data_columns=None)
138 # detest = DummyEval(directory = None, dataset=test_dataset, dataset_name="TestDS", cls=np.unique(train_labels), data_columns=None)
139
140
141 # Entraîner le modèle
142 history = model.fit(train_dataset, epochs=epochs, validation_data=dev_dataset, class_weight=class_weights, callbacks=[csvlog])
143 ckpt = DummyCheckpoint(directory = "/CECI/trsf/users/w/i/wilkens/angela/ckpt")
144 history = model.fit(train_dataset, epochs=epochs, validation_data=dev_dataset, class_weight=class_weights, callbacks=[ckpt])
145 print("evaluation")
146
147 # Évaluer le modèle sur le jeu de test
148 for model_path in glob.glob("/CECI/trsf/users/w/i/wilkens/angela/ckpt/*.index"):
149     with open("/CECI/trsf/users/w/i/wilkens/angela/preds_" + model_path.split("/")[-1].replace(".index", "") + ".txt", "w") as output_file:
150         model.load_weights(model_path.replace(".index", ""))
151         y_pred = model.predict(test_dataset)
152         for r, p in zip(test_labels, y_pred.logits):
153             output_file.write(str(p) + "\t" + str(r) + '\n')
154         # print(r,p)
155

```

Annexe 16. Code Python pour l'analyse des résultats produits par les différents modèles BERT (classification binaire, en cinq classes ou à 23 classes). Code créé par Rodrigo Wilkens pour l'analyse des performances du modèle BERT par classes.

```
1  #####
2  # script : eval_error_detection.py
3  # auteur : Rodrigo Wilkens (et Angela Kasparian)
4  # date : juin 2023
5  #####
6  # Ce script prend les documents extrait de l'evaluation des différents modèles BERT
7  # (classification binaire, à 5 classes et à 23 classes) sous forme de prédiction LOGIT
8  # et évalue les performances de la classification par le modèle avec les balises prédites
9  # et les balises réelles du corpus de test. Le script produit des mesures d'évaluation
10 # par classes (2, 5 ou 23).
11 #####
12
13 import glob
14 import numpy as np
15 import pandas as pd
16 from sklearn.metrics import classification_report
17
18 def load_preds(preds_path, binaire = False):
19     ret = {}
20     for path in glob.glob(preds_path):
21         print(path)
22         # df = pd.read_csv(path, sep="\s(\t|\\[|\\])")
23         # print(df)
24         lst = []
25         with open(path) as f:
26             #sequence control
27             state = "close"
28             prev_newline = False
29             col = 0
30
31             #infomatation
32             col1 = ""
33             col2 = ""
34             while True:
35                 c = f.read(1)
36                 # print(c, state, prev_newline, col)
37                 if not c:
38                     break #End of file
39                 if c == "[":
40                     state = "open"
41                     prev_newline = False
42                 elif c == "]":
43                     state = "closed"
44                     prev_newline = False
45                 elif c == "\t":
```

```

46         col = 1
47         prev_newline = False
48     elif c == "\n":
49         prev_newline = True
50         # col = 0
51         if state=="closed" and prev_newline:
52             col1 = [float(v) for v in col1.replace("\n","").split(" ") if len(v)>0]
53             col2 = [float(v) for v in col2.replace("\n","").split(" ") if len(v)>0]
54             if not binaire:
55                 assert len(col1) == len(col2)
56                 lst.append(col1.index(max(col1)))
57                 # print(len(col1), "##", len(col2), col2)
58                 col1 = ""
59                 col2 = ""
60                 col = 0
61         else:
62             if col == 0 and state=="open":
63                 col1 += c
64             elif col == 1 and state=="open":
65                 col2 += c
66             # elif state=="closed" and prev_newline:
67             #     col1 = [v for v in col1.replace("\n","").split(" ") if len(v)>0]
68             #     col2 = [v for v in col2.replace("\n","").split(" ") if len(v)>0]
69             #     print(len(col1), "##", len(col2), col2)
70             #     col1 = ""
