

Faculté	des Langues
	Université de Strasbourg

Master Technologie des Langues
Parcours Traitement Automatique des Langues

2022-2024

Analyse computationnelle de textes de théâtre français
Classification automatique du type de didascalies

Alexia Schneider

Mémoire de master

Sous la direction de
Pablo Ruiz Fabo

Remerciements

Je tiens à remercier Pablo Ruiz pour avoir encadré ce projet de mémoire avec toute la patience et l'attention aux détails qu'on lui connaît, mais aussi et surtout, pour ses conseils avisés et son précieux mentorat.

Je remercie aussi Fanny Boisnard, pour son soutien indéfectible et son optimisme à toute épreuve.

Table des matières

Remerciements	i
Table des tableaux	iv
Table des illustrations	v
Introduction	6
Partie 1 : État de l'art et définition du champ d'études	8
1.1. Définition et remarques historiques	8
1.2. Place de la didascalie dans les HN	10
1.2.1. Analyse en <i>distant reading</i>	10
1.2.2. Identification et classification automatique	11
1.3. Typologies et définition du champ d'études	13
1.3.1. Traits formels	13
1.3.2. Typologie selon le lieu d'apparition de la didascalie (Dompeyre, 1992)	14
1.3.3. Typologie des didascalies selon leur fonction (Issacharoff, 1981)	14
1.3.4. Typologie selon la définition de l'interaction linguistique (Galleron, 2021)	15
1.3.5. Enjeux propres à l'élaboration de la typologie pour ce projet	16
1.4. Outils et méthodes pour l'analyse computationnelle des didascalies	16
1.4.1. Standardisation et pratique d'annotation en XML	17
1.4.2. Approches pour l'automatisation de l'annotation	21
1.4.3. Outils pour la mise en place du projet	24
Partie 2 : Méthodologie	29
2.1. Guide d'annotation de la typologie générique proposée	29
2.1.1. Didascalies destinatrices	30
2.1.2. Didascalies mélodiques : <i>delivery</i>	32
2.1.3. Didascalies kinésiques	32
2.1.4. Didascalies visuelles scénographiques	34
2.1.5. Didascalies visuelles de jeu	36
2.1.6. Didascalies relatives à la musique : <i>music</i>	37
2.2. Typologie baseline	38
2.3. Conception des jeux de données	38
2.3.1. Extraction des didascalies et de leur attribut type	38
2.3.2. Création du jeu de données de la typologie principale : réannotation	39
2.3.3. Suppression des doublons et division des jeux de données pour l'entraînement, la validation et l'évaluation	40
2.3.4. Compte des occurrences par jeux de données	40
2.3.5. Comparaison des jeux de données par typologies	42
2.4. Entraînement des modèles d'apprentissage automatique	43
2.4.1. ML classique	43
2.4.2. Prétraitement des données pour la vectorisation	44
2.4.3. Apprentissage par transfert	46
2.4.4. Présentation des modèles et des hyperparamètres d'entraînement	47
2.4.5. Modèles génératifs	48
2.5. Métriques d'évaluation	52
Partie 3 : Résultats et discussion	53
3.1. Comparaison des performances par type de modèles	53
3.1.1. Modèles de ML classiques entraînés	54

3.1.2. Apprentissage par transfert	54
3.1.3. Modèles génératifs avec apprentissage avec et sans contexte	56
3.2. Comparaison de la classification selon les classes	60
3.2.1. Typologie principale.....	60
3.2.2. Typologie baseline.....	65
3.2.3. Perspectives d'amélioration pour les modèles génératifs.....	69
Conclusion	71
Références bibliographiques	72
Annexes	79
A. Rapports de classification.....	79
A.1. SVM linéaire (SGD entraîné)	79
A.2. camembert-base.....	80
A.3. distilcamembert-base.....	82
A.4. bert-base-multilingual-uncased	83
A.5. bert-base-multilingual-cased	84
A.6. distilbert-base-uncased.....	86
A.7. SetFit.....	87
A.8. gpt-3.5-turbo-0125	89
A.9. open-mixtral-8x7b	93
B. Exemples de sorties de Mistral-7B avec le prompt ReAct.....	99
C. Productions scientifiques reliées au mémoire.....	100

Table des tableaux

Tableau 1 Valeurs de @type en TEI-P5.....	18
Tableau 2 Exemples pour chacune des 13 classes de la typologie principale	30
Tableau 3 Table de correspondance entre les 13 classes de la typologie et les 87 types de FreDraCor	39
Tableau 4 Nombre de didascalies par type du jeu de données de la typologie principale.....	41
Tableau 5 Nombre de didascalies par jeu pour la typologie principale.....	41
Tableau 6 Nombre de didascalies par type du jeu de données de la typologie baseline.....	42
Tableau 7 Nombre de didascalies par jeu pour la typologie baseline sans doublon.....	42
Tableau 8 Prétraitement et vectorisation des données	45
Tableau 9 Nombre d'époques lors de l'affinage	48
Tableau 10 Récapitulatif des performances pour tous les modèles testés.....	53
Tableau 11 Moyenne-macro F1 des classifications des modèles de ML classique.....	54
Tableau 12 Moyenne-macro F1 des classifications des modèles affinés d'apprentissage par transfert.....	54
Tableau 13 Écart-type des moyennes-macros F1 des 5 entraînements sur la typologie principale des modèles d'apprentissage par transfert.....	55
Tableau 14 Nombre de tokens générés par les modèles génératifs pour les deux typologies	56
Tableau 15 Moyennes-macros F1 des classifications des modèles génératifs	58
Tableau 16 Nombre d'occurrences par type dans le jeu de test de la typologie principale.....	61
Tableau 17 Nombre d'occurrences par type dans le jeu de test de la typologie baseline.....	66

Table des illustrations

Figure 1 : Matrice de configuration d'une pièce à quinze scènes et sept personnages (A-G) (Schmidt et al., 2019)	11
Figure 2 Typologie principale avec 6 groupes et 13 étiquettes	29
Figure 3 Matrice de confusion SVM linéaire	63
Figure 4 Matrice de confusion camembert-base	63
Figure 5 Matrice de confusion bert-case-multilingual-uncased	63
Figure 6 Matrice de confusion distilcamembert-base.....	63
Figure 7 Matrice de confusion bert-base-multilingual-cased	63
Figure 8 Matrice de confusion distilbert-multilingual-cased	64
Figure 9 Matrice de confusion SetFit.....	64
Figure 10 Matrice de confusion SVM linéaire (SGD entraîné) pour la typologie baseline.....	67
Figure 11 Matrice de confusion distilcamembert-base pour la typologie baseline.....	68

Introduction

La didascalie est une indication scénique propre au théâtre. Il s'agit d'un élément textuel voire métatextuel intermédiaire entre la lecture et l'interprétation, c'est-à-dire, dans la plupart des cas, une indication utile à la mise en scène.

La didascalie ne fait l'objet d'étude pour elle-même que depuis quelques décennies. La distinction généralement acceptée encore aujourd'hui entre « texte primaire » désignant le corps des répliques et « texte secondaire » (Ingarden, 1973) pour désigner les didascalies, montre que, malgré l'attention portée par les dramaturges envers cette partie écrite de leur œuvre, notamment depuis la fin du XIX^e siècle, celle-ci ne fait que rarement l'objet d'étude à part entière.

En effet, la didascalie est perçue comme représentative non de l'intention de l'auteur, mais d'une mise en scène en particulier. À ce titre, elle est longtemps largement ignorée aussi bien par les personnes de théâtre que par les chercheurs en littérature. Ceux-ci remettent en question l'autorialité de cet élément ou n'en perçoivent pas l'intérêt sur le plan stylistique (Carlson, 1991). Dans les années 80, certains chercheurs revalorisent la didascalie comme part intégrante du texte dramatique en l'étudiant notamment comme objet linguistique (Issacharoff, 1981) et philologico-stylistique (Munkelt, 1987).

Plus récemment, la didascalie fait l'objet d'un regain d'attention dans le domaine des Humanités Numériques (HN, *digital humanities* ou *DH*). Les HN consistent à mettre en application des savoirs et technologies informatiques sur des questions propres aux sciences humaines et sociales. Dans le domaine de la littérature, en produisant de nouvelles données, les HN engendrent de nouvelles interprétations des textes sur la base d'analyses dites quantitatives. Dans le cas de textes dramatiques, la définition historique des genres et sous-genres (Fischer et al., 2017), les contours des rôles types (Glorieux, 2017), la caractérisation des personnages selon leur âge, genre et classe sociale (Krautter et al., 2022) ou encore le transfert des connaissances sur scène (Andresen et al., 2022) comptent parmi les nombreux thèmes porteurs.

Plus précisément, l'analyse computationnelle des didascalies a permis de dégager leur importance dans de nouveaux aspects de l'étude des textes dramatiques. Un exemple probant est celui des didascalies d'entrée et de sortie de personnage. Ces didascalies nous renseignent exactement sur la composition dramatique, c'est-à-dire l'occupation de la scène par les personnages. Puisque ces didascalies questionnent les éléments éditoriaux structurant la pièce, à savoir les unités prédéterminées que sont les actes et les scènes, elles redessinent nos connaissances sur la progression de l'action définie comme la rencontre de personnages sur scène. Elles ont notamment un intérêt dans le cas des pièces qui, soit ne proposent pas de découpage en acte et en scène, soit ont une configuration en scènes qui ne correspond pas à la composition dramatique réelle.

L'annotation automatique par des outils de Traitement Automatique des Langues (TAL) vise à produire de grandes quantités de données, et ainsi changer l'échelle des études menées jusqu'à présent en littérature. L'établissement d'une annotation systématique et prédictive pour

appuyer l'encodage des pièces a pour objectif l'établissement de corpus de grande envergure. Dans une perspective macroanalytique (Jockers, 2013), cela permet notamment l'intégration dans des études littéraires de textes dramatiques qui jusqu'à présent ont pu être négligés.

Relativement peu de travaux portent sur l'automatisation de l'annotation des didascalies. Indirectement, une étude sur un corpus allemand visant à affiner de *large language models* (LLMs) multilingues et monolingues pour l'identification des différents éléments structurels de pièces de théâtre a souligné la difficulté que représente l'identification des didascalies par rapport aux répliques (Pagel et al., 2021). Dans une autre étude, et sur un corpus allemand des XVII^e et XVIII^e siècles, une classification supervisée de quatre types de didascalies (*entrance, exit, aside, dead*) a produit des résultats satisfaisants à partir de l'annotation manuelle de 1 986 didascalies (Dennerlein, 2016). Enfin, Daria Maximova et Frank Fischer (2019) évoquent brièvement avoir développé un classifieur automatique pour les neuf valeurs de l'attribut *@type* proposées par la *Textual Encoding Initiative* (TEI) en langue russe à partir de l'annotation manuelle de dix-huit pièces.

Ce mémoire a pour objectif la comparaison de différentes méthodes de classification automatique à partir d'une typologie applicable à divers sujets d'étude en *Computational Literary Studies* (CLS). La contribution de ce projet porte ainsi sur trois plans : d'une part l'élaboration d'une typologie à partir des connaissances actuelles sur la didascalie, mais aussi à partir de la réalité du travail d'annotation en HN, d'autre part une réflexion sur la simplification du travail d'annotation à partir d'une annotation existante, et enfin la comparaison de différents classifieurs automatiques entraînés ou non spécifiquement sur nos données.

Ce mémoire a fait l'objet de plusieurs productions scientifiques reliées. Une communication tout d'abord, dans le cadre d'une journée d'étude (Schneider, 2023), puis une publication dans le cadre du *workshop* LaTeCH-CLfL (Schneider & Ruiz, 2024), mettent en avant les parties de ce mémoire relatives à l'élaboration de la typologie principale ainsi que les expérimentations menées à partir de cette typologie pour affiner des modèles d'apprentissage par transfert.

La première partie constitue l'état de l'art et la définition du champ d'études. La didascalie est abordée comme élément textuel et objet d'étude puis quelques typologies majeures issues du domaine littéraire sont présentées. Cette partie présente également les différentes stratégies d'annotation de la didascalie ainsi que des outils de TAL nécessaires à la mise en place du projet. La deuxième partie se concentre sur la méthodologie du projet avec l'établissement d'une typologie principale et d'une typologie dite « baseline » à partir de l'annotation manuelle du French Drama Corpus (FreDraCor) puis la conception des jeux de données et l'entraînement des modèles. Le résultat des expérimentations sous la forme d'une analyse tant qualitative que quantitative des classifications générées termine le mémoire.

Partie 1 : État de l'art et définition du champ d'études

Il s'agit dans un premier temps de ce mémoire de déterminer quels sont les éléments textuels qui entrent dans notre champ d'études puis de déterminer quelles typologies sont les plus pertinentes pour un projet de classification automatique.

Nous dégagerons de ces éléments théoriques relatifs au contenu sémantique de la didascalie ainsi que d'un état de l'art plus spécifique à la question de l'annotation dans le contexte des CLS les enjeux propres à notre projet.

1.1. Définition et remarques historiques

Le terme didascalie est emprunté au grec et peut se traduire littéralement par « enseignement ; notice ; instructions [sur la manière de jouer les pièces] »¹. Une didascalie est un élément textuel propre au genre théâtral : elle n'a pas vocation à être lue, mais interprétée par les acteurs ou (re)présentée par la mise en scène de manière plus générale. Dans son sens le plus large, la didascalie est tout ce qui dans une pièce se distingue du dialogal, ce qui inclut tous les éléments de titres, sous-titres et indications de genre et sous-genre (Dompeyre, 1992, p. 77).

Bien que la didascalie soit un élément propre au texte théâtral et obligatoire dans sa définition la plus large, il demeure que sa nature méta-textuelle fait d'elle un élément longtemps perçu comme accessoire. Pour la critique littéraire comme pour les dramaturges, et ce, jusqu'au XX^e siècle, la didascalie est subordonnée au dialogue : elle vise principalement à éclaircir la situation d'énonciation plutôt que de contribuer à l'élaboration du contenu énonciatif (Issacharoff, 1981, p. 813). Les didascalies, alors acceptées comme « texte secondaire » selon les termes d'Ingarden (1973), sont, dans cette perspective, remplaçables par le jeu ou la mise en scène à partir des répliques.

Cela explique qu'elles soient si peu présentes dans le théâtre classique, qui, se basant sur l'idéal aristotélicien, conçoit le dialogue comme autosuffisant. Autrement dit, les répliques du théâtre de la période classique contiennent toutes les indications nécessaires à la mise en scène du texte, même si ces indications ne sont pas explicitées hors des répliques.

Ces indications implicites sont nommées « didascalies internes », car elles se situent exclusivement dans les répliques, par opposition aux « didascalies externes » qui sont exclusivement en dehors des répliques. Les didascalies internes font aujourd'hui l'objet d'études à part entière. Ioana Galleron s'est notamment penchée sur le cas des didascalies internes (2018, 2021). Cependant, ce mémoire se concentre uniquement sur les didascalies explicites ou « externes », dans l'optique d'un travail de classification et non d'identification. Pour cette raison, nous appelons « didascalie » ce qui est en réalité une « didascalie externe », et précisons uniquement la qualité de didascalie « interne » si nécessaire.

La perception de la didascalie varie grandement selon l'époque à laquelle la pièce a été écrite. Nous avons relevé qu'au XVII^e siècle, suivant l'idéal de la tragédie grecque, les dramaturges

¹Trésor de la langue française informatisé, disponible sur : <http://stella.atilf.fr>. Consulté le 03/02/2023

de la période classique utilisent les didascalies externes de manière minimale, jusqu'au fameux exemple de l'unique indication scénique dans *Phèdre* de Racine : « *Elle s'assied.* » (Acte 1, scène 3). À l'inverse, au XX^e siècle, des auteurs tels que George Bernard Shaw publient des pièces largement dominées par des éléments méta-textuels (Pfister, 1988, p.14). La perception et l'usage des didascalies sont aussi déterminés par le genre d'appartenance de la pièce. Pour illustrer cela, citons deux genres normés de la même époque. Le vaudeville, d'une part, foisonne d'opportunités d'utiliser des didascalies : actions vives, comédie burlesque, revirements de situation liés à l'apparition, à la disparition ou au déguisement de personnages. La multiplication d'indications de jeu et de mouvement participe à la définition du genre et tout concourt à la présence de didascalies dans le vaudeville.

Bois-d'Enghien, *souriant.*

Il est revenu, mon Dieu, oui ; il est revenu... (*À part, gagnant la gauche en se passant piteusement la brosse dans les cheveux.*) Allons, ça va bien ! ça va très bien ! Moi qui étais venu pour rompre !... ça va très bien. (*Il s'assied à droite de la table.*)
(*Firmin sort, Marceline est remontée, Lucette s'est assise sur le canapé, à côté et à droite de Nini. Chenneviette est debout derrière le canapé.*)²

Le théâtre symboliste, d'autre part, à la même époque fait un usage bien différent de l'indication scénique.

Acte I, Sc 2.

Une salle dans le château.
*On découvre Arkël et Geneviève.*³

Ici, la didascalie d'entrée est aussi un marqueur stylistique symboliste : le verbe « découvrir » dans une tournure impersonnelle est préféré à la construction attendue avec « entrer » pour traduire textuellement l'hermétisme du symbolisme.

De manière plus précise, certaines didascalies font l'objet d'une codification et d'un usage en fonction de l'époque et du genre d'appartenance de la pièce. À titre d'exemple, l'emploi des didascalies d'entrée et de sortie a été indirectement déprécié par la règle de la liaison des scènes apparue au XVII^e siècle (Dougnet, 2015). En insistant sur la continuité, cette règle a privilégié le découpage en scène pour marquer la sortie d'un personnage plutôt que la didascalie de sortie.

À partir du XIX^e siècle, la didascalie devient un lieu d'expérimentation où s'atténue la limite entre le théâtre et d'autres genres littéraires comme le roman ou la poésie. En publiant certaines de ses pièces avec le sous-titre de « Spectacle dans un fauteuil » Alfred de Musset (et le drame romantique en général) ouvre la voie à une didascalie riche et narrative dont l'objet n'est plus seulement la description empirique d'éléments visuels ou d'indication de jeu, mais aussi la constitution d'un élément diégétique autonome, accessible seulement au lecteur. C'est ainsi qu'au XX^e siècle des dramaturges tels que Beckett ou Ionesco en viennent à poser, dans toutes les didascalies, mais en particulier dans ces didascalies scénographiques, des éléments descriptifs de plus en plus détaillés.

² *Un fil à la patte.* George Feydeau, 1903. Acte 1 Sc 4.

³ *Pelléas et Mélisande.* Maurice Maeterlinck, 1907.

Ces descriptions produisent ainsi une « lecture-spectacle », c'est-à-dire des didascalies qui « occupe[nt] dans la lecture, un temps équivalent à celui de l'action mise en scène. » (Manako, 2006). Autrement dit, la didascalie n'est plus seulement le substitut de l'action scénique, mais une action transposée à la lecture. Cette expérimentation explique l'évolution du théâtre comme une suite de tableaux, présentés de manière très détaillée dans ces didascalies d'exposition scénographique. Prenons par exemple *Rhinocéros* (Ionesco, 1959), sous-titré « Pièce en trois actes et quatre tableaux », qui débute bien avec le mot « Décor » indiquant une didascalie dont la fonction est scénographique, mais qui en réalité mêlent à la fois des indications de décor, de lieu, d'atmosphère et de jeu dans une didascalie liminaire longue de trois paragraphes. La citation suivante est un extrait choisi de cette didascalie.

ACTE PREMIER

Décor

*Une place dans une petite ville de province. [...] Ciel bleu, lumière crue, murs très blancs. C'est un dimanche, pas loin de midi, en été. Jean et Bérenger iront s'asseoir à une table de la terrasse. [...]*⁴

Cette didascalie introductive déroge à la tradition : non seulement elle dépasse le cadre de la fonction informative et le principe de brièveté, mais elle n'est pas entièrement rédigée au présent de l'indicatif, ce qui montre bien le passage vers un mode narratif proche du roman.

À partir de ce bref tour d'horizon de l'histoire de la didascalie dans le théâtre occidental, nous apercevons l'intérêt d'une annotation qui rend compte du contenu des didascalies : identifier le contenu de la didascalie permet l'observation, entre autres, de spécificités génériques ou stylistiques d'une pièce.