71             #     col2 = ""
72             #     col = 0
73         ret[path] = lst
74     return ret
75
76 def load_gold(path):
77     words = []
78     gold = []
79     with open(path) as input_file:
80         for ln in input_file:
81             wrd, ans = ln.split("\t")
82             words.append(wrd)
83             gold.append(float(ans))
84     return words, gold
85
86
87 # selet the sample by decommenting
88 gold_path = "/Users/akaspri/Desktop/cours/docs complets/TEST/TEST_nombres.tsv"
89 preds_path = "/Users/akaspri/Downloads/CECI 3/trsf/users/w/i/wilkens/angela/preds_final_nombres.txt"
90 binaire = False

```

```

91
92 # gold_path = "/Users/akaspri/Desktop/cours/docs complets/TEST/TEST_classes.tsv"
93 # preds_path = "/Users/akaspri/Downloads/CECI 2/trsf/users/w/i/wilkens/angela/preds_final_classes.txt"
94 # binaire = False
95
96 # gold_path = "/Users/akaspri/Desktop/cours/docs complets/TEST/FORMAT_TEST_binaire.tsv"
97 # preds_path = "/Users/akaspri/Downloads/CECI/trsf/users/w/i/wilkens/angela/preds_final.txt"
98 # binaire = True
99
100 wrds, gold = load_gold(gold_path)
101 dict_preds = load_preds(preds_path, binaire=binaire)
102
103 for epoch in dict_preds:
104     print(epoch)
105     assert len(dict_preds[epoch]) == len(gold), str(len(gold)) + " != " + str(len(dict_preds[epoch]))
106     print(classification_report(y_true=gold, y_pred=dict_preds[epoch], zero_division=False))
107
108
109 print("checking some cases using", epoch)
110 for index, (w, g, p) in enumerate(zip(wrds, gold, dict_preds[epoch])):
111     print(index, w, g, p)
112     if index == 5:
113         break

```

Annexe 17. Tableau représentant les résultats de l'évaluation du modèle BERT entraîné à la classification multi-classe à 24 balises, soit une par type d'erreur de la typologie. Évaluation en termes de précision, rappel et mesure F1 après 9 epoch pour un Learning rate de 1e-6.

		Précision	Rappel	Mesure F1	Support
9 epoch	0	88	2	3	34440
	1	0	0	0	28
	2	15	4	6	375
	3	3	73	6	112
	4	29	8	12	1533
	5	17	75	28	1246
	6	2	0	1	782
	7	2	17	3	105
	8	3	93	5	253
	9	1	1	1	374
	10	5	51	10	361
	11	3	7	5	451
	12	0	50	1	30
	13	0	0	0	344
	14	0	1	1	360
	15	0	0	0	72
	16	1	13	1	115
	17	0	0	0	8
	18	1	4	1	51
	19	4	1	1	593
	20	0	0	0	210
	21	3	26	5	881
	22	1	23	1	53
	23	4	3	3	191

Annexe 18. Tableau reprenant la répartition des niveaux de langues (niveaux du CECR) selon les 191 textes du Gold-standard. La division des textes du corpus de base (FEI) pour l’annotation en erreurs d’orthographe selon la typologie présentée ici a été réalisée par Rodrigo Wilkens. Cette répartition est représentative de la répartition des textes par niveaux dans chacun des échantillons annotés. Les textes les plus présents sont les niveaux A1 et A2, ce qui explique certaines des fautes propres aux apprenants de bas niveaux. Il est intéressant de noter que le niveau C2 est sous représenté, ce qui rend l’analyse d’erreurs inapplicable aux apprenants de niveaux supérieurs (dont les erreurs se rapprochent des erreurs des natifs).

Niveaux CECR	Pourcentage de textes du Gold standard
A1	21 %
A2	24 %
B1	17 %
B2	13 %
C1	18 %
C2	7 %

Partie III. Glossaire.

Glossaire :

[1] *Accord inter-annotateur :*

« Mesure de la fréquence à laquelle différents [annotateurs] humains sont d'accord lors de l'exécution d'une tâche. Si les [annotateurs] sont en désaccord, une amélioration des instructions peut être nécessaire. Parfois également appelé accord inter-[évaluateurs] ou fiabilité inter-évaluateurs ».

Source : https://datafranca.org/wiki/Accord_inter-%C3%A9valuateurs (dernière visite le 11 mars 2023)

[2] *Accuracy :*

L'accuracy, ou précision globale, est une mesure couramment utilisée pour évaluer la performance d'un modèle de classification. Elle représente la proportion d'exemples correctement classés par rapport au nombre total d'exemples dans le jeu de données.

L'accuracy est calculée en divisant le nombre d'exemples correctement classés par le modèle (vrais positifs et vrais négatifs) par le nombre total d'exemples dans le jeu de données.

Source : <https://kobia.fr/classification-metrics-accuracy/> (dernière visite le 7 août 2023)

[3] *Apprentissage supervisé :*

L'apprentissage supervisé est une technique d'apprentissage automatique dans laquelle un modèle est entraîné à partir d'un ensemble de données d'entraînement étiquetées : les données d'entrée et données de sorties (sortie attendue du modèle). Le processus d'apprentissage supervisé commence par l'analyse de l'ensemble de données d'entraînement étiquetées pour identifier des modèles et des relations entre les entrées et les sorties. Ensuite, le modèle est ajusté pour prédire la sortie souhaitée pour de nouvelles entrées. (Fonction mathématique optimisée pour minimiser l'erreur dans la production du modèle).

Source : cours de traitement automatique du langage naturel (LFIAL2620)

[4] *Baseline :*

La « baseline » d'un projet de modélisation IA (Intelligence Artificielle) est la référence ou le point de départ à partir duquel les performances seront mesurées. Elle peut être définie comme une mesure de performance initiale ou un état actuel de la situation qui doit être amélioré.

[5] *Bi-LSTM (ou LSTM bidirectionnels)* :

Bi-LSTM est une architecture de réseau de neurones artificiels utilisée pour résoudre des tâches d'apprentissage automatique impliquant des séquences de données, telles que le traitement du langage naturel (NLP) et la reconnaissance vocale. La structure du Bi-LSTM repose sur les LSTM (Long Short-Term Memory), un type de réseau de neurones récurrents (RNN). Le Bi-LSTM résout le problème du LSTM simple (« vanishing gradient ») en utilisant deux LSTM : un LSTM qui traite la séquence dans l'ordre direct et un LSTM qui traite la séquence dans l'ordre inverse.

Source : Speech and Language Processing. Daniel Jurafsky & James H. Martin. Copyright © 2023. All rights reserved. Draft of January 7, 2023. (chapitre 9).

[6] *Architecture Transformers (BERT)* :

L'architecture des Transformers dans BERT est une architecture de réseau de neurones utilisée pour la compréhension de texte en utilisant une approche d'apprentissage non supervisé pour entraîner le modèle sur de grandes quantités de données textuelles. L'architecture des Transformers permet de capturer les dépendances sémantiques à long terme dans le texte, ce qui permet d'obtenir de meilleures performances sur les tâches de classification de texte ou d'étiquetage de séquences (modèle utilisé pour des tâches supervisées une fois entraîné).

Speech and Language Processing. Daniel Jurafsky & James H. Martin. Copyright © 2023. All rights reserved. Draft of January 7, 2023. (chapitre 10).

[7] *Ensemble learning* :

Apprentissage ensembliste, en français, consiste en la combinaison de différents algorithmes d'apprentissage automatique pour optimiser les performances.

Source : Géron, A., 2019. « Machine-learning avec Scikit-learn », Dunod, Malakoff, pp. 181-202.

[8] *Fine-tuning* :

Le fine-tuning est une technique d'apprentissage automatique utilisée pour adapter un modèle de machine learning pré-entraîné à une tâche spécifique. Le modèle pré-entraîné est ajusté ou "fine-tuné" pour une tâche spécifique en entraînant le modèle sur un ensemble de données de formation pour cette tâche spécifique. Cette phase d'ajustement modifie (optimise) les poids du modèle initial afin de maximiser les performances pour la tâche demandée.

Speech and Language Processing. Daniel Jurafsky & James H. Martin. Copyright © 2023. All rights reserved. Draft of January 7, 2023. (chapitre 11).

[9] *Gensim* :

Gensim est une bibliothèque open source en Python qui est principalement utilisée pour la modélisation de thèmes (topic modeling) et le traitement des langues naturelles (NLP - Natural Language Processing).

Source : <https://pypi.org/project/gensim/>
(dernière visite le 7 août 2023).

[10] *Gold standard* :

Le Gold standard fait référence à une norme, un étalon de qualité ou de mesure dans un domaine particulier. Dans le cas du TAL, il désigne un ensemble de données annotées de manière précise et cohérente, qui sert de référence pour évaluer la performance des modèles de TAL.

[11] *HuggingFace* :

Hugging Face est une entreprise spécialisée dans le développement de technologies d'intelligence artificielle pour le traitement du langage naturel. Elle est notamment connue pour avoir créé une bibliothèque open-source de modèles de traitement du langage naturel appelée « Transformers ».

Source : <https://huggingface.co/huggingface>
(dernière visite le 7 août 2023)

[12] *Keras* :

Keras est une bibliothèque de haut niveau qui permet aux utilisateurs de créer des modèles de Deep Learning avec une syntaxe simple et facile à utiliser. Keras permet aux utilisateurs de définir des réseaux de neurones en empilant des couches de traitement des données (Keras en tant qu'API utilise Tensorflow comme backend).