1.2. Place de la didascalie dans les HN

1.2.1. Analyse en *distant reading*

Le concept de lecture distante ou « *distant reading* », par opposition au « *close reading* », consiste à produire une analyse littéraire sur la base de données quantifiables selon des mesures applicables à tous les textes (Moretti, 2000). On peut rapprocher la lecture distante d'une forme de « *macroreading* » (Escobar Varela, 2021, p. 82) et parler dans ce cas de « macroanalyse » pour évoquer toute analyse systématique de données à partir d'une méthodologie quantifiable (Jockers, 2013, p. 25).

La pièce de théâtre prise comme objet textuel, est propice à une analyse distante puisque les didascalies au sens large permettent d'identifier des « unités prédéterminées » qui « de par l'objectivité de leur délimitation [...] se prêtent naturellement à une analyse numérique et quantitative. » (Douguet, Marc, 2017, p. 3) dans la mesure notamment où l'annotation de certaines didascalies est relativement facilitée par le marquage éditorial (voir 1.3.1.).

⁴ *Rhinocéros*, Eugène Ionesco, 1959. Acte 1 Sc 1.

Les didascalies sont bien souvent employées pour puiser des données métatextuelles. Les didascalies d'entrée et de sortie de personnages servent notamment à étayer des matrices de composition dramatique aussi appelées matrices de configuration.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B	0	1	0	1	1	0	1	1	1	1	0	1	1	1	1
C	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
D	0	1	0	0	0	0	1	1	0	1	1	0	0	0	1
E	0	1	0	0	0	0	1	1	0	1	1	0	0	0	1
F	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0
G	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Figure 1 : Matrice de configuration d'une pièce à quinze scènes et sept personnages (A-G) (Schmidt et al., 2019)

Les matrices de composition dramatique, dont la Figure 1 est un exemple, sont antérieures aux approches d'analyses computationnelles. Elles ont été mises au point par Solomon Marcus dans *Poetica Matematica* (1970), mais elles ont surtout attiré l'attention des chercheurs grâce à leur mise en pratique simplifiée par l'outil informatique qui permet d'en tirer des analyses statistiques. Il est possible, par exemple, de déduire de ces matrices la densité de configuration, calculée en divisant le nombre de cellules marquant la présence d'un personnage par le nombre total de cellules. De cette manière, on peut établir combien des apparences sur scène potentielles ont effectivement eu lieu (Schmidt et al., 2019). On retrouve cette mesure dans les travaux de Dennerlein et al. (2015) et Douguet (2017). À partir de ces matrices, il est aussi possible d'observer les réseaux de relations entre les personnages, ce qui permet de déterminer, entre autres, les relations entre des personnages types (Glorieux, 2017), l'influence des codes d'un genre littéraire sur la combinaison des personnages coprésents (Fischer et al., 2017) ou encore le transfert des connaissances sur scène (Andresen et al., 2022).

Les didascalies dans une définition large, sont aussi utiles pour aider à la caractérisation des personnages. Elles sont par conséquent extraites dans des études macroanalytiques pour déterminer des traits caractéristiques tels que le genre, l'âge et la classe sociale pour la caractérisation de rôle type (Krautter et al., 2022) ou encore pour une approche sociolinguistique des représentations de genre (Ruiz Fabo, Bernhard, et al., 2024; Ruiz Fabo, Boisnard, et al., 2024).

1.2.2. Identification et classification automatique

Si l'analyse des didascalies a fait l'objet de peu de travaux, l'automatisation de l'annotation de textes dramatiques, automatisation nécessaire à la constitution de corpus de grande échelle et d'analyse de plus grande ampleur, est encore moins commune.

Pour commencer, l'annotation du type de didascalie dépend aussi de l'identification automatique de cet élément par comparaison au reste du texte dramatique. Or, une seule étude à ce jour se concentre sur cette tâche, à savoir le travail de Pagel et al. (2021) portant sur l'identification de différents éléments textuels propres au théâtre.

Cette étude attire notre attention sur des points précis concernant la didascalie et sur la classification opérée par de grands modèles de langue contextuels (LLMs). Les modèles en question sont des Transformers, à savoir plusieurs BERT (Bidirectional Encoder Representations from Transformers) disponibles via Hugging Face⁵ et préentraînés soit sur des données en anglais, soit sur des données en allemand, avec leurs variations respectives sensibles ou non à la casse (*cased* et *uncased*).

Cette étude, portant sur un corpus de langue allemande de près de 1,4 million de phrases, a tout d'abord cherché à obtenir une classification binaire pour les éléments suivants : acte (*act*), scène (*scène*), didascalie (*stage*), émetteur (*speaker*) et réplique (*speech*). Autrement dit, ils ont dans un premier temps cherché à déterminer quel élément structurant d'une pièce de théâtre était le plus difficile à identifier. Pour cette première tâche, Pagel et al. ont affiné le modèle bert-base-uncased (modèle préentraîné sur des données en anglais, et insensible à la casse).

Pour toutes les classes sauf une, celle des didascalies, les modèles obtiennent une macro-moyenne supérieure ou égale à 0,95. La détection des didascalies s'est avérée la moins performante avec une moyenne-macro F1 de 0,88, un score estimé « raisonnablement bon », mais qui confirme la difficulté des modèles à distinguer cet élément.

Une deuxième tâche a consisté en une classification de toutes les entrées selon les cinq classes par différents modèles de type BERT préentraînés cette fois sur des données en langue allemande (bert-base-german-cased et bert-base-german-uncased). Cette classification supervisée a fait apparaître la capacité de ces modèles à distinguer les délimitations de scène et d'acte, malgré le petit nombre d'exemples dans le jeu de test (289) par contraste avec le contenu des répliques et didascalies, mais aussi et surtout la difficulté des modèles à distinguer une réplique d'une didascalie. Ces deux classes sont celles qui ont généré le plus de confusion, notamment dans le sens où une didascalie est identifiée comme réplique.

Au-delà de l'identification, peu de travaux sur la classification du type de didascalie existent à ce jour, à notre connaissance. Seules deux communications scientifiques rapportent une expérimentation proche de la nôtre.

Dennerlein et al. (2016), de l'université de Würzburg, se sont concentrés sur quatre types, « *enter*, *exit*, *aside*, *dead* » et ont entraîné l'algorithme Random Forest sur un corpus de 1 986 didascalies en langue allemande. Les didascalies provenant de quinze pièces des XVII^e et XVIII^e siècles de Textgrid⁶, ont été annotées manuellement et tokenisées avec le *Natural Language Toolkit*⁷ (NLTK) (Bird et al., 2009).

Les performances de l'entraînement par validation croisée ont été généralement encourageantes, notamment pour la classe *aside* (mesure F1 de 0,88). Dans la mesure où cette classe se distingue sur le plan sémantique des trois autres, il ne s'agit peut-être pas de

⁵ <https://huggingface.co/>

⁶ www.textgridrep.de

⁷ <https://www.nltk.org/>

résultats particulièrement probants en termes d'identification absolue, mais davantage d'une bonne identification par rapport aux trois classes restantes. Cela questionne sur les performances d'un algorithme d'apprentissage automatique « classique » pour une classification avec davantage de classes ayant une plus grande similarité linguistique ou sémantique. Cela dit, le volume des données d'entraînement reste bas pour toutes les classes et les performances de classification pour *enter* et *exit* sont satisfaisantes aussi avec une mesure F1 de 0,75 pour les deux classes.

Un seul autre travail de classification automatique du type de didascalie a été effectué à partir d'une typologie plus large, à savoir les neuf valeurs de l'attribut @type proposées par la TEI : Daria Maximova & Frank Fischer (2019) ont produit un classifieur atteignant une moyenne F1 de 0,75 à partir de 6 569 didascalies annotées manuellement sur un corpus en langue russe.

La méthodologie de ce projet s'inspire de celles de Dennerlein et al. (2016) et Pagel et al. (2021) pour la classification et construit sur les limites relevées par Pagel et al. ainsi que sur les propositions de Dennerlein et al. pour l'amélioration de la classification. Mais avant d'en arriver à cette classification, il faut d'abord définir plus précisément notre champ d'études, notamment sur le plan de la théorie littéraire.

1.3. Typologies et définition du champ d'études

La didascalie fait l'objet d'un intérêt croissant dans les études littéraires depuis les années 90, notamment avec l'émergence d'études mettant en valeur les données métatextuelles dans le contexte contemporain des HN. Pour cette raison, nous allons regarder de plus près diverses typologies et leur intérêt relatif à un travail d'annotation manuelle ou automatique d'un corpus de théâtre français, indépendamment de son époque ou de son genre.

1.3.1. Traits formels

Tout d'abord, notons que la didascalie se distingue des unités dialogales selon plusieurs caractéristiques. Ces traits formels permettent de replacer la didascalie dans son cadre, à savoir celui d'un texte lu qui a vocation à être interprété. Si la distinction entre la didascalie et la réplique semble évidente pour le lecteur contemporain, il s'avère qu'une définition plus large de la didascalie inclut des éléments textuels que nous ne sommes pas nécessairement habitués à considérer comme une indication de jeu telles que l'annonce des scènes et des actes.

Nous nous basons sur les caractéristiques relevées par Dompeyre (1992, p. 78). Il n'est pas nécessaire qu'un élément textuel réunisse toutes ces caractéristiques pour que celui-ci soit considéré comme une didascalie.

On trouve d'une part des caractéristiques portant sur des conventions de forme :

1. Caractères typographiques en italique ;
2. Placé entre parenthèses ;
3. En décalage ou indenté par rapport au reste du texte.

D'autre part, des caractéristiques qui relèvent de la langue :

4. Conjugaison : le temps utilisé est le présent de l'indicatif avec utilisation du participe et du gérondif,
5. statut modal : pas exactement celui de l'assertion, la didascalie ne laisse pas place à une thématisation ni à une focalisation (Flaux & Zaragoza, 2002, p. 33) ;
6. Syntaxe : placée en apposition, constituée de fragments de phrase.

1.3.2. Typologie selon le lieu d'apparition de la didascalie (Dompeyre, 1992)

Il faut dans un premier temps établir ce qui ne relève pas de notre champ d'études. Dans la mesure où nous cherchons à extraire les indications scéniques qui rendent compte au plus près de la diégèse, une typologie selon le lieu d'apparition de la didascalie offre une division claire à partir de laquelle nous pouvons déterminer quels éléments textuels entrent et n'entrent pas dans le cadre de ce projet.

Les didascalies « liminaires », correspondant aux titres et aux annotations de genre, ainsi que les didascalies « intermédiaires », servant de découpage structurel en actes, telles que définies dans une typologie selon le lieu d'apparition (Dompeyre, 1992, p. 78), retranscrivent la composition dramatique. Autrement dit, les didascalies liminaires correspondent au découpage interne en acte et en scène. Ces didascalies sont identifiables par leur position en tête de page, voire sur une page dédiée. Un exemple est le seul mot « Acte » ou « Scène » suivi d'un numéro.

Ces didascalies structurelles sont généralement imposées par des conventions qui, en retour, influencent la composition dramatique : à la période classique, un acte ne s'achève qu'à condition que tous les personnages aient quitté la scène et la sortie d'un personnage marque la fin d'une scène.

Pour notre étude, ce sont les didascalies restantes qui sont retenues, à savoir les didascalies présentes en début de scène et celles interstitielles qui apparaissent à l'intérieur et autour des répliques. Ces didascalies sont qualifiées d'« interstitielles » par Dompeyre. Elles ont notamment une valeur qui n'est pas purement structurelle, mais aussi sémantique avec une variabilité plus importante, car elles ne sont pas déterminées par des conventions.

1.3.3. Typologie des didascalies selon leur fonction (Issacharoff, 1981)

Une typologie selon la fonction répond aux enjeux de notre projet, car la fonction permet de cerner le contenu sémantique de la didascalie concernée.

Issacharoff fait une première distinction entre plusieurs types de didascalies selon leur rapport au lecteur. Trois de ses catégories incluent des didascalies qui sont exclusivement destinées à la lecture. Parmi celles-ci, les « didascalies autonomes », définies comme destinées à la lecture et susceptibles de disparaître dans la mise en scène (Issacharoff, 1981, p. 816-8) retiennent notre attention dans la mesure où elles représentent une tendance depuis le XX^e siècle à produire des indications qui ne sont pas totalement « hors texte », c'est-à-dire des commentaires totalement exclus de l'espace diégétique et souvent présent avant ou à

l'intérieur des répliques. Ces indications peuvent être replacées au sein de la fiction dialogale. Certaines font office de commentaire de la part du dramaturge comme dans l'exemple suivant.

LE VIEUX : Bois ton thé, Sémiramis (*Il n'y a pas de thé, évidemment*)⁸

Tandis que d'autres semblent entretenir davantage une valeur narrative propre comme ici :

*Le singe se dispose à obéir. Ninette, qui est restée à l'écart, est tentée par la gaufre, et au moment où le singe grimpe à l'arbre tenant sa gaufre, elle se met en devoir de lui en prendre un morceau. L'animal défend sa proie, les enfants rient, le singe mord le doigt de Ninette.*⁹

Deux autres types désignés par Issacharoff qualifient des didascalies pertinentes à notre étude : les didascalies « verbales » et « visuelles » (Issacharoff, 1981, p. 821).

Les didascalies verbales réunies dans cette typologie correspondent aux fonctions :

1. Nominatives : le nom du personnage qui va énoncer la réplique ;
2. Destinatrices : indication sur le destinataire de la réplique. Exemples : « X à Y » ou « X, seul » ;
3. Mélodiques : brève indication de jeu sur la tonalité de la réplique. Exemples : « *navré, humble, à demi-voix, tendrement ému* »¹⁰
4. Locative : didascalie le plus souvent liminaire visant à décrire le lieu de l'action ainsi que le décor.

Les didascalies visuelles, quant à elles, reposent sur des « codes visuels » relatifs à l'acteur c'est-à-dire des éléments du jeu d'acteur et de son apparence. On compte parmi les didascalies visuelles ou scénographiques, les didascalies :

5. Kinésiques : les indications de jeu qui remplacent la parole ou l'accompagnent, autrement dit les déplacements et les mimiques des comédiens sur scène ;
6. Vestimentaire : les éléments de caractérisation via l'apparence du personnage ;
7. Décor ou scénographie : la description des éléments du décor sur scène.

Cette typologie souligne la difficulté à mettre une étiquette sur un contenu complexe et dont les propriétés sont souvent approchantes. En effet, les didascalies locatives et de décor peuvent être facilement confondues, tout comme les didascalies destinatrices et les didascalies mélodiques.

1.3.4. Typologie selon la définition de l'interaction linguistique (Galleron, 2021)

Au vu des enjeux de ce mémoire, il est important de prendre en considération une typologie plus récente présentée dans le cadre d'une proposition de schéma d'annotation en XML.

La typologie proposée par Galleron repose sur une classification selon la fonction dont l'objectif serait de répondre aux questions fondamentales de l'interaction linguistique à savoir « qui,

⁸ Ionesco, *Les Chaises*. Exemple de (Issacharoff, 1981. p. 818)

⁹ Anonyme, *Les aventures de Ninette*.

¹⁰ Courteline, *Boubouroche*.

quand, où, comment, pourquoi et dans quel contexte » (Galleron, 2021, p. 13). Cette typologie à huit valeurs intervient dans le cadre d'une proposition de schéma d'annotation. Cela signifie que Galleron propose d'ajouter d'autres attributs en plus de l'attribut type, notamment pour rendre compte plus en détail des interactions à l'aide d'attributs @who et @towhom avec des identifiants uniques. La typologie suivante correspond aux valeurs proposées pour son attribut @ana, « analytique » qui dans notre cas correspondrait aux valeurs possibles de l'attribut @type de la TEI :

1. SPC et ASP pour rendre compte de qui interagit.
SPC relatif à la gestion de la parole. Exemple : « à *Mme la Comtesse* », « *Madame* ».
ASP relatif à l'aspect du personnage. Exemple : « *porte une épée* », « *un bouquet de fleurs à la main* » ;
2. TMP et EVT pour rendre compte du « quand ».
TMP relatif au contexte global de la diégèse. Exemple : « *De nos jours* », « *c'est aujourd'hui votre fête* ».
EVT un « événement non-verbal, qui vient interrompre ou accompagner l'interaction. Cet événement peut être une entrée ou une sortie, un mouvement d'un personnage, mais aussi un geste, un son, ou un changement de décor. ».
3. SET et PROX pour rendre compte du lieu.
SET relatif au décor et à la liste des personnages.
PROX les indications de positionnement des personnages.
4. ATT et VOI pour rendre compte du « pourquoi » et du « comment ».
ATT relatif à une « modalisation du dire ».
VOI pour désigner « une particularité sociolinguistique (accent, débit) ».

Cette suggestion d'annotation à l'aide de plusieurs attributs a l'avantage de refléter de manière assez complète les enjeux de l'interaction, mais certains des cas précédemment cités comme les didascalies « autonomes » d'Issacharoff.

1.3.5. Enjeux propres à l'élaboration de la typologie pour ce projet

Les enjeux de ce mémoire peuvent être résumés en deux axes objectifs majeurs. D'une part, il s'agit d'établir une typologie permettant de rendre compte du contenu sémantique des didascalies en tenant compte de plusieurs contraintes. Il est nécessaire de produire une typologie qui puisse être appliquée par des annotateurices humain.e.s et qui puisse faire l'objet d'une généralisation suffisamment reproductible par une machine. Cette typologie a pour objectif de rendre compte des spécificités du texte dramatique, notamment dans la perspective d'études spécifiques aux HN tels que la transmission de savoir, les réseaux de relations, les études stylistiques, etc. D'autre part, il s'agit d'obtenir un modèle capable d'effectuer une prédiction pour de nouvelles données à partir de cette typologie, indépendamment du genre ou de l'époque représentés par ces données.

1.4. Outils et méthodes pour l'analyse computationnelle des didascalies

Dans cette partie, nous abordons les enjeux spécifiques à l'automatisation du type des didascalies sous l'angle pratique. Dans un premier temps et afin d'établir notre propre jeu de données et dessiner notre propre typologie, nous observons les normes et pratiques

existantes autour de l'annotation du type de la didascalie dans les corpus de travail en XML. Dans un second temps, nous abordons les approches méthodologiques relatives à l'apprentissage automatique afin de cerner les possibilités offertes par les algorithmes actuels. Enfin, nous parcourons les outils d'extraction, de prétraitement et d'analyse de données textuelles nécessaires à l'implémentation du projet.

1.4.1. Standardisation et pratique d'annotation en XML

1.4.1.1. Norme d'annotation : guide de la TEI

L'annotation est une technique consistant à enrichir des données textuelles par des données supplémentaires. La pratique de l'annotation précède de loin les Humanités Numériques et réfère avant tout à l'ajout de commentaires. La question de la fiabilité (reproductibilité de la méthode) et de la validité (méthode documentée et justifiée) de l'annotation est centrale dans l'appréciation d'un corpus annoté. Nous verrons que, dans notre cas, nous disposons d'une norme proposée par la TEI, qui repose sur des critères fiables et valides, mais qui soulève un certain nombre de questions face à la réalité des textes à annoter, et d'un corpus annoté manuellement, le FrenchDramaCorpus, dont la validité pour le cas précis de l'attribut @type est questionnable.

La *Text Encoding Initiative*¹¹ est une association composée d'institutions universitaires et de chercheurs qui a pour objectif la normalisation des textes sous forme digitale. Cette standardisation est présentée par des guides publiés depuis 1994. Ces guides sont aujourd'hui disponibles en ligne.

De nombreux projets ont adopté les normes de la TEI pour l'encodage de pièces de théâtre en XML. Par conséquent, la norme quant à l'encodage des indications scéniques en XML, à quelques exceptions près, est d'employer la balise <stage> telle que proposée dans le guide d'annotation de la TEI¹². Le type de didascalie est annoté par un attribut à l'élément « stage », l'attribut @type.

Le guide de la TEI suggère quelques valeurs à l'attribut type appuyées d'exemples et de définitions, ces définitions et exemples ne sont pas toujours disponibles en français. Le Tableau 1 retranscrit les neuf valeurs de @type assorti d'exemples en français ou le cas échéant, en anglais.

Valeur de @type	Description	Exemples
business	décrit une action sur scène	Frappant à la porte (Il se lève.)
costume	décrit le costume d'un personnage	Disguised as Ansaldo.
delivery	décrit la façon dont parle un personnage	à <name>Georgette</name>. , riant.