Source : <https://github.com/keras-team/keras> (section « about keras »).
(dernière visite le 7 août 2023).

[13] *Mécanismes d'attention* :

Méthode clé employée dans les architectures Transformers pour comprendre les relations entre les mots dans une phrase en accordant plus d'attention aux mots les plus pertinents (en accordant différents poids aux mots en fonctions de la tâche en cours sans être limité à une fenêtre de contexte fixe). En somme, la méthode d'attention permet d'obtenir des résultats plus précis et de mieux comprendre la structure sémantique des phrases.

Speech and Language Processing. Daniel Jurafsky & James H. Martin.
Copyright © 2023. All rights reserved. Draft of January 7, 2023.
(chapitre 10).

[14] *N-grams* :

Modèle de prédiction d'une séquence de lettres ou de mots selon l'hypothèse d'horizons limités de Markov, c'est-à-dire que la séquence prédite dépend uniquement des n séquences précédentes.

Source : Speech and Language Processing. Daniel Jurafsky & James H. Martin. Copyright © 2023. All rights reserved. Draft of January 7, 2023. (chapitre 3).

[15] *Plongements lexicaux (embeddings)* :

L'idée derrière les plongements lexicaux est de représenter chaque mot par un vecteur de nombres réels qui capture les relations sémantiques entre les mots. Les plongements lexicaux sont appris à partir de grandes quantités de données textuelles en utilisant des algorithmes d'apprentissage automatique tels que *Word2Vec* [28] ou *GloVe*. En utilisant les plongements lexicaux, les algorithmes de traitement du langage naturel peuvent comprendre la signification des mots en se basant sur leur contexte d'utilisation.

Source : Speech and Language Processing. Daniel Jurafsky & James H. Martin. Copyright © 2023. All rights reserved. Draft of January 7, 2023. (chapitre 6).

[16] *POS (classe grammaticale)*:

Le POS, ou Part-Of-Speech, est une abréviation qui désigne la catégorie grammaticale à laquelle appartient un mot dans une phrase. En français, les catégories grammaticales principales sont le nom, le verbe, l'adjectif, l'adverbe, la préposition, le pronom, la conjonction et la particule. Chacune de ces catégories grammaticales a des propriétés distinctes qui déterminent son rôle dans la phrase.

[17] *Précision* :

La précision est une mesure de performance utilisée en classification pour évaluer la capacité d'un modèle à prédire correctement les exemples positifs parmi toutes les prédictions positives qu'il fait. La précision est calculée en divisant le nombre d'exemples positifs correctement prédits par le modèle (vrais positifs) par le nombre total d'exemples prédits comme positifs par le modèle (vrais positifs + faux positifs).

Source : <https://kobia.fr/classification-metrics-precision-recall/>
(dernière visite le 7 août 2023)

[18] *Random Forest* :

Algorithme d'apprentissage supervisé utilisé spécifiquement pour des tâches de classification et de régression. Il consiste en un ensemble d'arbres de décision où chaque arbre est entraîné sur un échantillon aléatoire du corpus d'entraînement. L'idée étant de minimiser les biais et variations présents dans chaque arbre individuellement.

Source : Géron, A., 2019. « Machine-learning avec Scikit-learn », Dunod, Malakoff, pp. 181-202.

[19] *Rappel* :

Le rappel, également connu sous le nom de sensibilité ou taux de vrais positifs, est une mesure de performance utilisée en classification pour évaluer la capacité d'un modèle à identifier tous les exemples positifs réels.

Le rappel est calculé en divisant le nombre d'exemples positifs correctement prédits par le modèle (vrais positifs) par le nombre total d'exemples positifs réels dans les données (vrais positifs + faux négatifs).

Source : <https://kobia.fr/classification-metrics-precision-recall/>
(dernière visite le 7 août 2023)

[20] *Réseaux de neurones artificiels* :

Sous ensemble de l'apprentissage machine (apprentissage automatique), qui s'inspire du cerveau humain. L'architecture de cette intelligence artificielle s'articule donc en couches interconnectées qui apprennent les unes des autres (s'apparente à l'apprentissage profond ou deep learning).

Source : <https://www.ibm.com/fr-fr/cloud/learn/neural-networks>
(Dernière visite le 11 mars 2023).