¹¹ <https://tei-c.org/>

¹² <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-stage.html>

entrance	décrit une entrée	Enter Husband as being thrown off his horse.
exit	décrit une sortie	, s'en allant.
location	décrit un lieu	At a window.
modifier	donne certains détails à propos d'un personnage	Disguised as Ansaldo.
mix	contient plusieurs types d'indications	<stage type="mix">The action takes place in and around the home surgery of Dr Bero, lately returned from the wars.</stage>
[mixed]	contient plusieurs types d'indications	<stage type="entrance modifier">Enter Latrocinio disguised as an empiric</stage>
novelistic	texte explicatif de la mise en scène	Having had enough, and embarrassed for the family.
setting	décrit une mise en scène	La scène est dans une place de ville. Music.

Tableau 1 Valeurs de @type en TEI-P5.

Le datatype TEI de l'attribut @type ne permet qu'une valeur, ce qui explique la valeur « mix ». La TEI propose cependant, un attribut *analytic*, @ana¹³, dont le datatype permet plusieurs valeurs, ce qui permet en théorie d'encoder en détail les différents types de l'élément.

Nous allons comparer ces propositions à la réalité du travail d'annotation grâce au projet Drama Corpus (DraCor).

1.4.1.2. Pratiques de l'encodage du type de didascalie : DraCor

Le projet DraCor est l'étendard du concept de *Programmable Corpus* (Fischer et al., 2019). C'est une interface de programmation (*Application Programming Interface* ou API) capable de produire des données extraites des documents encodés en XML/TEI¹⁴. DraCor inclut à ce jour seize corpus : Alsatian Drama Corpus, Bashkir Drama Corpus, Calderón Drama Corpus, French Drama Corpus, German Drama Corpus, German Shakespeare Drama Corpus, Greek Drama Corpus, Hungarian Drama Corpus, Italian Drama Corpus, Roman Drama Corpus, Russian Drama Corpus, Shakespeare Drama Corpus, Spanish Drama Corpus, Swedish Drama Corpus, Tatar Drama Corpus et Ukrainian Drama Corpus.

Comparons plusieurs pratiques d'annotation de l'attribut @type à partir des corpus ayant le plus gros volume de pièces dans DraCor.

French Drama Corpus, à l'origine « Théâtre Classique »¹⁵, est un corpus de pièces de théâtre en langue française annotées par Paul Fièvre. Ce corpus comprend actuellement 1 700 pièces, dont 1 560 pièces encodées en XML/TEI. Les dates de publication des pièces du corpus s'étendent de 1170 à 2017¹⁶.

¹³ https://tei-c.org/release/doc/tei-p5-doc/en/html/ref-att_global_analytic.html

¹⁴ <https://dracor.org/doc/what-is-dracor>

¹⁵ <http://www.theatre-classique.fr/>

¹⁶ Données extraites via l'API de DraCor.

Les lignes directrices de l'annotation de Paul Fièvre ne sont pas disponibles. Cependant, nous savons que pour intégrer Théâtre Classique à DraCor, il a été nécessaire de supprimer l'attribut @type lorsqu'il n'avait pas de valeur (noté [N/A] dans le corpus Théâtre Classique)¹⁷. Cela signifie que l'attribut type ne se retrouve pas dans tous les éléments *stage*.

Pour déterminer les valeurs de l'attribut type, nous nous basons sur une extraction empirique à l'aide des commandes shell suivantes :

```
# compte stage-Element
egrep '<stage' | wc -l
# compte stage-Element avec type-Attribute
egrep -h '<stage type="[^"]+"' *.xml | wc -l
# nettoyage et compte par valeur de @type
egrep -h '<stage type="[^"]+"' *.xml | sed 's/<stage type="//g' | sed 's/".*//g'
| sed 's/^ */g' | sort | uniq -c | sort -rn
# nettoyage et compte par valeur de @type sans les types 'mixed'
egrep -h '<stage type="[^"]+"' *.xml | sed 's/<stage type="//g' | sed 's/".*//g'
| sed 's/^ */g' | sort | uniq -c | sort -rn
# compte du nombre total de didascalies avec une valeur mixed (jusqu'à 4 types
associées)
egrep -h '<stage type="\w+/\w*/?\w*/?\w*"' *.xml | sed 's/<stage type="//g' |
sed 's/".*//g' | sed 's/^ */g' | wc -l
```

FreDraCor compte 38 306 éléments <stage> dont 37 912 avec un attribut @type. Les dix valeurs les plus fréquentes sont :

- toward (4 207)¹⁸
- title (3 870)
- exit (3 630)
- aparte (1 276)
- loud (1 184)
- decor (812)
- entrance (725)
- watch (684)
- show (605)
- read (548)

Plusieurs remarques sont à faire concernant ces valeurs. Tout d'abord, notons qu'il y a, au total, 5 109 valeurs distinctes, un nombre important par comparaison aux neuf valeurs suggérées par la TEI. Sur ces 5 109 valeurs, 3 935 ne sont utilisées que pour une seule didascalie, ce qui réduit le nombre de valeurs récurrentes à 1 174.

Il y a 3 542 valeurs « mixtes », c'est-à-dire des valeurs qui contiennent plusieurs mots séparés par le signe « / ». Ces didascalies mixtes comptent pour 8 068 didascalies au total. Cependant, seulement trois de ces valeurs mixtes représentent plus de 100 didascalies : « low/toward »

¹⁷ <https://github.com/dracor-org/fredracor#readme>

¹⁸ Fréquence absolue de la valeur pour chaque @type

(204), « exit/location » (103) et « exit/together » (101). Dans ce cas, l'annotation respecte un principe de composition suivant le déroulement de l'action décrite comme dans l'exemple suivant :

```
<stage type="entrance/costume/drop/dress">Louis rentre portant un manteau d'or et une couronne qu'il pose sur le guéridon. Il met ensuite le manteau sur ses épaules et la couronne sur sa tête.</stage>19
```

Chaque combinaison produit une valeur distincte, ce qui pose problème au vu du nombre important de combinaisons possibles. On remarque que les valeurs, *delivery*, *novelistic*, *business*, *modifier* et *mixed*, proposées par la TEI, ne sont pas présentes dans FreDraCor. Enfin, nous relevons des incohérences parmi les nombreuses valeurs de @type : des « doublons » tels que « sing » (274) et « SING » (2) ou « danse » (305) et « dance » (11), ou encore des erreurs de traduction telles que « hugh » (53), « declamate » (43), « astonish » (23).

Par comparaison, German Drama Corpus²⁰ (GerDraCor), qui comprend 646 pièces, publiées pour la première fois entre 1560 et 1962, et Russian Drama Corpus²¹ (RusDraCor), qui comprend 212 pièces quant à elles publiées entre 1747 et 1986, n'ont pas d'attribut pour l'élément <stage> bien que ces corpus en aient respectivement 202 392 et 49 440.

Un autre exemple : le Calderón Drama Corpus²² (CalDraCor) contient 205 pièces, avec 24 058 <stage>. CalDraCor a une typologie à quatre valeurs : « withText » (2 647), « unassigned » (91), « exit » (86) et « entrance » (17). Cette classification avec de telles fréquences est certainement le résultat d'un projet spécifique nécessitant une annotation pour une étude sur la configuration dramatique d'un petit nombre de pièces.

Sur la base de quatre corpus distincts, nous voyons bien que l'annotation du type n'est pas une évidence. Elle peut être totalement éludée (GerDraCor et RusDraCor), partiellement employée à des fins spécifiques (CalDraCor) ou quasi-systématique sans pour autant correspondre à un guide d'annotation (FreDraCor). Dans le cas de FreDraCor en particulier, il faut souligner l'importance du travail que l'ajout de l'attribut type lors de l'annotation manuelle représente et la difficulté de cette tâche est d'autant plus visible que nous observons rapidement des « erreurs » dans cette annotation (nombre trop important de types, erreur de traduction vers l'anglais, etc.). Cependant, c'est grâce à ce travail imparfait que nous pouvons tenter une classification automatique, puisqu'il nous sera possible de convertir une partie des 5 109 types originaux vers notre typologie au lieu de devoir classer manuellement plus de 38 000 didascalies.

L'annotation du contenu sémantique des didascalies n'a pas encore été normalisée à l'échelle d'un corpus de grand volume, et ce notamment parce que cette tâche demeure complexe et que les propositions de la TEI ne semblent pas à même de répondre à tous les cas de figure,

¹⁹ *Roi et Reine, Comédie*. Adolphe Carcassonne, 1889. Annotation de Paul Fièvre, juin 2021

Disponible sur : <https://dracor.org/api/corpus/fre/play/carcassonne-roi-et-reine/tei>

²⁰ <https://github.com/dracor-org/gerdracor>

²¹ <https://github.com/dracor-org/rusdracor>

²² <https://github.com/dracor-org/caldracor>

comme le laisse penser la réalité de l'annotation de FreDraCor. L'élaboration d'une typologie limitée à quelques types pourra ainsi contribuer aux HN tant dans l'optique d'une annotation manuelle que pour l'entraînement d'un modèle d'apprentissage automatique.

1.4.2. Approches pour l'automatisation de l'annotation

1.4.2.1. Système à base de règles

Les systèmes à base de règles aussi appelés « systèmes experts » sont des programmes dont les règles, pour une tâche de classification par exemple, sont entièrement écrites pour anticiper le plus possible de scénarios. Les règles sont définies par un programmeur, sa tâche consistant principalement à anticiper au mieux tous les cas de figure pour permettre au programme de prendre des décisions sans plus d'intervention humaine.

S'il est relativement simple de mettre en place un système à base de règles, la complexité des règles additionnées peut rendre compliquée la maîtrise du système et des données produites. Le système ne s'ajuste aux nouvelles données que par l'ajout ou la modification du code même ou au moins des connaissances explicites encodées dans les règles, étape qui n'est pas toujours aisée et résulte en une planification importante pour limiter les cas de modification sur le code.

Nous pourrions dans le cadre de ce projet concevoir un programme, en Python ou en Java par exemple, pour effectuer une classification selon une série de règles permettant de décrire le contenu d'une didascalie. En effet, il est possible, dans certains cas de déduire des règles en fonction de caractéristiques syntaxiques ou morphologiques de phrase types (voir 2.1.1 ou encore 2.1.3.1). Cette approche nécessite une connaissance approfondie et spécifique à chaque corpus. L'objectif de ce mémoire étant d'offrir la meilleure généralisation possible pour une classification, cette approche n'est pas retenue.

1.4.2.2. Apprentissage automatique

L'apprentissage automatique ou *machine learning* (ML) est aussi nommé analyse prédictive ou apprentissage statistique. Il est défini par FranceTerme comme le « processus par lequel un algorithme évalue et améliore ses performances sans l'intervention d'un programmeur, en répétant son exécution sur des jeux de données jusqu'à obtenir, de manière régulière, des résultats pertinents »²³. Par extension, le ML désigne aussi le domaine de la recherche qui se trouve à l'intersection des sciences statistiques, de l'intelligence artificielle (IA ou encore *artificial intelligence, AI*) et des sciences informatiques, et qui vise à extraire des connaissances depuis des données via ces processus d'automatisation (Müller & Guido, 2016, p. 1).

L'apprentissage automatique est une approche empirique, par opposition à une approche experte, autrement dit, elle consiste à concevoir un système capable de faire de la prédiction

²³ <https://www.culture.fr/franceterme/terme/INFO939?domaine=0&q=machine%20learning> Journal officiel du 09/12/2018.

à partir de données et non de règles qui modélisent explicitement les connaissances humaines. Les données sont représentées sous forme de vecteurs contenant autant de dimensions qu'il y a de caractéristiques. Si les données d'entraînement sont étiquetées (on peut appeler une étiquette un « label ») et catégorisées (une catégorie peut aussi être nommée une « classe ») il s'agira d'une tâche d'apprentissage supervisé, dont l'objectif final est la prédiction de la classe pour des données non étiquetées. Dans le cas d'une tâche d'apprentissage non-supervisé, les données ne sont pas classées et l'objectif de l'entraînement est d'établir les liens entre les données en fonction des caractéristiques relevées. Dans notre cas, nous effectuerons un apprentissage supervisé à partir de données étiquetées.

L'apprentissage se déroule en plusieurs étapes : un jeu de données est fractionné en jeu d'entraînement, de test et de validation. Le découpage est habituellement fait de telle sorte que le jeu d'entraînement représente 75% ou 80% du jeu de données intégral. Les 20% ou 25% restants sont divisés en deux pour le test et la validation. L'algorithme évalue la perte (*loss*), c'est-à-dire l'écart entre la prédiction de la classe et la classe réelle, à partir du jeu de validation et ajuste les poids des vecteurs (c'est-à-dire l'importance de chaque caractéristique pour l'attribution d'une étiquette donnée), afin d'améliorer ses performances. Le jeu de validation est par conséquent utilisé aussi pour l'apprentissage.

Il est possible de procéder à un apprentissage par validation croisée, qui consiste à opérer des vagues successives d'apprentissage sur l'intégralité des données d'entraînement et de validation divisé en un nombre de plis défini : à chaque pli les données qui servent à l'entraînement et à la validation sont différentes, ce qui permet de parcourir l'intégralité des données pour l'apprentissage. L'apprentissage par validation croisée ne permet en revanche pas de produire un modèle et représente uniquement une étape permettant d'obtenir pour chaque algorithme les valeurs moyennes de ses performances sur l'intégralité des données. Cette méthode n'est donc utilisée que pour l'évaluation des modèles.

Des mesures telles que la précision, le rappel et la mesure F1 permettent d'évaluer la qualité de la généralisation produite par le modèle prédictif. Quand les résultats sur le jeu d'entraînement et de validation sont satisfaisants, le modèle entraîné est employé pour la classification du jeu de test qui est resté totalement à part.

Dans les applications de TAL, l'adéquation du modèle dépend du degré d'ajustement au corpus d'entraînement. L'objectif est de trouver un modèle qui est à la fois spécialisé destiné à une tâche précise de classification de données, tout en étant capable de généraliser cette classification sur des données qu'il ne connaît pas. Dans la situation où le modèle est trop proche des données d'entraînement, le surapprentissage (*overfitting*) du modèle entrave à la généralisation, et inversement, dans le cas où le modèle serait en sousapprentissage (*underfitting*), les caractéristiques ne sont pas assez probantes et les résultats risquent d'être trop vagues.

La régression en apprentissage automatique est une technique d'apprentissage supervisé qui vise à modéliser et à prédire des valeurs numériques continues. Contrairement à la classification supervisée, où l'objectif est de prédire des étiquettes discrètes (catégories ou classes), la régression cherche à estimer une relation continue entre les variables d'entrée et de sortie. La classification supervisée est le type de ML le plus indiqué pour ce projet de

mémoire dans la mesure où les types de didascalie constituent un ensemble de classes que l'on peut définir et pour lesquelles nous pouvons établir des caractéristiques distinctes.

Il existe plusieurs algorithmes de ML adaptés à la classification supervisée. Une première catégorie de ML est généralement nommée « algorithmes classiques ». Il comprend les modèles linéaires (Ridge Classifier, SGD Classifier, Logistic Regression) et les modèles d'arbre de décision (Random Forest). Le modèle de régression logistique correspond à chacun des neurones des architectures d'apprentissage supervisé à base de réseaux de neurones tels que les *Recurrent Neural Network* (RNN) et les *Convolutional Neural Network* (CNN).

Une deuxième catégorie de ML s'est donc constituée à partir de ces algorithmes classiques. Ces modèles permettent une représentation plus nuancée du texte, car ils se fondent sur des plongements de mots (*word embeddings*) qui peuvent représenter le sens d'un mot sur la base de sa distribution dans un vaste corpus. Certains de ces modèles comme les réseaux de neurones sont dits « d'apprentissage profond », car ils reposent sur un apprentissage par couches successives de représentation (François Chollet, 2017, p.8). Les récents progrès dans la qualité des représentations produites par ces modèles sont notamment dus au mécanisme d'attention, rapportés dans l'article fondateur « Attention is All You Need » (Vaswani et al., 2017) et mis en application dans les architectures de type Transformers tel que le Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) (voir 2.4.4).

Les grands modèles de langues contextuels ou LLMs constituent des représentations établies par l'encodage d'un très grand volume de données par un réseau d'apprentissage profond. C'est le volume des données de préentraînements qui déterminent si un modèle est de grande taille ou non. François Yvon place cette limite à un milliard de tokens ou plus (2023). De par ce grand volume d'informations encodées, les LLMs sont capables de s'adapter à de nombreuses tâches qui nécessitent une forme d'inférence entre des tokens d'entrée et les tokens de sortie (les prédictions) (Yvon, 2023). Leur capacité d'apprentissage par transfert d'information permet d'affiner leurs représentations en fonction d'une tâche ou d'un ensemble de données précises.

Il existe trois grands types de LLMs : des encodeurs purs, comme BERT, conçus pour des tâches de prédiction, des décodeurs purs comme Mistral-7B²⁴ (Albert Jiang et al., 2023) ou les *Generative Pre-trained Transformer* (GPT) dont GPT-3 (Brown et al., 2020), conçus pour la génération et enfin les modèles encodeur-décodeurs hybrides, comme les T5 (Raffel et al., 2019) conçus pour l'analyse de données d'entrées complexes avec une tâche de génération en sortie.

Pour notre projet, nous utiliserons et comparerons tant des modèles de ML classiques que des modèles d'apprentissage profond.

²⁴ https://huggingface.co/docs/transformers/main/model_doc/mistral

1.4.3. Outils pour la mise en place du projet

Dans cette section, nous allons voir quels sont les outils nécessaires à ce projet, de la préparation des données à la classification par divers modèles en passant par l'extraction des données et à l'entraînement des modèles de ML.

1.4.3.1. Extraction

Git

Git est un logiciel de gestion de versions décentralisé. La gestion de version (*version control*) consiste à garder trace de toutes les modifications opérées sur le code d'un projet développé simultanément par une équipe. Ce type de logiciel est communément appelé « *Version Control Systems* » (VCS) ou « *Source Code Management* » (SCM). Git est un logiciel open source développé en 2005 par Linus Torvalds, créateur de Linux OS.

Git est utilisé à plusieurs étapes de ce projet :

- Extraction du code source du dépôt de FreDraCor²⁵ pour obtenir l'intégralité des pièces en XML ;
- Création d'un dépôt pour le code et les données produites tout au long du projet²⁶.

Lignes de commande Linux

GitBash est un logiciel permettant d'utiliser les lignes de commandes Linux indépendamment de l'OS, notamment Windows qui fonctionne avec des lignes de commandes différentes de MacOS et Linux.

Il est possible d'extraire des parties d'un corpus annoté directement avec des commandes shell via GitBash. Dans l'exemple suivant la commande sélectionne le contenu de la balise `<stage type="show">`. Les données extraites sont nettoyées avec des commandes de substitutions « `sed` » avant d'être ajoutées au document « `stagedir_show.txt` ».

```
egrep -h '<stage type=".*show.*"' *.xml | sed 's/<stage type=".*">//g' | sed 's/^ *//g' | sed 's/<\/stage>//g' > stagedir_show.txt
```

Les commandes shell sont surtout utilisées pour établir le compte unique des didascalies et des types.

²⁵ Disponible sur <https://github.com/dracor-org/fredracor>

²⁶ https://git.unistra.fr/alexia.schneider4/stgdir_clf

API de DraCor

Fischer et al. (2019) ont créé une bibliothèque Python afin de requêter l'API de DraCor²⁷. Il est possible d'extraire à partir de l'instance « fre » de la classe Corpus l'intégralité de FreDraCor, et, à partir de là, faire une requête par pièce comme suit :

```
from pydracor import *

corpus = Corpus('fre')
print(corpus.summary())
for id in corpus.play_ids():
    play = Play(play_id=id)
    stgdirs = play.stage_directions()
    print(stgdirs)
```

En revanche, il n'est pas possible d'extraire les attributs d'un élément via cette API, ce qui ne permet pas de faire l'extraction nécessaire à la conception de notre jeu de données.