[21] *Sklearn* :

Scikit-learn (ou sklearn) est une bibliothèque populaire en Python qui offre une large gamme d'outils et de fonctionnalités pour l'apprentissage automatique (machine learning). Elle fournit des fonctionnalités pour la préparation des données, la construction de modèles, l'évaluation des performances et bien plus encore.

Source : Géron, A., 2019. « Machine-learning avec Scikit-learn », Dunod, Malakoff.

[22] *Sequence labelling task* :

Une tâche de séquençage (sequence labelling task en anglais) est une tâche d'apprentissage automatique où l'objectif est de prédire une séquence de labels pour une séquence d'entrée donnée. Les séquences d'entrée peuvent être des phrases, des séquences de mots, des séquences de symboles ou tout autre type de séquence (souvent utilisée pour résoudre des problèmes de traitement du langage naturel (NLP), tels que la reconnaissance d'entités nommées, la segmentation de phrases, la catégorisation de parties du discours ...).

Source : <https://medium.com/artiwise-nlp/sequence-labeling-a90f2b60bb2a>

(dernière visite le 7 août 2023)

[23] *SVM (machine à vecteur de support) :*

Algorithme d'apprentissage supervisé utilisé spécifiquement pour des tâches de classification et de régression. Le fonctionnement d'un SVM réside dans la recherche d'un hyperplan (linéaire) permettant de séparer efficacement les données. L'entraînement d'un SVM consiste à maximiser au fur et à mesure l'hyperplan en réduisant les erreurs de classification pour maximiser la prédiction de classes pour des données non vues.

Source : Géron, A., 2019. « Machine-learning avec Scikit-learn », *Dunod, Malakoff*, pp. 145-165.

[24] *Tensorflow:*

TensorFlow est une bibliothèque de bas niveau qui offre une plus grande flexibilité pour construire des modèles de Deep Learning. TensorFlow est plus axé sur les opérations mathématiques complexes nécessaires pour entraîner des modèles d'apprentissage en profondeur.

Source : <https://www.tensorflow.org/?hl=fr>
(dernière visite le 7 août 2023).

[25] *Token :*

En informatique, un token est une unité élémentaire de données ou de texte. En règle générale, on simplifie la définition de token à celle de mot (aussi complexe soit-elle). Mais selon la tâche, un token peut aussi être un signe de ponctuation, une syllabe, un syntagme

Source : cours de traitement automatique du langage naturel (LFIAL2620).

[26] *Tokenizer :*

La tokenisation est le processus de conversion d'un texte en une séquence de tokens. C'est une étape importante dans le traitement de données textuelles, car elle permet de diviser le texte en unités plus petites et significatives qui peuvent ensuite être analysées et traitées par un ordinateur. La tokenisation peut être effectuée à l'aide d'expressions régulières, de modèles de langage, de dictionnaires de mots, ou de techniques de machine learning comme le traitement de langage naturel (NLP) ou le deep learning.

Source : cours de traitement automatique du langage naturel (LFIAL2620).

[27] *Valence verbale :*

La valence verbale en français se réfère au nombre d'arguments (sujets, objets, compléments, etc.) qu'un verbe peut prendre. Autrement dit, c'est le nombre de personnes ou de choses impliquées dans une action décrite par le verbe.

[28] *Word2vec (W2V)* :

Développé par les chercheurs de Google en 2003, Word2Vec est un algorithme d'apprentissage automatique utilisé pour apprendre des plongements lexicaux. Les plongements lexicaux obtenus sont des vecteurs de nombres réels de dimension réduite, où chaque dimension représente une caractéristique sémantique ou syntaxique du mot.

Source : cours de séminaire de linguistique computationnelle (LCLIG2140).

[29] F-bêta mesures :

La mesure F-bêta est une métrique utilisée pour évaluer la performance d'un modèle de classification binaire. Elle combine la précision (capacité du modèle à prédire correctement les exemples positifs) et le rappel (capacité du modèle à identifier tous les exemples positifs). Une valeur de bêta de 1 donne une mesure F1, qui équilibre la précision et le rappel de manière égale. Si la précision est importante, on peut utiliser une bêta inférieure à 1 (par exemple, 0,5) pour donner plus de poids à la précision. Si le rappel est plus important, on peut utiliser une bêta supérieure à 1 (par exemple, 2) pour donner plus de poids au rappel.

Source : <https://machinelearningmastery.com/fbeta-measure-for-machine-learning/>

(dernière visite le 7 août 2023).

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté de philosophie, arts et lettres

Place Blaise Pascal, 1 bte L3.03.11, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/fial