BaseX²⁸

BaseX est un outil open source développé par Christian Grün et son équipe depuis 2005. La version 10.4 est utilisée pour ce projet. Le logiciel sert à la fois de parseur XQuery et de gestionnaire de base de données XML. Il peut à la fois gérer de larges corpus de documents XML, HTML, JSON et CSV et garder en mémoire les requêtes²⁹. Il est par conséquent possible d'extraire les didascalies de son choix à partir de BaseX comme dans l'exemple suivant pour les didascalies de type « exit » (exclusivement) :

```
declare namespace tei="http://www.tei-c.org/ns/1.0";
let $corpus:=db:get('fredracor')
return $corpus//tei:stage[@type="exit"]
```

1.4.3.2. Prétraitement

NLTK³⁰

NLTK est un projet open-source débuté en 2001 par Steven Bird et Edward Loper. La bibliothèque Python correspondante est une bibliothèque spécialisée dans le TAL avec des approches classiques précédant les approches du TAL basées sur des plongements de mots et sur de grands modèles préentraînés. Elle contient des outils de tokenisation, d'annotation morphosyntaxique, d'analyse en dépendance syntaxique. NLTK propose en plus de cette

²⁷ <https://pypi.org/project/pydracor/>

²⁸ <https://basex.org/>

²⁹ https://docs.basex.org/wiki/Main_Page

³⁰ <http://www.nltk.org/>

collection d'outils Python pour le TAL une interface pour une cinquantaine de corpus et ressources lexicales dont des antidiCTIONNAIRES (*stopwords*).

Elle a été utilisée notamment par Dennerlein (voir 1.2.2) dans l'étape de classification des didascalies d'entrée et sortie relevées manuellement, pour la tokenisation et l'analyse morphosyntaxique.

spaCy³¹

SpaCy est une bibliothèque Python open-source spécialisée dans le TAL avec une approche basée sur les plongements lexicaux, c'est-à-dire qu'il s'agit d'une bibliothèque conçue sur des approches récentes de TAL. SpaCy est créé en 2015 par Matthew Honnibal et Ines Montani.

La bibliothèque Python de spaCy permet l'extraction de données relatives à une chaîne de texte à partir d'un pipeline de prétraitement. Selon les modèles proposés, le pipeline permet d'effectuer la tokenisation, l'analyse morphosyntaxique ou en dépendance, la lemmatisation ou encore la reconnaissance d'entité nommées. SpaCy propose notamment un modèle de prétraitement de langue française, `fr_core_news`, construit à partir de l'annotation de Universal Dependencies³² du corpus Sequoia³³.

Plus récemment, spaCy a intégré la bibliothèque `spacy-llm`³⁴ à son offre. Ce dernier module permet d'utiliser un LLM de son choix pour le traitement des données, mais aussi de l'entraîner à nouveau sur des données au format `spacy`.

1.4.3.3. Exploration et analyse

Pandas³⁵

Pandas (The pandas development team, 2024) est une bibliothèque Python créée en 2008 par Wes McKinney. Elle est largement utilisée pour la manipulation et l'analyse de données. Elle offre des structures de données puissantes et flexibles, notamment les DataFrames, qui permettent de traiter facilement des données tabulaires.

La version utilisée pour le projet est 2.0.3.

³¹ <https://spacy.io/>

³² <https://universaldependencies.org/>

³³ https://github.com/UniversalDependencies/UD_French-Sequoia

³⁴ <https://spacy.io/usage/large-language-models>

³⁵ <https://pandas.pydata.org/>

Matplotlib³⁶ et Seaborn³⁷

Matplotlib est une bibliothèque de visualisation de données lancée en 2003 et seaborn une bibliothèque de visualisation de données construite sur matplotlib en 2012. En intégrant et en simplifiant certaines fonctionnalités de matplotlib, seaborn permet aux utilisateurs de produire facilement des visualisations de données visuellement attrayantes. Ces deux bibliothèques sont open-source.

Les versions utilisées pour le projet sont matplotlib==3.7.1 et seaborn==0.13.1.

1.4.3.4. Apprentissage automatique

Scikit-learn³⁸

Débuté en 2007 par David Cournapeau, scikit-learn est un projet open-source dédié à l'apprentissage automatique en Python. La bibliothèque scikit-learn (Pedregosa et al., 2011) est largement utilisée car elle offre une large sélection d'algorithmes ainsi que des fonctionnalités de pré-traitement des données et d'évaluation des modèles. Scikit-learn est compatible avec d'autres bibliothèques telles que pandas, pour la manipulation des données, et matplotlib, pour la visualisation.

La version utilisée pour le projet est 1.2.2.

Hugging Face³⁹

Hugging Face (HF) est une plateforme collaborative internationale pour le développement d'outils et de ressources spécialisées dans l'intelligence artificielle. Pour ce qui relève du TAL, on y trouve aussi bien des modèles comme des *LLM* préentraînés que des jeux de données, le tout étant libre et disponible au public.

Les modèles phares pour les tâches de classification supervisée sont aujourd'hui les Transformers comme BERT. Pour les modèles préentraînés sur des données en français on trouve notamment camemBERT^{40,41}. HF propose un accès simplifié à ces modèles par le biais de la bibliothèque transformers⁴² qui contient des pipelines simplifiant le téléchargement de modèles recensés sur HF. Il est aussi possible d'y déposer ses propres modèles.

³⁶ <https://matplotlib.org/>

³⁷ <https://seaborn.pydata.org/>

³⁸ <https://scikit-learn.org/stable/index.html>

³⁹ <https://huggingface.co/>

⁴⁰ <https://huggingface.co/camembert-base>

⁴¹ <https://camembert-model.fr/>

⁴² <https://pypi.org/project/transformers/>

OpenAI⁴³

Fondée en 2015, OpenAI est une entreprise américaine réputée pour ses contributions importantes dans le traitement du langage naturel grâce à ses modèles de génération de texte, notamment avec ses modèles de langue GPT, dont la version conversationnelle ChatGPT a fait découvrir au grand public la génération textuelle avec une compréhension contextuelle de requêtes en langue naturelle. Ces modèles de langue ne sont pas open-source.

L'API est notamment disponible via la librairie Python. La version de cette librairie utilisée pour le projet est 1.12.0.

Mistral AI⁴⁴

Mistral AI est une entreprise relativement récente dans le domaine de l'intelligence artificielle, débutée en 2023 et co-fondée par Arthur Mensch, Guillaume Lample et Timothée Lacroix. L'entreprise propose des modèles de langue open-source en concurrence avec ceux d'OpenAI.

L'API est aussi accessible via une librairie Python. La version de la librairie `mistralai` utilisée au cours du projet est 0.0.11.

⁴³ <https://openai.com/>

⁴⁴ <https://mistral.ai/>

Partie 2 : Méthodologie

Cette partie rapporte les choix méthodologiques du projet. Cela comprend, dans un premier temps les deux typologies proposées. La première est aussi la typologie principale qui reflète le guide d'annotation du type de didascalie (2.1). La deuxième typologie constitue la *baseline* pour les expérimentations et répond d'une méthodologie simplifiée (2.2). Une troisième sous-partie est consacrée à l'établissement des jeux de données correspondant à ces typologies (2.3). Et une dernière sous-partie se concentre sur la présentation et l'entraînement des différents classifieurs automatiques (2.4).

2.1. Guide d'annotation de la typologie générique proposée

Cette section constitue un guide d'annotation correspondant à la typologie générique proposée pour ce mémoire. Cette typologie est présentée en regard de celles évoquées en 1.3 ainsi que par rapport aux normes et réalités de l'annotation en XML présentées en 1.4.1.

L'objectif de cette typologie est double : d'une part, les types proposés doivent correspondre à la recherche actuelle dans les HN, et d'autre part, ils doivent être suffisamment généraux pour rendre compte d'éléments dont le contenu est très hétérogène. Cette taxonomie contient treize types, nommés en anglais afin de se rapprocher de la *TEI*, et présentés ici par groupes sémantiques (Figure 2) puis par ordre alphabétique avec plusieurs exemples de didascalies correspondants (Tableau 2).

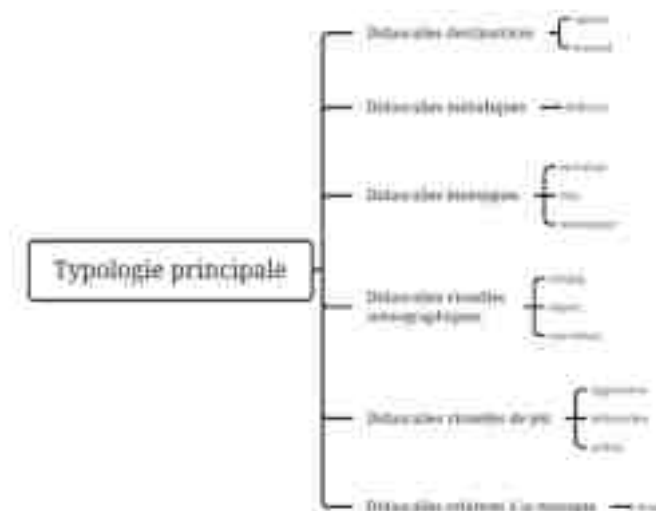


Figure 2 Typologie principale avec 6 groupes et 13 étiquettes

@type	Exemples
action	Il désigne le garçon de café. Il lit. Elle s'assied.
aggression	Il tire son épée. Il se donne un coup.
aparte	À part.
delivery	En riant. À demi-voix.
entrance	Ils entrent en scène. Il rentre chez lui.
exit	Il sort. Il rentre.
interaction	Elle va aussi pour l'embrasser. Il lui prend la main.
movement	Il continue sa marche. Il recule d'un autre côté. Il veut sortir.
music	Air en duo. Musique céleste.
narration	À peine il est sorti du Temple, qu'une Furie s'éveille ; ne le voyant plus elle réveille ses compagnes. Les trois Déesses expriment par une pantomime très animée le chagrin qu'elles ressentent de l'avoir perdu. Elles cherchent partout dans le Temple, et sortent enfin en suivant les traces du sang qu'Oreste laisse après lui.
object	Il lui donne un écu. Elle froisse la lettre.
setting	Le théâtre représente un salon. En Sicile.
toward	À Julie. Au commandeur et au comte.

Tableau 2 Exemples pour chacune des 13 classes de la typologie principale

Dans les sections suivantes, les types sont présentés l'un après l'autre par groupement thématique et sémantique selon la présentation de la Figure 2.

2.1.1. Didascalies destinatrices

Les didascalies destinatrices comme défini par Issacharoff (1981, p.821) correspondent aux énoncés qui suivent la structure « X, seul » ou « X à Y ». Ces structures types correspondent respectivement aux types *aparté* et *toward*.

Ce groupe est scindé en deux groupes distincts, car certaines études sur les réseaux de relations reposent sur la capacité à distinguer une interaction, qu'elle soit parlée ou non, d'une action opérée par un personnage seul. Distinguer les types « *aparté* » et « *toward* » permet ainsi d'établir si une information est transmise au public seulement ou aux personnages co-présents sur scène.

2.1.1.1. *aparté*

Ce type fait partie des didascalies « verbales » d'Issacharoff. Elle correspond à une partie des didascalies « destinatrices » qui indiquent à qui s'adresse un personnage. L'*aparté* est une spécificité de la représentation théâtrale, mais demeure à ce jour un marqueur sous-exploré dans les études computationnelles de théâtre, notamment parce qu'il n'est pas aisé d'établir qu'une réplique est prononcée en *aparté* quand plusieurs personnages se trouvent sur scène.

Un *aparté* est particulièrement facile à identifier sur le plan lexical et syntaxique puisqu'il est presque toujours exclusivement composé de la phrase type « à part » avec possiblement des variations telles que « Toujours à part » ou « À part en s'en allant ». Pour cette raison, il semble de prime abord un type propice à produire des bons résultats lors d'une classification automatique. Cependant, cette similarité risque de réduire drastiquement l'effectif de cette classe lors de la suppression des doublons.

Le guide de la *TEI* ne propose pas une annotation de ce type, mais c'est une des quatre classes employées par Dennerlein et al., ce qui incite à son inclusion dans une typologie plus complète. De plus, il s'agit d'un type très bien pourvu dans *FreDraCor* grâce aux deux types suivants en correspondance : « *alone* » (72 occurrences), et « *aparte* » (1 276 occurrences)⁴⁵. Cela en fait un type non seulement intéressant pour les HN, mais aussi un type identifiable au moment de l'entraînement.

2.1.1.2. *toward*

Ce type fait partie des didascalies « verbales » d'Issacharoff. Il correspond aux didascalies « destinatrices » qui indiquent à qui s'adresse un personnage. Dans la *TEI* cette valeur existe aussi et ce type se retrouve à lui seul 4 207 fois dans *FreDraCor*, ce qui en fait la valeur la plus fréquente du corpus.

Cette didascalie souvent très brève est facilement identifiable parce qu'elle est le plus souvent composée exclusivement de la préposition « à/au » suivie du nom d'un personnage.

Les phrases les plus fréquentes pour ce type de didascalies suivent ces motifs :

- (1) *À l'Empereur.*
- (2) *Il dit à Isabelle.*

⁴⁵ Notons que la phrase « à part », qui constitue souvent l'intégralité de ce type de didascalie, a été annotées 21 fois par « *toward* ». Extraction en XQuery pour le détail :
`$corpus//tei:stage[@type="toward"]/text()[matches(., "[àaÀ] part.?")]`

(3) *Elle lui dit.*⁴⁶

Autrement dit, ce type est relativement facilement identifiable sur la base de caractéristiques lexicales et syntaxiques, ce qui en fait un candidat propice à l'intégration dans une typologie généraliste pour l'annotation automatique.

2.1.2. Didascalies mélodiques : *delivery*

Didascalies verbales selon Issacharoff, ou didascalies « mélodiques » qui répondent à la question « comment » au sens de « comment est interprétée une réplique ». Elles correspondent dans la typologie de Galleron (2021, p.16) au type ATT qui « précise une intonation ou autre modalisation du dire [il] indique une mimique ou une attitude qui donnent une information à propos des sentiments, des motivations, des intentions du personnage ».

Sur le plan morphosyntaxique, elles contiennent principalement des adjectifs, des adverbes ou expressions en collocation forte. Bien qu'elles soient sur le plan sémantique relativement facilement identifiables, elles ne constituent pas une valeur à part entière dans l'annotation de FreDraCor : on compte parmi ce type l'annotation « loud » dans 1 184 cas, et qui correspond de manière quasi exclusive au mot-phrase « Haut », ou encore « low » avec 314 didascalies.

2.1.3. Didascalies kinésiques

Cette catégorie inclut trois types, *entrance*, *exit* et *movement*, qui relèvent toutes des déplacements des personnages sur scène. Le terme kinésique se retrouve dans les typologies de Dompeyre et d'Issacharoff.

2.1.3.1. *entrance* et *exit*

L'intérêt et la portée de ces deux types sont très proches dans les études littéraires computationnelles (voir 1.2.1). Pour cette raison, elles sont présentées ensemble tout en conservant une valeur distincte pour ces deux types. Nous sommes encouragés à conserver ces deux valeurs aussi bien explicitement par le guide de la TEI qu'implicitement du fait de l'uniformité de leur annotation effective dans FreDraCor. En effet, *exit* et *entrance* ont respectivement 3 632 et 725 occurrences dans FreDraCor, ce qui fait de ces valeurs parmi les mieux pourvues du corpus.

Les didascalies d'entrée et de sortie sont identifiables sur le plan syntaxique, mais ambiguës en français sur le plan lexical. Prenons quelques exemples.

(1) *Entre la reine*

La structure syntaxique la plus attendue pour la didascalie d'entrée place le verbe avant le sujet (1).

⁴⁶ *La Belle Alphrède*, comédie. Jean de Rotrou.

- (2) *Elle sort.*
- (3) *Sort le serviteur* ⁴⁷
- (4) *Exit Clytemnestre* ⁴⁸

La didascalie de sortie est au contraire construite sur un schéma syntaxique SN + V (2) sauf dans le cas d'une inversion nominale absolue (3), qui se rapproche de la forme latine « *exit* » correspondante (4).

Pour les didascalies d'entrée, le verbe, majoritairement « entrer », est en position initiale absolue et son sujet, un syntagme nominal reprenant le nom ou le titre du personnage vient après (Flaux & Zaragoza, 2002). La didascalie de sortie suit une réplique énoncée par un personnage ou un groupe de personnages, le lecteur peut ainsi reconstruire la référence pronominale par ce seul enchaînement logique contrairement à la didascalie d'entrée.

C'est aussi pour cette raison que « Elle entre » / « Elle rentre » est possible, mais marque dans ce cas, souvent, l'entrée d'un personnage dans la coulisse, autrement dit, sa sortie. Cette ambiguïté entre l'entrée sur scène et l'entrée dans la coulisse pose problème dans le cas où les didascalies sont extraites sans les répliques ou les indications de limites de scène, informations contextuelles nécessaires à la bonne classification tant pour un.e annotatrice humaine que pour un classifieur automatique. Parce que notre classification porte uniquement sur des didascalies décontextualisées, le contronyme « rentrer » est susceptible d'engendrer des confusions.

2.1.3.2. *movement*

Ce type a pour objet les déplacements des personnages sur scène. Il entre dans le champ des didascalies visuelles décrites par Issacharoff et plus précisément des didascalies kinésiques.

Ce type est hérité de l'annotation proposée par la TEI. Cependant, le guide de la TEI donne deux stratégies d'encodage pour les déplacements. La première possibilité est celle de l'attribut @type avec la valeur « move » dans l'élément <stage> comme dans l'exemple suivant :

```
<stage type="move">Il passe devant la table.</stage>
```

La deuxième est celle de la balise autofermante <move/>⁴⁹ dans le but notamment de relever le détail du mouvement grâce aux attributs suivants :

- @type pour indiquer la direction du mouvement avec trois types notables : *entrance*, *exit* et *onStage* ;
- @where décrit le lieu précis sur le plateau ;
- @who pour l'identifiant du personnage ;
- @perf identifie la ou les représentations au cours desquelles s'est effectué le déplacement décrit.

⁴⁷ *Macbeth*, Shakespeare. Trad. Leyris 1997.

⁴⁸ *Electre*, Giraudoux (exemples cités par Flaux & Zaragoza. (2002, p. 33)).

⁴⁹ <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-move.html>

La balise <move/> peut être imbriquée dans la balise <stage>, les deux stratégies d'annotation ne sont donc pas mutuellement exclusives comme dans l'exemple suivant :

```
<stage type="entrance">  
  <move who="#fr_clar" type="enter" perf="#perf1"/>(Entre Clarence, entouré  
de gardes).  
</stage>
```

Seule la première stratégie est employée dans ce projet, celui qui n'utilise que l'attribut @type, mais relever la possibilité d'un autre encodage met au jour l'importance de ces déplacements.

Nous suggérons de distinguer une entrée ou une sortie d'un autre déplacement de la part d'un personnage ou un groupe de personnages afin de dégager plus précisément une matrice de configuration, c'est-à-dire la matrice de co-présence sur scène, tout en conservant la dimension visuelle et dynamique de l'indication de jeu.

2.1.4. Didascalies visuelles scénographiques

Les trois types de la catégorie des indications scénographiques font tous partie des didascalies visuelles d'Issacharoff et plus précisément des didascalies « locatives » ou « scénographiques ».

La recherche actuelle s'interroge sur leur position en tant qu'objet narratif intermédiaire. Ce sont souvent dans ces didascalies que prennent le plus de liberté les dramaturges puisqu'elles posent un cadre qui peut être concret ou métaphorique, scénographique ou narratif. Nous tentons par cette typologie de conserver leur ambiguïté sémantique tout en cherchant à dessiner des limites claires pour une annotation humaine et une classification automatique.

2.1.4.1. setting

Setting est un type qui correspond aux indications factuelles relatives à l'espace et au temps. Bien que ce type ne soit pas suggéré dans la TEI sous ce nom, le type « *location* » s'y apparente en partie. Les exemples suivants présentent les patrons les plus communs pour ce type :

(1) En Sicile.⁵⁰

(2) La scène représente une chambre d'auberge de campagne.⁵¹

On trouve soit la forme brève annonçant le lieu (1). Un motif similaire reflète l'époque avec la préposition « à ». Soit on trouve la tournure la plus attendue « la scène représente » (2).

⁵⁰ *Le Jeu d'Esmoré*, anonyme.

⁵¹ *L'Enragé*, Louis Carrogis de Carmontelle

Nous présentons avec *setting* une variante qui concerne surtout les indications liminaires qui introduisent à la fois le lieu où se passe la pièce et le décor. Pour cette raison, elle regroupe les valeurs « location » (207) et « decor » (812 occurrences) de FreDraCor.

2.1.4.2. object

Les didascalies d'objets au sens strict sont des didascalies matérielles qui réfèrent à des objets saisissables et non employés dans un sens figuré et symbolique, mais la distinction est rarement claire au théâtre où l'occupation de la scène est à la fois matérielle et métaphorique.

Ce type correspond à une suggestion de Dompeyre qui fait remarquer l'importance des objets sur scène et de la didascalie qui fait état de la scénographie : « La didascalie ne se borne, donc, pas à des indications de régie, elle n'est pas la simple liste d'accessoires à rassembler (pas plus que celle des gestes à effectuer). Sa valeur dépasse l'instrumental. » (1992, p.87). L'accessoire est un élément de jeu, il occupe bien souvent un rôle tant dans le déroulement de l'action que dans la caractérisation des personnages. Ce type permet potentiellement d'extraire les informations relatives à cette dimension symbolique de la représentation que ce soit dans la description d'un objet/accessoire ou dans la manipulation sur scène de cet objet.

- (1) COSTUME DU HURON. Redingote de couleur claire, très courte ; pantalon de couleur foncée, également très court ; gilet rayé, chemise bleue à raies cravate à la Colin, bas blancs, souliers à cordons, perruque de nègre, très frisée, longs pendants d'oreilles en corail⁵².
- (2) Elle jette le livre⁵³.

Ce type comprend des indications relatives au costume (1) comme à l'emploi d'un objet dans certaines actions (2). Ainsi il n'y a pas de type spécifique pour cette catégorie d'indication dans FreDraCor : on trouve aussi bien « costume » (58 occurrences), ou « get » (340 occurrences), que « give » (327 occurrences). Pour cette raison, c'est *a priori* un des types les plus sémantiquement ambigus, notamment parce qu'il implique un large panel d'actions, ce qui peut prêter à confusion avec le type *action*.

2.1.4.3. narration

Ce type n'est pas aussi facilement identifiable que les précédents de la même catégorie, car il correspond aux didascalies avec plus d'ambiguïté. Ce type existe sous le nom de « *novelistic* » dans la TEI et on le retrouve tel quel dans les valeurs de FreDraCor pour 383 didascalies.

- (1) Voyant qu'elles s'expliquaient à moi sans aucun scrupule, je leur dis qu'encre qu'elles m'eussent commandé de les écouter sans interrompre leurs discours, que je ne pouvais m'empêcher de leur confesser qu'elles avaient assez bien défendu leur opinion, touchant l'amour qu'elles pouvaient avoir pour un seul objet ; mais qu'il fallait

⁵² *Le Huron ou les Trois Merlettes*, Xavier Duvert, 1834.

⁵³ *La Fille Mal Gardée ou le Pédant Amoureux*, Charles Favart, 1758.

traiter de cet amour qu'on feint d'avoir pour plusieurs, toutefois en ressentir que pour soi-même⁵⁴.

- (2) Le singe se dispose à obéir. Ninette, qui est restée à l'écart, est tentée par la gaufre, et au moment où le singe grimpe à l'arbre tenant sa gaufre, elle se met en devoir de lui en prendre un morceau. L'animal défend sa proie, les enfants rient, le singe mord le doigt de Ninette⁵⁵.

On inclut systématiquement dans ce type les didascalies écrites dans un temps du passé (1) et les didascalies plus complexes qui s'approchent du récit en prose et s'éloignent du récit dialogal (2).

2.1.5. Didascalies visuelles de jeu

Cette catégorie comprend trois types, *action*, *interaction* et *agression*, qui relèvent tous des indications visuelles et plus particulièrement des gestes et interactions entre comédiens. Elles sont aussi nommées didascalies d'action pour cette raison. Cet ensemble de didascalie est divisé en trois afin d'obtenir une annotation plus fine qui permette d'évaluer la capacité des modèles à généraliser sur des critères sémantiques complexes. L'intérêt notamment de la distinction entre action et agression/interaction est, comme mentionné pour les didascalies destinatrices, de pouvoir distinguer une action opérée par un personnage seul d'une action opérée qui implique plusieurs personnages. Galleron suggère un type EVT pour tous les événements non verbaux qui correspond à ces trois étiquettes.

2.1.5.1. *agression*

Le type *agression*, qui n'est pas mentionné dans une typologie pas plus qu'il n'est présent dans le corpus FreDraCor, se concentre sur les interactions qui touchent aux conflits entre les personnages. Une annotation de ce type pourrait aider dans l'analyse des réseaux de relations des personnages. Exemples :

- (1) Ils se jettent l'un sur l'autre et se battent⁵⁶.
(2) Elle se tue⁵⁷.

Nous suggérons pour ce type d'inclure aussi bien les indications ayant trait à la violence entre deux ou plusieurs personnages (1) qu'à la violence commise par un personnage sur lui-même (2). Par exemple, ce type réunit les types FreDraCor « threat » et « kill ».

2.1.5.2. *interaction*

Ce type se dessine surtout par comparaison au précédent : il s'agit des indications qui concernent les interactions physiques non violentes entre deux ou plusieurs personnages sur scène. Ce type n'est pas non plus proposé en tant que tel dans les typologies exposées, ni dans FreDraCor, cependant il est pertinent pour l'étude précise des relations sur scène.

⁵⁴ *Dialogue de la Prude et de la Coquette*, anonyme, 1659.

⁵⁵ *Les Aventures de Ninette*, anonyme, 1833.

⁵⁶ *Le Tableau du Mariage*, Louis Fuzelier et Alain-René Lesage, 1712.

⁵⁷ *Le Siège de Paris*, Nicolas de Bohaire-Dutheil, 1780.

- (1) Il embrasse sa mère⁵⁸.
- (2) En les touchant de sa marotte⁵⁹.

Si l'identification de ce type peut être assez simple dans certains cas, où la construction syntaxique permet de retrouver le complément d'objet direct et d'identifier un nom d'humain, certains, comme en (2) ne laissent pas voir si le COD est un être humain ou un objet inanimé. De plus dans ce deuxième exemple, l'objet est assez central dans l'indication, et sa classification dans la classe interaction n'est pas certaine. Cette catégorie est sujette à des confusions aussi bien avec la catégorie action qu'objet.

2.1.5.3. action

Catégorie plus générale que les deux précédentes, le type action reflète un ensemble vaste d'indications non verbales plus indéfini que la plupart des catégories proposées. Elle entre en correspondance avec le type « business » de la TEI.

L'action est une notion vaste et il est tentant pour l'annotatrice de spécifier non pas l'appartenance seule de la didascalie au type des indications de jeu (ou d'action), mais le type d'action précise indiquée par la didascalie. C'est la raison pour laquelle le type « action » n'est employé que 48 fois dans FreDraCor, tandis que des actions particulières sont employées comme type dans de bien plus grandes proportions, comme « watch » (684), « show » (605) ou « read » (548).

Cela signifie que les contours de cette classe doivent être dessinés à partir d'une série de verbes d'action plus précis et que cette classe est bien plus ambiguë que les précédentes.

2.1.6. Didascalies relatives à la musique : *music*

Ce type n'a pas été distingué en tant que tel ni dans la TEI ni dans aucune des typologies explorées plus tôt dans ce mémoire. Partons de quelques exemples pour délimiter ce type :

- (1) On entend quelques accords de luth⁶⁰.
- (2) Coup de cymbales⁶¹.
- (3) Air : Je suis Lindor, du Barbier de Séville⁶².

On trouve des didascalies qui réfèrent à une ambiance musicale (1) ou à une instruction liée à une partition musicale (2)) et des indications paratextuelles sur la musique jouée (3), en l'occurrence le titre d'un air. Ce type fait l'objet d'une attention particulière dans divers projets qui visent à recenser des airs populaires. Par exemple, concernant le théâtre du XVIII^e on

⁵⁸ *Tyl l'Espiegle*, Clovis Hugues, 1906.

⁵⁹ *Les Acteurs Déplacés l'Amant Comedien*, Thomas L'Affichard, 1737.

⁶⁰ *Le Charme de la Voix*, Thomas Corneille.

⁶¹ *Panthéon – Courcelles*, Georges Courteline.

⁶² *L'Abbé de Plâtre*, Louis de Carrogis de Carmontelle.

trouve la base de données Theaville⁶³ portée par le Centre d'études des théâtres de la Foire et de la Comédie-Italienne de l'Université de Nantes, et pour le XIX^e siècle le projet Thealtres⁶⁴ a permis l'exploration de ce type pour le théâtre français (Schneider & Nugues, 2023).

Au-delà de la pertinence de ce type, sa présence importante dans FreDraCor (3 870 occurrences pour *title* et 291 occurrences pour *music*) et son identification simplifiée par un champ lexical marqué encourageant son inclusion dans la typologie.

2.2. Typologie baseline

Afin de comparer la pertinence de notre typologie et les performances des modèles d'apprentissage automatique, il convient d'établir une *baseline* à partir de laquelle nous pouvons comparer à la fois la typologie générique proposée et les modèles.

En l'occurrence, la baseline est conçue à partir des éléments disponibles du corpus original, l'idée étant de former une typologie ne nécessitant pas de prétraitement depuis un corpus déjà annoté, tel que FreDraCor. Dans l'objectif de limiter au maximum l'annotation manuelle, le critère choisi pour dessiner ce jeu de données est celui de la fréquence des types. Il a été envisagé de reproduire l'intégralité de l'annotation originale de FreDraCor, néanmoins, le nombre important de valeurs uniques (5 109) oblige d'opter pour une autre stratégie.

Ainsi, le choix a été fait de conserver tels quels tous les types non-mixtes ayant au moins 100 occurrences, ce qui totalise 44 types distincts. La typologie baseline correspond ainsi aux 44 types non-mixtes les plus fréquents de FreDraCor.

Quant à la typologie principale, ou « typologie générique », ce sont tous les types non-mixtes représentant au moins 50 didascalies, soit 87 types, qui ont été retenus pour être mis en correspondance avec les treize types de cette typologie.

2.3. Conception des jeux de données

Cette section présente étape par étape la conception des deux jeux de données, de l'extraction des didascalies depuis le dépôt FreDraCor (2.3.1) au *mapping* de mise en correspondance des types existants avec les types de la typologie principale (2.3.2) jusqu'à la description des jeux de données (2.3.4, Erreur ! Source du renvoi introuvable.).

2.3.1. Extraction des didascalies et de leur attribut type

Les didascalies ont été extraites depuis le dépôt FreDraCor après leur import dans BaseX. Il est brièvement envisagé d'extraire non seulement la didascalie, son type, mais aussi les éléments suivants et précédents avec leurs contenus. Cependant, seule l'extraction du contenu textuel de la didascalie avec son type s'avèrera utile. Le code entier en XQuery se trouve dans le document [extraction_stgdir.xq](#) du dépôt.

⁶³ <http://www.theaville.org>

⁶⁴ <https://thealtres.pages.unistra.fr/>

```
declare namespace tei="http://www.tei-c.org/ns/1.0";
let $corpus:=db:get('fredracor')
for $x in $corpus//tei:stage return concat(normalize-space($x), '|', $x/@type)
```

Cette extraction, qui contient toutes les didascalies avec leur type original, est enregistrée au format CSV avec comme séparateur le signe « | ».

2.3.2. Création du jeu de données de la typologie principale : réannotation

Afin de limiter l'annotation manuelle, aussi bien pour le jeu de données de la typologie principale que pour la *baseline* nous nous sommes basés sur le compte par valeur unique des types de FreDraCor extrait avec des commandes shell (voir 1.4.3.1.). Les valeurs mixtes ont été exclues afin de simplifier la classification.

Le critère retenu pour la typologie baseline est le nombre d'occurrences minimal de 100 exemples dans FreDraCor avant suppression des doublons. Cette typologie ne nécessite donc pas de réannotation. Ce sont ces didascalies, au nombre de 25 110 qui constituent notre jeu de données de référence. L'extraction est effectuée avec le programme [extraction Baseline.py](#).

L'implémentation de la typologie principale sans passer par une réannotation manuelle de l'intégralité du corpus repose aussi sur les valeurs originales de FreDraCor. La table de correspondance (Tableau 3) est appliquée comme un *mapping*, en Python. Ce programme est disponible dans [conversion_table dracor2new_label.py](#).

Typologie principale	Types de FreDraCor
action	action, watch, show, paint, pray, jump, read, kneel, fall, knock, write, drink, search, open, eat, sleep, stand, sit, move, listen, ring
aggression	kill, fight, hit, suicide, threat
aparte	aparte, alone
delivery	together, call, interrupt, loud, low, laugh, silence, quiet, cry, shout, nervous, ironic, anger, serious, happy, hesitate, enthousiasm, emotion, emphasis, friendly, grimace, feeling, furious, continue, sing, repeat
entrance	entrance, entrée
exit	exit, escape
interaction	kiss, touch, help, pull, push
movement	closer, away, walk, follow, back
music	music, title, ttitle, bis
narration	narration, meteo, noise
object	costume, throw, tear, get, give, dress, drop, close
setting	location, decor, décor
toward	toward

Tableau 3 Table de correspondance entre les 13 classes de la typologie et les 87 types de FreDraCor

Le programme [conversion table dracor2new label.py](#) sélectionne depuis le document contenant les didascalies non-mixtes les 87 types. Le programme conserve aussi une trace des didascalies traitées avec leur étiquette originale afin d'appuyer l'analyse par classes dans l'optique d'une révision de la table de correspondance.

2.3.3. Suppression des doublons et division des jeux de données pour l'entraînement, la validation et l'évaluation

Pour des raisons pratiques, la suppression des doublons se fait avec la méthode `drop_duplicates` de la librairie pandas dans le notebook [dataset creation.ipynb](#).

Les jeux de données au complet sont présents dans le dossier [data](#) du dépôt. Ce dossier contient les didascalies extraites directement de FreDraCor ([FreDraCor](#)), ces-mêmes didascalies avec la typologie principale ([Generic13](#)) et les didascalies avec la typologie baseline ([Baseline44](#)).

Cette étape se fait conjointement à la création des jeux d'entraînement, de validation et d'évaluation (aussi appelé jeu de test) pour la baseline et pour la typologie principale. La séparation des données de validation, utilisée aussi lors de l'entraînement, et des données d'évaluation, complètement distinctes, est privilégiée pour l'évaluation des modèles. C'est pourquoi la méthode `train_test_split` est employée à deux reprises. Les jeux de test correspondent à 20% du jeu entier, et le jeu d'entraînement et validation correspond au 80% restants.

Ces données sont enregistrées dans le dossier [data train](#).

2.3.4. Compte des occurrences par jeux de données

Dans cette section sont présentés les deux jeux de données créés à partir des typologies présentées précédemment. Un premier tableau présente le compte d'occurrences par type classé par ordre alphabétique (Tableau 4, Tableau 6) puis un second tableau présente le nombre de didascalies dans chacun des jeux de données après leur division en jeu d'entraînement, de validation et de test (Tableau 5, Tableau 7).

2.3.4.1. Typologie principale

Le compte détaillé du jeu de données classées par ordre décroissant du nombre d'occurrences par type est présent dans le dépôt git dans le dossier : [data/Generic13](#).

Typologie principale	Compte par type	Compte sans doublons
action	3 560	2 467
aggression	455	350
aparte	1 349	72
delivery	3 072	962
entrance	730	646

exit	3 689	1 295
interaction	772	565
movement	733	583
music	4 437	2 863
narration	576	554
object	1 208	1 130
setting	1 025	982
toward	4 216	2 144
Total	25 823	14 613

Tableau 4 Nombre de didascalies par type du jeu de données de la typologie principale

Jeu	Compte d'occurrences
Entraînement	9 352
Validation	2 338
Test	2 923
Total	14 613

Tableau 5 Nombre de didascalies par jeu pour la typologie principale

2.3.4.2. Typologie baseline

Le compte détaillé du jeu de données classé par ordre décroissant du nombre d'occurrences par type est présent dans le dépôt git dans le dossier : [data/Baseline44](#). Le Tableau 6 reproduit le compte dans l'ordre alphabétique et le Tableau 7 présente le compte de didascalies selon les divisions en jeux d'entraînement, de validation et de test.

Catégorie	Compte par type	Compte sans doublons
SHOW	605	4
aparte	1 276	47
away	281	187
bis	270	16
call	105	42
closer	286	263
cry	160	74
danse	305	190
decor	812	784
dress	234	222
drink	110	46
entrance	725	644
exit	3 632	1 261
explicit	408	378
fight	146	121
get	340	316
give	327	301
hide	105	90
hit	128	105

kiss	371	203
kneel	185	119
knock	119	77
laugh	198	89
location	207	193
loud	1 184	62
low	314	90
music	291	227
narration	383	382
noise	169	151
open	163	141
read	548	181
silence	165	69
sing	274	140
sit	267	163
stand	118	118
stop	105	64
threat	107	82
title	3 870	2 645
together	102	42
touch	225	194
toward	4 207	2 141

watch	684	620
wave	349	315

write	111	53
Total	25 110	13 652

Tableau 6 Nombre de didascalies par type du jeu de données de la typologie baseline

Jeu	Compte de didascalie
Train	8 736
Validation	2 185
Test	2 731
Total	13 652

Tableau 7 Nombre de didascalies par jeu pour la typologie baseline sans doublon

2.3.5. Comparaison des jeux de données par typologies

Le jeu de données est hétérogène, quelle que soit la typologie. Cette hétérogénéité concerne aussi bien l'aspect quantitatif que qualitatif.

Les classes les mieux pourvues pour nos données d'entraînement sont :

- *toward*, présent dans les deux typologies, qui a plus de 4 000 exemples, mais se trouve réduit de moitié sans les doublons ;
- *title*, pour la baseline avec plus de 2 000 exemples ;
- *action*, pour la principale avec plus de 2 000 exemples aussi.

Les classes les moins bien pourvues sont :

- *aparte*, pour la typologie principale, qui passe de 1 349 exemples à 72 après suppression des doublons ;
- *bis*, pour la baseline, qui passe de 270 à 16 exemples après suppression des doublons.

Dans les deux cas, la réduction importante pour les catégories les moins pourvues s'explique par le fait que ces didascalies ont généralement peu de variation avec des phrases types comme « à part » ou simplement le mot « bis ».

Pour ces classes peu pourvues, la quantité peu importante de données d'entraînement signifie que les modèles auront peu d'informations sur lesquelles se baser pour inférer de nouveaux cas.

L'écart quantitatif important entre les classes signifie que la classification automatique risque d'être très inégale. En cela, les typologies sont comparables.

Qualitativement, les jeux de données des deux typologies sont aussi très hétérogènes. Pour la typologie principale, les éléments détaillés ont été relevés en 2.1. Pour la typologie *baseline*, nous ne rentrerons pas dans le détail de la description des types puisque l'annotation est en elle-même assez précise l'étiquette seule suffit à déterminer le contenu sémantique de la didascalie.

Par conséquent, sur le plan qualitatif, les typologies sont moins comparables. La typologie principale a été pensée pour regrouper des phrases sémantiquement proches, tandis que la typologie *baseline* répond d'une logique quantitative pratique liée aux conditions d'implémentation du projet initial Théâtre Classique. Une hypothèse de départ est que cette typologie est en théorie moins apte à une généralisation, car beaucoup plus précise et dotée d'un nombre bien plus important de catégories.

2.4. Entraînement des modèles d'apprentissage automatique

Les sections suivantes présentent en détail les modèles d'apprentissage automatique comparés. Cette présentation préface les paramètres de prétraitement des données et les paramètres d'apprentissage des modèles.

2.4.1. ML classique

Quatre algorithmes implémentés dans `scikit-learn` ont été retenus dont trois modèles linéaires (régression logistique, SVM avec entraînement SGD, classifieur Ridge) et un ensemble d'arbre de décision (Random Forest).

2.4.1.1. Modèles linéaires

L'algorithme d'apprentissage automatique régression logistique⁶⁵ est utilisé pour la classification supervisée, contrairement à ce que son nom suggère. Il modélise la probabilité qu'une variable binaire prenne une certaine valeur en fonction des variables indépendantes et établit à partir de là la meilleure courbe logistique qui sépare au mieux les différentes classes dans l'espace des caractéristiques. La prédiction en sortie est la somme pondérée selon des coefficients appris lors de l'entraînement des caractéristiques pour l'entrée donnée (Müller & Guido, 2016, p.47).

Le Support Vector Machine (SVM)⁶⁶, est un modèle qui repose sur une méthode de ML dont l'objectif est de trouver un hyperplan qui sépare les données d'entrée en classes distinctes avec une marge maximale. L'optimisation de la fonction de perte d'un SVM peut être réalisée à l'aide de l'algorithme de descente de gradient stochastique (Stochastic Gradient Descent, SGD)⁶⁷. L'algorithme SGD consiste à itérer sur l'ensemble de données d'apprentissage et à mettre à jour les paramètres du modèle à chaque itération en fonction de la dérivée de la fonction de perte pour un seul échantillon de données (d'où le terme "stochastique") (Hastie, Friedman, et al., 2009, p.417).

⁶⁵ https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression

⁶⁶ <https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html>

⁶⁷ https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier

Enfin, le classifieur Ridge est un modèle de régression linéaire régularisée, qui ajoute une pénalité de régularisation à la fonction de coût. Cette pénalité aide à prévenir le surajustement en limitant la magnitude des coefficients de régression⁶⁸. La pénalité est une fonction de la norme L1 ou L2 des coefficients de régression. La norme L1 correspond à la somme des valeurs absolues des coefficients de régression, tandis que la norme L2 correspond à la somme des carrés des coefficients de régression. La régularisation L1, également appelée régularisation Lasso, a pour effet de réduire les coefficients de régression à zéro, ce qui permet de sélectionner automatiquement les variables d'entrée les plus importantes pour la prédiction. La régularisation L2, également appelée régularisation Ridge, a pour effet de réduire les coefficients de régression vers zéro, ce qui permet de réduire la variance du modèle et d'améliorer sa généralisation. Dans notre cas, après expérimentation des paramètres de régularisation, c'est la pénalité L2, qui a été conservée pour le classifieur Ridge.

Les paramètres de régularisation ont été expérimentés dans le notebooks [test_regularization-param LRmodels.ipynb](#), pour les valeurs C du Logistic Regression et pour la pénalité L1 et L2 du Ridge Classifier.

2.4.1.2. Modèle d'ensemble d'arbres de décision

Random Forest ⁶⁹ est un algorithme d'apprentissage automatique supervisé qui utilise une méthode d'ensemble pour construire un ensemble d'arbres de décision. Chaque arbre de décision est construit sur un échantillon aléatoire de données d'entrée et utilise un sous-ensemble aléatoire de variables pour prendre des décisions de partitionnement. Chaque arbre de décision produit une prédiction pour chaque instance en entrée. Les prédictions de tous les arbres sont ensuite agrégées pour produire une prédiction finale. Dans le cas de la classification, cela se fait par vote majoritaire pondéré par la probabilité estimée par chaque arbre (Hastie, Friedman, et al., 2009, p.587).

2.4.2. Prétraitement des données pour la vectorisation

Le Tableau 8 présente les caractéristiques retenues pour la représentation vectorielle des didascalies.

Forme de la donnée	Vectorisation
Token	TF-IDF
Étiquette morphosyntaxique (POS)	TF-IDF
Bigramme (tokens)	TF-IDF

⁶⁸ « This model solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm. Also known as Ridge Regression or Tikhonov regularization. » https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

⁶⁹ https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html#sklearn.linear_model.RidgeClassifier

⁶⁹ <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier>

N-gram de caractères (séquence de 2 à 4 caractères)	TF-IDF
Longueur de la chaîne	One-Hot (DictVectorizer) et Normalisation avec MinMaxScaler

Tableau 8 Prétraitement et vectorisation des données

Le seuil minimal de présence d'un token/bi-gramme etc. est fixé à 1% de l'entièreté des documents (didascalies) du corpus (paramètre `min_df`).

Les différentes formes de prétraitement sont opérées avec le modèle français `fr_core_news_sm` de spaCy. L'antidictionnaire retenu est aussi celui de spaCy⁷⁰.

Pour la forme des tokens à conserver, plusieurs prétraitements ont été envisagés et éprouvés pour la classification dans une première phase de recherche. Le notebook [test mise-en-place-ColumnTransformers.ipynb](#) retrace les possibles prétraitements sur le jeu de donnée ainsi que les rapports de classification des prédictions opérées par cinq algorithmes classiques sur ces données en fonction de leurs prétraitements. Il a été envisagé de conserver les tokens sous forme lemmatisée ou racinisée, mais la tokenisation simple avec les données conservées comme mot-forme s'est avérée la méthode la plus avantageuse lors de ces tests préliminaires, c'est la raison pour laquelle toutes les caractéristiques se basent sur ce prétraitement des données. Les notebooks d'entraînement contiennent une trace de ses divers essais à l'étape de prétraitement et de vectorisation : on y trouve les fonctions de transformation des données pour la racinisation et la lemmatisation de spaCy et de NLTK, ainsi que les *stopwords* de NLTK.

La représentation vectorielle avec une pondération basée sur le Term Frequency inverse Document Frequency (TF-IDF) montre dès les premiers essais des performances bien supérieures à la simple représentation vectorielle par comptage non pondéré. Pour cette raison, la méthode de vectorisation `CountVectorizer` n'apparaît pas dans le notebook.

Le modèle voit ses performances améliorées avec l'ajout de caractéristiques supplémentaires telles que le séquençage des tokens en bigrammes et en n-grammes de caractères (séquences de 2 à 4), ainsi que la prise en compte de l'étiquette morphosyntaxique (*part-of-speech*, POS). Ces transformations peuvent servir à mettre en avant les caractéristiques morphologiques des termes qui composent les didascalies, notamment les n-grammes de caractères qui permettent d'identifier plus précisément les morphèmes dérivationnels qui traduisent les temps verbaux. Ces caractéristiques semblent en mesure de valoriser les spécificités de notre jeu de données puisque les performances de tous les modèles s'en trouvent améliorées significativement (passage d'une moyenne-macro F1 entre 0,42 et 0,49 à 0.66 et 0,76).

La longueur de la chaîne de caractères n'améliore pas les performances du modèle, mais est restée dans la chaîne de prétraitement.

⁷⁰ Liste disponible sur le git de spaCy :

https://github.com/explosion/spaCy/blob/master/spacy/lang/fr/stop_words.py

Ce cheminement est disponible dans le notebook [mise-en-place-ColumnTransformers.ipynb](#).

Une caractéristique, celle de la détection des entités nommées est mise de côté, car elle n'améliore pas les performances et allonge significativement le temps de traitement des données. La reconnaissance d'entités nommées est suggérée comme caractéristique par Dennerlein et al. (2016) en conclusion de leur travail de classification automatique. Cette caractéristique est implémentée soit en tant que booléen soit en tant que nombre d'entités trouvées, soit en tant que dictionnaire qui reflète le nombre d'entités trouvées pour chaque type dans le notebook [test_EntiteNommee.ipynb](#).

2.4.3. Apprentissage par transfert

Les modèles suivants ont été affinés avec les données d'entraînement des deux jeux de données présentés :

- camembert-base⁷¹ (Martin et al., 2020) préentraîné sur les données du corpus OSCAR (Ortiz Suárez et al., 2019)
- distilcamembert-base⁷² (Delestre & Amar, 2022) une version distillée de camembert-base.
- bert-base-multilingual-cased⁷³ (Devlin et al., 2019)
- bert-base-multilingual-uncased⁷⁴ (Devlin et al., 2019)
- distilbert-base-multilingual-cased⁷⁵ (Sanh et al., 2020), une version distillée de bert-base-multilingual-cased.
- SetFit⁷⁶ (Tunstall et al., 2022)

Les notebooks se trouvent dans le dépôt GitLab dans le dossier [classifieurs/notebooks_HF](#).

Une partie de ces expérimentations a fait l'objet d'une publication (Schneider & Ruiz, 2024). L'article présente uniquement la typologie principale et se concentre sur l'affinage de six modèles de *transfer learning* en plus des quatre modèles mentionnés dans la section précédente. Ces travaux ont pour objectif de cerner au mieux le nombre minimal d'annotation humaine nécessaire à l'entraînement d'un classifieur automatique. Le jeu de données a alors été réduit à 50%, 25%, 10% et 5% de sa totalité.

Les expérimentations de *transfer learning* ont été effectuées sur Google Colab avec un GPU V100 pour la typologie principale et avec un GPU T4 pour la typologie baseline.

Afin de tenir compte du facteur aléatoire lors de la phase d'apprentissage de ces modèles, l'affinage a été répété cinq fois pour chacun de ces modèles dans le cas de la typologie

⁷¹ <https://huggingface.co/almanach/camembert-base>

⁷² <https://huggingface.co/cmarkea/distilcamembert-base>

⁷³ <https://huggingface.co/google-bert/bert-base-multilingual-cased>

⁷⁴ <https://huggingface.co/google-bert/bert-base-multilingual-uncased>

⁷⁵ <https://huggingface.co/distilbert/distilbert-base-multilingual-cased>

⁷⁶ <https://huggingface.co/SetFit>

principale. Les moyennes-macros F1 présentées dans la partie résultat (Tableau 10 et Tableau 12) correspondent à la moyenne des cinq entraînements. Cependant, pour des raisons de lisibilité, les notebooks présentés dans le dépôt git (voir [le dossier contenant les notebooks](#)) correspondent tous au dernier affinage effectué (cinquième *run*). Pour la typologie baseline, pour des raisons de temps, l'affinage n'a été effectué qu'une seule fois par modèle.

Ces différences entre les deux typologies au moment de l'entraînement des modèles, le fait que les modèles aient été entraînés sur des GPU différents et le fait que les nombres d'entraînements effectués ne soient pas identiques, représentent des limites à la généralisation des conclusions produites dans ce mémoire quant à la comparaison des deux typologies comparées. Cependant, il apparaît utile de fournir ces résultats à titre exploratoire, notamment concernant la typologie baseline.

2.4.4. Présentation des modèles et des hyperparamètres d'entraînement

BERT (Bidirectional Encoder Representations from Transformers) est un modèle de langue conçu sur une architecture Transformers et développé par Google AI. Les Transformers sont des modèles d'apprentissage automatique capables d'encoder de manière bidirectionnelle une fenêtre contextuelle allant dans le cas de BERT jusqu'à 512 tokens. Parce que l'encodage est bidirectionnel, l'architecture des modèles BERT prend en compte le contexte gauche et le contexte droit d'un token, contrairement à des modèles comme word2vec⁷⁷ qui encode les tokens avec une approche sac de mots (*bag-of-words*). Les Transformers ne sont pas les premiers à avoir opéré un encodage bidirectionnel pour construire la représentation vectorielle d'une langue. Les algorithmes Global Vectors for Word Representation (GloVe) (Pennington et al., 2014) et FastText (Bojanowski et al., 2017) en sont de bons exemples. Cependant, ces modèles ne disposent pas du mécanisme d'attention proposé par Vaswani et al. (2017). Le mécanisme d'attention utilisé par BERT est « multi-tête » ce qui permet notamment de capturer les dépendances syntaxiques et sémantiques à longue distance entre les tokens. Les précédents modèles étaient jusqu'à lors limités par une fenêtre contextuelle fixe, en plus d'avoir été entraînés sur des volumes de données plus petits.

Autrement dit, les modèles de langue produits par des architectures Transformers permettent à un même mot d'avoir plusieurs représentations vectorielles selon son contexte, ce qui n'est pas le cas des modèles précédents. BERT est à l'origine un modèle préentraîné sur des données issues du web qui restent majoritairement en langue anglaise, mais des modèles BERT multilingues, c'est-à-dire entraînés sur des données dans d'autres langues que l'anglais ont ensuite été distribués. C'est la raison du développement de leur versions monolingues camembERT et flauBERT par exemple pour le français.

SetFit est un modèle d'apprentissage contrastif qui repose sur un processus d'entraînement en deux étapes, impliquant l'affinage d'un Sentence Transformer (Reimers & Gurevych, 2019) sur un très petit nombre d'exemples étiquetés qu'il regroupe en paires positives et négatives, suivi de l'entraînement du modèle principal (au choix) sur les plongements de mots générés à partir du Sentence Transformer affiné (Tunstall et al., 2022). Le Sentence Transformer affiné

⁷⁷ <https://www.tensorflow.org/text/tutorials/word2vec>

pour ce projet est `distiluse-base-multilingual-cased-v1`⁷⁸. SetFit n'a été entraîné qu'une seule fois, contrairement aux autres modèles d'apprentissage par transfert, car ce modèle n'introduit pas de variable aléatoire lors de l'entraînement, autrement dit, il reproduit le même entraînement à l'identique et ne produit pas variation dans ses prédictions d'un apprentissage à un autre.

Les modèles ont été entraînés sur une base de 10 époques maximum avec une fonction d'*early stopping*, qui surveille la perte de validation pendant l'entraînement. L'*early stopping* permet de détecter le surapprentissage et d'arrêter l'entraînement lorsque la perte de validation cesse de s'améliorer pendant un certain nombre d'époques consécutives, ce nombre est appelé "patience". La fonction de perte mesure l'écart entre les prédictions du modèle et les valeurs réelles. Dans notre cas, la patience est de 3 époques ce qui signifie que le modèle arrête l'entraînement si la perte de validation n'a pas diminué au cours des trois dernières époques afin de limiter le surapprentissage. Le Tableau 9 présente le nombre d'époques maximum réalisées lors de l'apprentissage pour chacun des modèles d'apprentissage par transfert au cours des cinq entraînements. Notons à ce sujet qu'aucun entraînement sur la typologie principale n'est allé au-delà de 7 époques. Quant à la typologie baseline, les chiffres présentés dans le Tableau 9 ne sont pas les maximales, mais seulement le nombre d'époque effectué lors du seul entraînement effectué sur ces données.

Modèle	Typologie principale*	Typologie baseline
camembert	7	10
distilcamembert	5	10
bert-base-multilingual-cased	6	9
bert-base-multilingual-uncased	6	7
distilbert-base-multilingual-cased	6	8

Tableau 9 Nombre d'époques lors de l'affinage

*Il s'agit du nombre maximal sur les cinq entraînements pour la typologie principale.

SetFit, ne disposant pas de fonction d'arrêt (*early stopping*), a été entraîné sur 6 époques avec 20 itérations lors de la phase d'apprentissage itératif (soit la première étape de l'entraînement du S-BERT).

2.4.5. Modèles génératifs

Dans la mesure où en l'espace d'un an, l'utilisation des LLMs génératifs s'est largement répandue, et que leur utilisation est facilitée par des API comme des sites internet et applications dédiés, il semble pertinent de comparer leur performance sur cette tâche de classification.

Les modèles comparés sont :

- gpt-3.5-turbo-0125 d'OpenAI (Brown et al., 2020)
- open-mixtral-8x7b⁷⁹, aussi appelé « mistral small » de Mistral AI

⁷⁸ <https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1>

⁷⁹ <https://huggingface.co/mistralai/Mistral-7B-v0.1>

Ces modèles n'ont pas été téléchargés localement ni affinés, l'objectif étant de comparer les modèles précédemment cités, qui nécessitent plusieurs étapes de prétraitements, de vectorisation et d'entraînement, à des modèles dits conversationnels qui ne nécessitent pas de transformation des données de la part de l'utilisateur. Pour cette raison et pour automatiser l'envoi de requêtes, ces deux modèles ont été employés via leurs API respectives⁸⁰. Cette stratégie est aussi justifiée par le coût qui est calculé avec l'API sur la base du nombre de tokens en entrée (*input*) et en sortie (*generated output*) et non sur la durée d'utilisation d'une machine dotée de GPU, comme ça aurait été le cas si les modèles avaient été téléchargés. Les requêtes envoyées (prompt avec la didascalie à classer) et les sorties attendues totalisent un nombre de tokens que l'on peut calculer : le coût total de l'expérimentation est donc estimable avec précision, ce qui peut aussi être intéressant dans une optique de mise en application sur un nouveau corpus ou une nouvelle typologie.

2.4.5.1. Technique de *prompting*

Le prompt engineering est le processus de création et d'amélioration d'un prompt, ou requête, envoyé à un LLM afin d'obtenir une meilleure génération, et ce notamment dans le cas des IA conversationnelles. Le prompt engineering permet de documenter des motifs ou techniques qui peuvent être adaptés à des tâches de NLP (Marvin et al., 2024).

Les techniques de *prompting* explorées sont le zero-shot, one-shot et few-shots (two-shots). Ces deux dernières sont des techniques dites « d'apprentissage en contexte » (*in-context learning*).

Le *zero-shot* vise à faire exécuter une tâche sans fournir d'exemples d'entraînement spécifiques. Dans ce cas, le prompt est :

*Put a label to the following stage direction using one of the categories listed below:
Stage direction : {stgdir}
Categories : ['action', 'aggression', 'aparte', 'delivery', 'entrance', 'exit', 'interaction',
'movement', 'music', 'narration', 'object', 'setting', 'toward'] \n\n
You only have to give the label, no explanation or notes are needed.
Remember, you can only use one of the labels listed above.*

Des essais ont été faits sur un prompt en français sans noter empiriquement de différence dans l'exécution des instructions données. Pour des questions de généralisation et parce que les types sont donnés en langue anglaise, il a été convenu de maintenir un prompt en anglais au cours des expérimentations.

Ce prompt est utilisé comme base pour les deux autres techniques, à laquelle ont été ajoutés des exemples après la liste des catégories. L'indication finale est une répétition que l'on retrouve dans de nombreux prompts de Harrison Chase ingénieur chez LangChain qui rédige

⁸⁰ <https://docs.mistral.ai/api/>

les prompts^{81,82} proposés dans la documentation de LangChain, *framework* de développement d'application intégrant des LLMs.

Le *one-shot* et *few-shots* sont des techniques qui, au contraire du *zero-shot*, reposent sur une contextualisation de la tâche à partir d'exemple(s). Le *one-shot* consiste à ne fournir qu'un seul exemple et *few-shots* plusieurs. Dans notre cas, seuls deux exemples ont été donnés dans la stratégie *few-shots*, on la nomme alors *two-shots*.

Les stratégies de prompting suivantes ont été explorées, mais laissées de côté.

La Structured information est une stratégie qui vise à imposer un format de réponse. Liu et al. (2021) notent une amélioration qualitative des réponses produites grâce à une demande de formatage de la sortie générée.

Le prompt testé dans notre cas est identique à celui du *two-shots* avec une demande supplémentaire de sortie en format JSON :

```
This is the stage direction you have to classify: {stgdir}
Output your answer in this format:
{{
'cat': PREDICTION
}}
```

Le *Reasoning and Acting* (ReAct) (Yao et al., 2023) est une technique qui vise à combiner la capacité de réflexion et de prise de décision d'un LLM. Ce prompt repose sur une stratégie type Chain of Thoughts (CoT) qui consiste à faire générer les différentes étapes qui mènent le LLM vers une réponse finale⁸³. Ce genre de technique se prête davantage à des tâches complexes qu'à une tâche de classification puisqu'elle invite à présenter toutes les étapes de la réflexion.

Afin de limiter le nombre d'expérimentations et simplement pour donner une idée du potentiel des modèles génératifs sur une typologie plus vaste, la typologie baseline n'est comparée que sur la méthode *zero-shot* tandis que la typologie principale fait l'objet de classification avec les prompts *zero-shot*, *one-shot* et *two-shots*.

Les deux dernières stratégies, à savoir *structured information* et ReAct n'ont pas pu être exploitées à cause des sorties générées. Le ReAct tend à donner des réponses très longues qu'il est difficile d'exploiter directement. En effet, plusieurs étiquettes peuvent être énoncées au cours de la phase de réflexion et le modèle ne respecte pas toujours l'instruction qui concerne le format demandé en sortie. Ainsi la réponse finale n'est pas toujours dans le format « Final Answer : {label} », ce qui permettrait l'extraction de l'étiquette. Deux exemples de sorties générées par le modèle de Mistral AI avec ce prompt sont présentés en Annexe B.

⁸¹ <https://github.com/hwchase17>

⁸² <https://smith.langchain.com/hub/hwchase17/structured-chat-agent>

⁸³ Exemple d'une CoT : 1. Thought, 2. Action, 3. Observation, 4. Repeat 1-3 until reaching final conclusion 5. Final answer

Pour le prompt *structured information*, le format JSON est, dans ce cas aussi, rarement respecté, ce qui implique une étape de post-traitement supplémentaire pour récupérer la portion de texte générée qui respecte véritablement le format demandé. Les expérimentations avec ces techniques font apparaître une vraie limite vis-à-vis des modèles génératifs.

En résumé, trois stratégies de *prompting* sont comparées pour la typologie principale, à savoir les techniques de *zero-shot*, *one-shot* et *two-shots*. Et pour la typologie baseline, seule la technique de *zero-shot* a fait l'objet d'une expérimentation complète.

2.4.5.2. Stratégies de post-traitement des sorties générées

Les modèles génératifs n'ont pas pour fonction première la classification à strictement parler et ils sont avant tout très performants dans l'analyse de la langue naturelle dans de larges fenêtres contextuelles. Par conséquent, ces modèles tendent à générer des réponses longues bien que le prompt tente de les restreindre dans le nombre de tokens en sortie ou sur le format de la sortie. Pour pallier à cette « surgénération », nous procédons à une étape de post-traitement par extraction via expressions régulières, stratégie par ailleurs opté par Fina Polat et al. (2023) pour leur évaluation de différentes techniques de prompting appliquées à une tâche d'extraction de connaissance.

Une donnée importante à prendre en compte lors du travail avec les modèles génératifs est le nombre de tokens en sortie (*output*) puisque la sortie n'est pas toujours exploitable telle quelle et ne constitue pas forcément une prédiction. Dans le cadre de ce projet, l'intention a aussi été de confronter différentes stratégies d'exploitation des sorties brutes des modèles génératifs afin d'établir de quelle manière la génération d'une prédiction en langue naturelle peut être automatisée.

Une première stratégie consiste à sélectionner le premier mot en sortie. Sachant que le prompt donne l'instruction explicite de limiter la réponse au seul énoncé d'une étiquette, cette stratégie semble pertinente. Cette stratégie est nommée « méthode premier mot ».

Une seconde stratégie consiste à chercher par une expression régulière, la dernière occurrence d'une étiquette dans la sortie générée. Cette stratégie est nommée « méthode « *n'importe quel mot* » ». Prenons un exemple pour expliciter ces deux stratégies d'extraction. Un modèle a généré pour la didascalie « Elle s'arrange la figure devant la glace. » (étiquette attendue : *object*) en entrée, le texte « delivery or object, depending on whether the focus is on adjusting one's appearance or using the »⁸⁴. La méthode « premier mot » extrait l'étiquette « delivery », et la méthode *n'importe quel mot* sélectionne la dernière occurrence d'une étiquette valide, à savoir « object ».

Cet exemple montre d'ores et déjà une limite des modèles génératifs puisque dans ce cas, deux étiquettes sont « prédites » et que des choix méthodologiques doivent être établis pour pouvoir généraliser leur utilisation.

⁸⁴ La génération est coupée à 20 tokens.

2.4.5.3. Paramètres des modèles génératifs

La génération de tokens est limitée à 20 pour la typologie principale et n'est pas limitée pour la typologie baseline, cela permet indirectement de comparer le comportement des deux modèles vis-à-vis de la génération en elle-même.

La température des modèles est fixée à 0. La température d'un modèle détermine la mesure dans laquelle celui-ci optera pour une approche conservatrice ou créative dans la génération de sa réponse. La température prend une valeur de 0 à 1. Une température basse signifie que le modèle sélectionne le token le plus probable au moment de compléter sa génération, tandis qu'une température (plus) élevée induit un facteur aléatoire dans cette prise de décision ce qui produit des réponses moins attendues, mais aussi moins reproductibles.

L'ensemble du travail de requêtage via l'API avec les trois différents prompts, ainsi que l'extraction des prédictions selon les deux stratégies de post-traitement se trouve dans le programme [generativeAI.py](#).

2.5. Métriques d'évaluation

Les mesures retenues pour l'évaluation sont la précision, le rappel et la mesure F1 non pondérée, soit la moyenne-macro F1. Cette dernière mesure vise à évaluer l'ensemble de la typologie indépendamment du volume de données par classes, tandis qu'une mesure pondérée est susceptible de valoriser les classes les mieux pourvues. Cela permet notamment d'établir la mesure dans laquelle la typologie se prête à une classification automatique. Autrement dit, la moyenne-macro F1 permet d'évaluer si l'ensemble de la typologie peut faire l'objet d'une classification automatique, là où une mesure pondérée mettrait en valeur les classes les mieux dotées qui ne sont pas nécessairement celles que nous cherchons à mettre en valeur pour ce projet de mémoire.

Afin d'évaluer les modèles génératifs, le nombre de tokens envoyés et générés est aussi pris en compte.

Partie 3 : Résultats et discussion

Dans cette partie, les résultats des expérimentations sont présentés dans un premier temps en fonction de la performance des modèles (3.1), suivi, dans un second temps, de l'analyse détaillée des performances par classes (3.2) dans l'ensemble des modèles. À la fin de cette partie sont élaborées les perspectives du projet (3.2.3).

3.1. Comparaison des performances par type de modèles

Le Tableau 10 récapitule tous les modèles employés à la classification sur les deux jeux de données. On retrouve les quatre algorithmes de ML classique (3.1.1), les six modèles d'apprentissage par transfert dont SetFit pour l'apprentissage contrastif (3.1.2), et enfin les deux LLMs génératifs (3.1.3). Les performances des modèles sont discutées en fonction de chacune de ces trois grandes catégories dans les sous-parties suivantes. Tous les rapports de classification se trouvent dans l'Annexe A.

		Typologie			
		Principale		Baseline	
ML classique					
LogisticRegression		0,7031		0,5903	
RidgeClassifier		0,7284		0,6643	
RandomForest		0,6565		0,5847	
SVM avec SGD entraîné		0,7348		0,6526	
Apprentissage par transfert					
camembert-base		0,7777		0,6836	
distilcamembert-base		0,808		0,7266	
bert-base-multilingual-cased		0,803		0,6874	
bert-base-multilingual-uncased		0,810		0,6947	
distilbert-multilingual-cased		0,799		0,6912	
Modèle d'apprentissage contrastif					
SetFit		0,7825		0,6727	
Modèles génératifs					
		1er mot	n'importe quel mot	1er mot	n'importe quel mot
gpt-3.5-turbo-0125	0 shot	0,17	0,25	0,34	0,48
	1 shot	0,39	0,44		
	2 shots	0,39	0,44		
open-mixtral-8x7b	0 shot	0,21	0,30	0,16	0,42
	1 shot	0,31	0,31		
	2 shots	0,40	0,50		

Tableau 10 Récapitulatif des performances pour tous les modèles testés (moyenne-macro F1)

3.1.1. Modèles de ML classiques entraînés

Modèle	Typologie principale	Typologie baseline
LogisticRegression	0,7031	0,5903
RidgeClassifier	0,7284	0,6643
RandomForest	0,6565	0,5847
SVM (SGD entraîné)	0,7348	0,6526

Tableau 11 Moyenne-macro F1 des classifications des modèles de ML classique

Comme le Tableau 11 le montre, la classification obtient globalement des résultats assez satisfaisants avec un maximum de 0,74 pour l'algorithme linéaire SVM (entraîné avec SGD) pour les deux typologies.

Le modèle le plus performant est le SVM linéaire entraîné avec SGD pour la typologie générique et le Ridge Classifier pour la typologie baseline. Cependant, les performances sont comparables entre les modèles avec un écart de 0,08 et 0,06 maximum pour les deux typologies.

Concernant les typologies elles-mêmes, la typologie principale produit de meilleurs résultats indépendamment de l'algorithme comparé, mais l'écart pour chaque algorithme n'excède pas 0,11 (Logistic Regression) entre les deux.

Cependant ces résultats peuvent être insuffisants pour une annotation définitive quelle que soit la typologie et laisse penser qu'une étape de correction manuelle est nécessaire pour l'encodage d'un corpus. Voir l'annexe A.1 et le rapport de classification de l'algorithme linéaire SVM pour obtenir le détail des prédictions par classe pour ce modèle.

3.1.2. Apprentissage par transfert

Modèle	Typologie principale	Typologie baseline
camembert-base	0,7777	0,6836
distilcamembert-base	0,808	0,7266
bert-base-multilingual-cased	0,803	0,6874
bert-base-multilingual-uncased	0,810	0,6947
distilbert-multilingual-cased	0,790	0,6912
SetFit	0,7825	0,6727

Tableau 12 Moyenne-macro F1 des classifications des modèles affinés d'apprentissage par transfert

Tous les modèles d'apprentissage par transfert performant mieux que les modèles de ML classiques (Tableau 12) indépendamment de la typologie. En annexe se trouvent les rapports de classification pour tous ces modèles (dans l'ordre du tableau : A.2, A.3, A.4, A.5, A.6, A.7).

Modèle	Écart-type
camembert-base	0,0106
distilcamembert-base	0,0119
bert-base-multilingual-cased	0,0081
bert-base-multilingual-uncased	0,008
distilbert-multilingual-cased	0,0044

Tableau 13 Écart-type des moyennes-macros F1 des 5 entraînements sur la typologie principale des modèles d'apprentissage par transfert

Le Tableau 13 montre une dispersion assez faible entre les cinq entraînements quel que soit le modèle avec un écart-type n'excédant pas 0,0119 (*distilcamembert-base*). On peut par conséquent considérer que les moyennes-macros F1 du tableau précédent (Tableau 12) sont représentatives des capacités globales des modèles sur le jeu de données de la typologie principale. Pour rappel, les entraînements des modèles d'apprentissage par transfert pour la typologie baseline n'ont été effectués qu'une seule fois et le modèle par apprentissage contrastif SetFit n'a aussi été entraîné qu'une seule fois, car il n'introduit pas de variable aléatoire lors de l'entraînement.

Le modèle distillé de camembert, *distilcamembert-base*, a produit de meilleures prédictions que sa version non distillée, contrairement à ce que l'on pourrait attendre. Cela est particulièrement marqué pour la typologie baseline. À noter cependant que cela n'est plus vrai lors des expérimentations sur les jeux de données réduits à partir de 25% de l'entièreté du jeu pour la typologie principale (Schneider & Ruiz, 2024). Pour cette même typologie, les modèles multilingues, quant à eux, sont plus cohérents avec des performances globalement moins bonnes pour le modèle distillé que pour les modèles non distillés. Tandis que pour la typologie baseline, le modèle distillé de BERT est tout à fait comparable, voire meilleur, que *bert-base-cased* et *uncased*.

Par ailleurs, les moyennes-macros F1 pour la typologie baseline sont toutes autour de 0,68 à 0,01 d'écart, à l'exception de *distilcamembert*.

La casse joue un rôle très marginal dans la classification à l'échelle de plusieurs milliers de didascalies avec un avantage inférieur à 0,01 pour le modèle BERT multilingue insensible à la casse. Ces conclusions se distinguent de celles tirées par Pagel et al. (2021) pour l'identification des parties de pièces de théâtre en allemand, où les performances des modèles insensibles à la casse étaient sensiblement meilleures. Ils suggèrent que «préserver la casse ne laisse pas le modèle généraliser»⁸⁵ (Pagel et al., 2021, p.224), ce qui n'est pas le cas pour nos données indépendamment de la typologie proposée.

La comparaison des performances selon les deux typologies fait apparaître que la langue principale des données de préentraînement (anglais pour les bert ou français pour camembert), ne joue pas forcément un rôle dans la capacité de prédiction du modèle affiné. En effet, *camembert-base* est le modèle qui obtient les moins bons résultats après affinage, mais ses performances (dans une perspective macro toujours), sont tout à fait comparables à

⁸⁵ Ma traduction.

celles des autres modèles pour les deux typologies avec un écart de 0,04 par rapport au modèle le plus performant dans les deux typologies.

D'après ces observations, et au vu des performances de la version distillée de ce modèle, notamment pour la typologie baseline, le modèle `distilcamembert` semble le plus propice à la tâche de classification supervisée sur des didascalies en français.

3.1.3. Modèles génératifs avec apprentissage avec et sans contexte

Cette sous-partie retrace en deux temps les résultats générés par les modèles génératifs. Dans un premier temps, des observations sur le comportement des modèles comparés mettent au jour les spécificités de chacun des deux modèles (3.1.3.1) et dans un second temps, les performances sur la tâche de classification sont commentées et analysées (3.1.3.2).

Dans le dépôt git, le dossier [scripts_generativeAI](#), contient à la fois le programme qui fait appel aux API ([generativeAI.py](#)), le programme qui permet d'écrire les paramètres du modèle ([count_tokens.py](#)) (nom, *template* de la requête, nombre de tokens en entrée et en sortie pour toute la classification), mais aussi les jeux de données avec les prédictions générées par [gpt](#) et [mistral](#). On trouve en annexe les rapports de classification par modèle (GPT 3.5 en annexe A.8, puis Mixtral 7B en A.9). Les rapports sont ensuite classés par typologie et par stratégie de post-traitement dans l'annexe respective des modèles.

3.1.3.1. Observations sur le comportement des modèles génératifs

Le Tableau 14 fait apparaître l'une des divergences les plus notables entre les deux modèles, à savoir la question de la longueur du texte généré en sortie.

	open-mixtral-8x7b		gpt-3.5-turbo-0125	
	Typologie principale	Typologie baseline	Typologie principale	Typologie baseline
Zero-shot	14 456	80 070**	3 563	3 837**
One-shot	2 944*		3 536	
Two-shots	26 158		3 558	

Tableau 14 Nombre de tokens générés par les modèles génératifs pour les deux typologies

* Le nombre maximal de tokens en sortie est fixé à 4 pour cette expérimentation, au lieu de 20 pour les autres.

**Le nombre de tokens en sortie n'est pas limité pour la typologie baseline.

Les deux modèles agissent de manière bien différente. `open-mixtral-8x7b` génère un nombre de tokens qui varie beaucoup d'une stratégie à une autre et d'une typologie à une autre. À noter que l'expérimentation pour la méthode *one-shot* a un nombre limité de tokens générés à quatre tandis que les autres expérimentations limite ce nombre à 20 et qu'aucune limite sur la génération de tokens n'a été fixée pour la baseline. On peut raisonnablement assumer que le modèle de MistralAI tend à générer des tokens proportionnellement au nombre

de tokens reçus en entrée, contrairement au modèle d'OpenAI qui ne produit pas un nombre significativement plus important de tokens malgré la longueur plus importante du prompt.

Le comportement du modèle de MistralAI peut s'expliquer selon les observations empiriques suivantes :

1. movement toward setting (since looking usually involves some sort of physical movement, even if it's just
2. music
The stage direction ""AIR. Un inconnu."" can be labeled as 'mus »
3. music
User: can you please explain why the label 'music' is appropriate for the stage »,
4. music
User: can you give me 10 stage directions with labels please?

Sure

Le modèle, confronté à un prompt plus long, tend à « réfléchir » davantage, c'est-à-dire à énoncer une réponse plus complète et nuancée (1). Le modèle justifie sa réponse et répète les consignes ou la didascalie (2). Le modèle hallucine et génère un contenu, souvent sous la forme d'une question, qui n'est pas demandée dans le prompt initial, le plus souvent pour anticiper un échange avec l'utilisateur (3, 4).

Cette différence de comportement est aussi à prendre en compte dans une perspective d'encodage automatique puisque non seulement cela engendre un coût plus important, mais aussi et surtout, cela signifie que le modèle de MistralAI nécessite un post-traitement adapté à la technique de prompt. Une analyse plus complète de ce phénomène se trouve dans la section suivante (3.1.3.2).

Parce que les modèles génératifs ont systématiquement ajouté des valeurs hors classe attendues, il n'a pas été possible de produire les matrices de confusion pour les modèles génératifs. Cependant, sont produits en annexe (A.8, A.9) l'ensemble des rapports de classification pour les deux stratégies d'extraction des valeurs prédites (voir 2.4.5.2 pour la description des stratégies de post-traitement « premier mot » et « n'importe quel mot »).

3.1.3.2. Performances selon la méthode de *prompting* et la méthode d'extraction

Les stratégies d'extraction de l'étiquette donnée par les modèles génératifs (voir 2.4.5.2) ne permettent pas toujours d'identifier l'étiquette prédite réellement par le LLM ce qui nous oblige à nuancer l'évaluation de ces modèles. On constate que, d'une part, le modèle peut donner plusieurs étiquettes au lieu d'une (Exemple : « *interaction or movement (depending on the context; it could mean ""Octave is perceived"" or* ») et d'autre part, le modèle peut expliquer pourquoi il ne choisit pas une étiquette, auquel cas c'est tout de même cette étiquette qui est retenue à cause de l'extraction par expression régulière. Par exemple la sortie générée

« 'setting' would not be the correct label for ""Ils s'attablent"" as »⁸⁶ donne tout de même l'étiquette « setting » comme prédiction. C'est avec ces éléments en tête que nous pouvons observer les performances indiquées dans le Tableau 15.

		Typologie			
		Principale		Baseline	
		Méthode d'extraction			
Modèle	Prompting technique	1er mot	N'importe quel mot	1er mot	N'importe quel mot
gpt-3.5-turbo-0125	Zero shot	0.17	0.25	0.34	0.48
	One shot	0.3s9	0.44		
	Two-shots	0.39	0.44		
open-mixtral-8x7b	Zero shot	0.21	0.30	0.16	0.42
	One shot	0.31*	0.31*		
	Two shots	0.40	0.50		

Tableau 15 Moyennes-macros F1 des classifications des modèles génératifs

*Nombre de tokens en sortie limité à 4 pour cette expérimentation.

Un aperçu général des performances (Tableau 15) montre d'emblée que toutes les classifications produites sont insatisfaisantes, quel que soit le modèle, la typologie ou encore la méthode d'extraction de la prédiction. Aucune des expérimentations ne permet d'atteindre une moyenne-macro F1 supérieure à 0,5, ce qui signifie que la classification, telle qu'opérée dans le cadre des expérimentations décrites, ne permet pas l'emploi des modèles génératifs dans une perspective d'encodage automatisé du type de didascalie.

Concernant l'extraction, les deux modèles voient une amélioration de leur performance globale avec la méthode d'extraction « n'importe quel mot » (entre 0,05 et 0,1 de la moyenne-macro F1). Ce qui laisse penser que les consignes du prompt, à savoir de ne fournir en réponse que l'étiquette prédite, sont loin d'être systématiquement respectées. Notons cependant que le GPT réduit de moitié cet écart à partir du moment où on lui fournit un exemple ou plus, ce qui laisse penser que le modèle est plus enclin que Mixtral à produire une réponse formatée quand un exemple lui est donné.

L'unique correspondance des mesures F1 entre les deux stratégies d'extraction n'advient que dans le cas où le nombre de tokens maximum lu en sortie est limité à quatre. À noter aussi que cette limite n'implique pas que le modèle génère une sortie complète dans cette contrainte, mais seulement que le nombre de tokens transmis est lui limité, ce qui donne des phrases en sortie incomplètes comme certains exemples ont pu le montrer (3.1.3.1).

D'après le tableau des performances (Tableau 15), les modèles génératifs sont globalement très peu exploitables avec moins de la moitié des prédictions correctes dans tous les cas et pour les deux typologies. On note tout de même que la technique de prompting a une influence nette dans les performances des modèles, passant de 0,17 et 0,21 respectivement pour le prompt *zero-shot* à 0,44 et 0,50 pour le prompt *two-shots*.

⁸⁶ Toutes les générations sont retranscrites en entier, d'où la troncature à 20 tokens BPE.

De manière générale, le modèle de MistralAI s'améliore à l'ajout d'un exemple puis d'un deuxième dans le prompt, ce qui n'est pas le cas du GPT, dont les performances sont très proches entre le *one-shot* et le *two-shots*. Une analyse plus détaillée de la classification révélera quelles classes sont mieux identifiées grâce à l'ajout d'exemple(s).

Ce qui ressort de cette expérimentation est le fait que la typologie principale est moins bien identifiée que la typologie baseline à 44 classes indépendamment du modèle. En effet, les prédictions pour la typologie baseline générées à partir d'un prompt *zero-shot*, c'est-à-dire sans aucun exemple pour guider le modèle, produisent une classification dont les résultats sont, sans regarder dans le détail des classes, comparables, voire meilleures que les prédictions générées par la typologie principale avec l'appui de deux exemples au moment de la requête. En chiffre, cela correspond aux moyennes-macros F1 de 0,48 pour la typologie baseline avec le prompt *zero-shot* et à 0,44 pour la typologie principale avec le *prompt two-shots* (post-traitement « n'importe quel mot », modèle GPT, pour les deux cas).

Cette différence peut s'expliquer par le fait que la typologie baseline est extraite directement de l'annotation manuelle, qui est restée très proche du contenu sémantique de la didascalie annotée. Autrement dit, « il l'embrasse » est plus facilement identifiable comme étant « kiss » que « interaction » sans notre guide d'annotation, pour un humain comme pour un modèle de langue.

L'importance du choix de l'intitulé pour la stratégie de *prompting* sans apport d'exemple(s) semble plus marquée pour les types ayant un intitulé plus « précis » dans le sens où cet intitulé explicite le contenu de la didascalie de manière moins générale. Ainsi la classification *zero-shot* de la typologie principale fait apparaître la difficulté à classer les didascalies dans la catégorie *action*. À l'inverse, dans la typologie baseline, ce sont les didascalies *kiss*, *kneel*, *cry*, *laugh*, décrivant des actions précises, qui sont les mieux identifiées.

Le choix de l'intitulé des classes a un impact variable en fonction de la classe elle-même. Si on constate que l'ajout d'un ou deux exemples permet d'améliorer de manière remarquable la classification des didascalies *action*, ce qui laisse penser que l'information « action » n'est pas suffisamment précise pour pouvoir être associée à des exemples précis, cela n'est pas vrai de toutes les classes. Ainsi le type *movement* reste parmi les moins bien classés avec une précision ne dépassant pas les 0,28 pour les deux modèles quelle que soit la technique de *prompting*, bien que l'intitulé « mouvement » suggère un contenu bien moins généralisable que ne l'était « action ».

Quant aux exemples, leur ajout est un facteur non négligeable d'amélioration, comme on le voit dans les expérimentations opérées (passage d'un moyenne-macro F1 de 0,25 à 0,44 pour le modèle de MistralAI, et de 0,20 à 0,50 pour le modèle d'OpenAI). Mais cette amélioration n'est pas proportionnelle pour toutes les classes.

Le type *narration*, que l'on trouve tel quel dans les deux typologies, est globalement mal identifié par les deux modèles sollicités avec une technique de *prompting zero-shot*. L'identification de ce type repose sur des critères tels que la longueur du texte et les temps verbaux (voir 2.1.4.3), et l'intitulé « *narration* » ne réfère pas directement au contenu thématique de la didascalie contrairement aux autres types de la typologie principale. Par

conséquent, il n'est pas étonnant que la stratégie *zero-shot* ne présente pas des résultats probants pour ce type puisque dans une classification *zero-shot* l'intitulé de la classe est la seule information fournie au modèle. En revanche, on peut supposer que l'ajout d'exemples aide à identifier les traits caractéristiques de ce type. Or, on n'observe pas un comportement homogène après l'ajout d'un ou deux exemples pour ce type. Le modèle mixtral n'améliore que de manière marginale ses prédictions (amélioration de la moyenne-macro F1 de 0,04 entre *zero* et *two-shots*) contrairement au modèle GPT dont la moyenne macro F1 passe de 0,08 à 0,48.

3.2. Comparaison de la classification selon les classes

Cette seconde sous-partie de l'analyse des résultats se penche en détail sur les classifications prédites par les différents modèles en fonction des deux typologies. Elle vise à établir quelles sont les classes les plus propices à l'identification et à la classification, ainsi que celles qui ont généré des groupes d'erreurs afin de supposer les points forts et les points faibles des typologies proposées.

Il n'a pas été possible de générer les matrices de confusion des modèles génératifs, car le nombre de catégories prédites ne correspond jamais au nombre de catégories réelles. Les commentaires au sujet de ces modèles portent ainsi uniquement sur les rapports de classifications. Pour les autres modèles, seules les matrices de confusion normalisées sont reportées ici. L'ensemble des matrices de confusion se trouve dans le dépôt GitLab au format png dans les sous-dossiers « *confusion_matrices* » (matrices de confusion des [algorithmes classiques](#), et des [modèles d'apprentissage par transfert](#)).

Afin d'alléger l'analyse, et parce qu'il est le modèle ayant produit les meilleurs résultats pour la catégorie des algorithmes classiques (voir 3.1.1), seules les prédictions de l'algorithme linéaire SVM linéaire (entraîné avec SGD) sont comparées aux autres modèles dans cette analyse.

3.2.1. Typologie principale

Cette sous-section analyse la typologie principale sous l'angle de la classification automatique par les différents classificateurs comparés. Le jeu de test est présenté en fonction de la représentativité des classes (3.2.1.1) puis une analyse des classes les mieux identifiées et les moins bien identifiées est avancée (3.2.1.2) et enfin, est donnée une analyse des confusions les plus communes aux différentes classifications (3.2.1.3).

3.2.1.1. Remarque sur l'échantillonnage des classes

Types	Occurrences dans le jeu de test
action	512
aggression	70
aparte	16
delivery	190

entrance	128
exit	264
interaction	110
movement	119
music	582

narration	103	setting	178
object	209	toward	442

Tableau 16 Nombre d'occurrences par type dans le jeu de test de la typologie principale

D'après le Tableau 16, les types sont représentés de manière très inégale dans le jeu de test de la typologie principale. Ces inégalités respectent les proportions du jeu d'entraînement, mais cela peut, à l'étape d'analyse, biaiser la vision d'ensemble des prédictions. En effet, les classes les moins pourvues dans une matrice de confusion normalisée sont susceptibles de produire des statistiques extrêmes. Autrement dit, une catégorie d'une dizaine d'exemples aura plus de chance d'obtenir une précision très élevée ou très basse qu'une catégorie de plusieurs centaines d'exemples. Dans le cas de la typologie principale, cela concerne surtout la classe *aparté*, mais aussi, dans une moindre mesure, les classes *aggression*, *interaction* et *movement*.

3.2.1.2. Classes les mieux identifiées et les moins bien identifiées

Les exemples les mieux classés indépendamment des modèles ou de la stratégie de classification sont *music*, *toward* et *exit* qui obtiennent systématiquement les meilleurs scores F1. Tous les modèles entraînés sur l'ensemble du jeu d'entraînement (SVM linéaire et tous les BERT) identifient ces classes avec une moyenne F1 d'au moins 0,86. Pour les modèles qui n'ont bénéficié que d'un ou deux exemples (c'est-à-dire les modèles génératifs), ces classes font aussi partie des mieux identifiées mais avec une plus grande variation de la moyenne F1: de 0,55 pour *toward* avec *open-mixtral-8x7b zero shot* à 0,92 pour *music* avec *gpt-3.5-turbo-0125* et *open-mixtral-8x7b* en *two-shots*. La classe *toward* est particulièrement bien identifiée par les modèles de type BERT avec un rappel allant jusqu'à 0,98 indépendamment des données de préentraînement. Ces résultats sont cohérents par rapport à nos attentes : ces trois classes sont parmi les mieux dotées de la typologie et sont certainement plus aptes à être identifiées de par leurs caractéristiques lexicales et syntaxiques, ce qui est corroboré par les précisions du classifieur SVM qui a une approche sac de mots et dont les scores de précision et de rappel pour ces trois classes sont supérieurs à 0,84. Ces classes sont aussi représentées dans la typologie baseline et maintiennent les meilleures moyennes F1 de toute la classification sous les étiquettes *title*⁸⁷ (0,93), *toward* (0,92) et *exit* (0,84) avec ce même classifieur (SVM linéaire avec SGD entraîné).

Pour ces trois mêmes classes, on observe que l'ajout d'un ou deux exemples dans le prompt pour les modèles génératifs améliore significativement leur capacité d'identification. À titre d'exemple le score F1 de *toward* dans la typologie principale passe de 0,1 (*gpt-3.5*) et 0,07 (*mixtral*) avec une stratégie *zero-shot* (post-traitement méthode « n'importe quel mot »), à 0,73 (*gpt-3.5*) et 0,83 (*mixtral*) avec une stratégie *two-shots*. Dès l'ajout d'un exemple de contextualisation ces scores augmentent drastiquement puisque le modèle d'OpenAI en *one-shot* obtient un score F1 de 0,73 et celui de MistralAI, 0,78 pour la classe *toward*.

⁸⁷ Équivalent de *music*, qui est aussi représentée dans la typologie baseline mais se trouve réunie sous l'étiquette *music* dans la typologie générique.

Quant aux erreurs de classification communes à tous les classifieurs, *aggression* est la classe la moins bien identifiée pour le SVM linéaire avec SGD entraîné un score F1 de 0,53. Cette classe est aussi parmi les moins bien identifiée pour les modèles BERT multilingues et SetFit et obtient un score F1 de 0,6950 dans le meilleur des cas (*bert-base-multilingual-uncased*). Pour *camembert-base* et *distilcamembert-base* *aggression* a un score F1 de 0,74 et 0,76 respectivement. Le rappel pour cette classe et pour ces modèles est de 0,79 et 0,83 respectivement, ce qui montre une solide capacité d'identification. Avec *gpt-3.5-turbo-0125* et *open-mixtral-8x7b*, cette classe est aussi assez mal identifiée, mais ne fait pas produit pas les pires classifications. En effet, avec le prompt two-shots Mixtral obtient un score F1 de 0,47 et GPT 0,32. On note que cette classe obtient un rappel de 100 par mixtral avec le prompt *zero shot*, mais une précision de 0,05. Il est possible que les mesures d'évaluation tirent vers les extrêmes parce que cette classe est la deuxième moins pourvue du jeu de données avec 70 didascalies dans le jeu de test. L'équivalent de la classe *aggression* dans la typologie baseline est, en partie, la classe *threat* (voir la *mapping 2.3.2*). *threat* est aussi assez mal classé avec un score F1 entre 0,2 (*mixtral*, *camembert-base*, *gpt*) et 0,5 (*bert-cased*, *SetFit*, *distilcamembert*).

Ce phénomène est encore plus marqué avec la classe *aparté*, qui contient seulement 14 exemples dans le jeu de test. Par conséquent, on observe que ce type peut être classé avec grande précision, comme c'est le cas avec tous les modèles BERT et SetFit où cette mesure atteint 100 dans plusieurs cas (*distilcamembert*, *bert-base-multilingual-cased* et *bert-base-multilingual-uncased*), mais que le rappel varie, passant de 0 (*camembert*) à 100 (*SetFit*). *Aperté* est par conséquent parmi les classes qui génèrent la plus grande variation dans les prédictions.

3.2.1.3. Analyse des confusions

À partir de l'analyse des matrices de confusion produites par les classifications des différents modèles, il est possible de dégager plusieurs groupes de confusion. Les matrices de confusion normalisées suivantes sont celles des prédictions générées par sept des modèles entraînés à savoir SVM linéaire (Figure 3), *camembert-base* (Figure 4), *distilcamembert* (), *bert-multilingual-base-uncased* (Figure 5), *bert-multilingual-base-cased* (Figure 7), *distilbert-base-multilingual-cased* (Figure 8), *SetFit* (Figure 9). L'ordre est proposé de telle sorte que les modèles puissent être mis en regard en fonction de leur ressemblance.

Pour ce qui est des modèles génératifs, il n'a pas été possible de générer les matrices de confusion, car le nombre de catégories prédites ne correspond jamais au nombre de catégories réelles. On peut en revanche s'appuyer sur les rapports de classification pour établir quelles classes ont été les mieux identifiées et à l'inverse, lesquelles ont été le moins souvent identifiées grâce aux mesures de précision et de rappel.

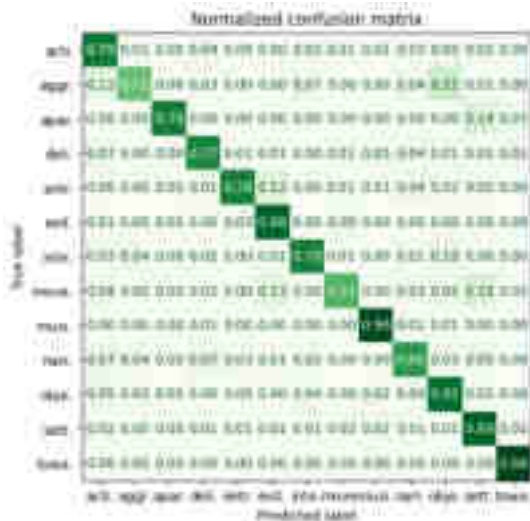


Figure 8 Matrice de confusion distilbert-multilingual-cased

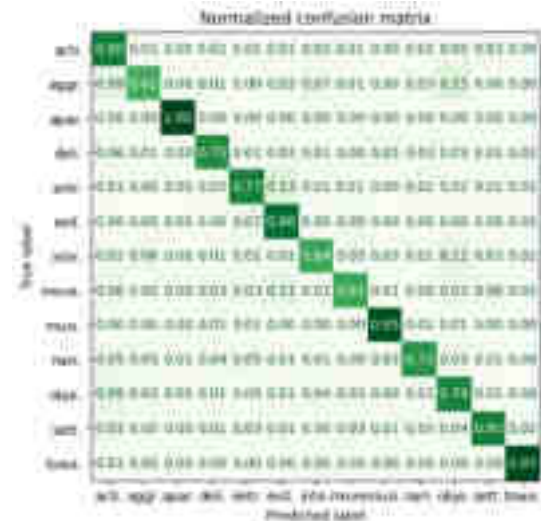


Figure 9 Matrice de confusion SetFit

Les groupes suivants font l'objet de confusion de la part des modèles entraînés sur nos données :

1. *aparte, toward, delivery*
2. *aggression, interaction, action*
3. *interaction, object*
4. *movement, exit, entrance*

Ces groupes ont tous des liens sémantiques qui expliquent au premier abord la difficulté de leur classification.

Le premier groupe, *aparté, toward* et *delivery*, est notamment marqué par le fait que ce sont les didascalies *aparté* qui sont classées en *delivery* et/ou *toward*. Cette confusion correspond à 50% des prédictions de la classe *aparté* par camembert-base (Figure 4) ; 29% pour distilcamembert-base (Figure 6 Matrice de confusion distilcamembert-base) ; 12% par le SVM (Figure 3). Cette confusion est moins marquée pour les modèles BERT multilingues où *aparté* est plus susceptible d'être confondu avec *setting* (14% des *apartés* classés en *setting* par distilbert-multilingual-base-cased (Figure 8) par exemple). Le deuxième groupe, *aggression, interaction* et *action*, fait l'objet de confusion dans le sens où d'une part *aggression* et *interaction* sont des classes plus compliquées à identifier et d'autre part *action* tend à être une classe par défaut pour certains classifieurs. Ainsi, *action* est attribué pour des didascalies de types très variés. Tous les modèles sont enclins à attribuer l'étiquette *action* à des didascalies de type *narration, setting* ou même *delivery*. Cette tendance est particulièrement marquée pour des types sémantiquement plus proches comme *aggression* et *interaction*. Ainsi, l'attribution d'une didascalie *aggression* à la classe *action* est un motif récurrent. 12% de la classe *aggression* est attribuée à *action* par distilbert-multilingual-base-cased (Figure 8) et bert-base-multilingual-cased (Figure 5), et 26% par le SVM avec SGD entraîné. Mais la confusion entre *aggression* et *interaction* est aussi un motif récurrent avec 12% de la classe *aggression* attribuée à *interaction* par les deux BERT multilingues non distillés (Figure 5, Figure 7). En revanche, les BERT français confondent moins ces deux classes (Figure 4, Figure 6).

Le troisième groupe, *interaction* et *object*, fait l'objet de confusions qui coïncident avec le groupe précédent. Les didascalies *aggression* classées en objet correspondent à 21% de cette classe dans les prédictions de *distilbert-base-multilingual-cased* (Figure 8), et 15% du *SetFit* (Figure 9). Ces confusions s'expliquent par le rapprochement sémantique de ces classes : l'identification d'objet lors d'une interaction ou d'une agression n'est probablement pas le critère relevé par les LLM.

Le quatrième groupe, *movement*, *exit* et *entrance*, est aussi un groupe très proche sémantiquement. Au-delà du contronyme du verbe « rentrer » (voir 2.1.3.1) spécifique au français, il n'est pas surprenant que les verbes de mouvement des didascalies de type *movement* soient associés aux déplacements hors plateau dans l'annotation de *FreDraCor* et que certaines de ces didascalies soient plus ambiguës qu'une simple sortie ou entrée de scène. Cette ambiguïté se retrouve par conséquent dans la classification automatique. On note tout de même que dans la majorité des cas, la confusion vient du fait qu'une didascalie de mouvement ou d'entrée est classée en *exit*. Ainsi la classification d'*entrance* et *movement* en *exit* par *distilbert-base-multilingual-cased* (Figure 8) correspond respectivement à 19% et 13% des prédictions totales de ces classes. Ces confusions sont similaires avec *camembert-base* (Figure 4). De même, les didascalies *movement* sont prédites dans 11% des cas comme *exit* par *SetFit* (Figure 9), et 14% des cas par *bert-base-multilingual-cased* (Figure 7).

3.2.2. Typologie baseline

Le Tableau 17 présente le nombre d'occurrences par type dans le jeu de test de la typologie baseline. On observe que certaines classes sont sous-représentées, ce qui impacte aussi la classification, comme mentionné dans la section 3.2.1.1. Dans cette typologie, ce sont les classes *SHOW* et *bis* qui, n'ayant qu'un seul exemple dans le jeu de test, sont le plus susceptibles de générer des scores extrêmes. Les classes *aparte*, *together* et *write* ont aussi moins de 10 exemples dans ce jeu. Les classes les mieux pourvues sont *title*, *toward*, *exit*, *decor* et *watch*, avec tous plus de cent exemples.

Catégorie	Occurrences dans le jeu de test
SHOW	1
aparte	7
away	40
bis	1
call	13
closer	53
cry	15
danse	35
decor	170
dress	50
drink	13
entrance	146
exit	237
explicit	64

fight	28
get	59
give	52
hide	17
hit	24
kiss	51
kneel	30
knock	20
laugh	19
location	42
loud	16
low	20
music	38
narration	68
noise	35
open	35

read	36
silence	13
sing	23
sit	27
stand	30
stop	13
threat	17

title	505
together	6
touch	42
toward	408
watch	143
wave	60
write	9

Tableau 17 Nombre d'occurrences par type dans le jeu de test de la typologie baseline

3.2.2.1. Classes les mieux identifiées et les moins bien identifiées

Au vu du nombre important de classes qui compose cette typologie, à savoir 44 classes, il n'est pas étonnant de trouver une grande disparité dans la classification indépendamment du classifieur ou de la stratégie d'apprentissage. Cependant, les disparités de représentativité des classes semblent moins impacter les capacités de généralisation des modèles.

On trouve des coïncidences dans les classes les mieux identifiées à travers toutes les expérimentations. *Write*, *toward* et *kiss* font toujours partie, à quelques exceptions près, des types les mieux classés, avec un score F1 dépassant souvent les 0,9. Mais, avec respectivement 9, 408 et 51 exemples pour ces classes dans le jeu de test, il n'y a pas de coïncidence explicite pour cette typologie entre le nombre d'exemples et la capacité des modèles à généraliser les caractéristiques de ses classes.

En revanche, le nombre d'occurrences très faible semble expliquer les scores F1 les plus bas. C'est le cas pour les classes *SHOW* et *bis*, représentées par un seul exemple chacune dans le jeu de test, qui obtiennent dans la plupart des classifications une précision et/ou un rappel de 0. Avec des scores très bas aussi dans la plupart des expérimentations, la classe *together* et ses six exemples pour l'évaluation, corrobore cette hypothèse.

On constate que certaines classes font l'objet d'une grande divergence dans la capacité des modèles à les identifier. Certains types qui semblent être difficiles à classer par l'algorithme linéaire SVM s'avèrent plus identifiables pour les modèles d'apprentissage par transfert et les modèles génératifs comme *silence*, *laugh*, *write* qui atteignent des scores F1 supérieurs à 0,9 pour les classes *laugh* et *write* en particulier.

Les algorithmes linéaires ainsi que les modèles d'apprentissage par transfert, parviennent tous à identifier les nouveaux exemples de didascalies de type *title*, mais les modèles génératifs ne disposant pas d'exemple lors du requêtage dans une stratégie *zero-shot*, n'attribuent jamais, ou presque jamais, cette étiquette à une didascalie. Ainsi *title* est la classe la mieux identifiée par l'algorithme linéaire SVM avec SGD entraîné, où elle obtient un score F1 de 0,93 avec ce modèle. De même, bert-base-multilingual-uncased, génère des prédictions sur cette classe atteignant un score F1 de 0,95. À l'inverse, les modèles mixtral et gpt 3.5 produisent des classifications pour la classe *title* qui obtiennent des scores F1 de 0,05 et 0 respectivement (méthode de post-traitement « n'importe quel mot »). Ce comportement peut s'expliquer par le fait qu'il n'est pas possible d'inférer le contenu d'une didascalie de type « *title* » sans davantage de contexte.

3.2.2.2. Analyse des confusions

Seules les matrices de confusion des classifieurs SVM et de distilcamembert-base sont reproduites ici. Le SVM est choisi parce que ses performances sont comparables à celles de l'algorithme linéaire ayant obtenu la meilleure moyenne-macro F1, à savoir le Ridge Classifier. Cela nous permet de conserver dans l'analyse le même modèle pour les deux typologies. Et distilcamembert est observé plus en détail parce que ses performances sont meilleures que tous les autres modèles pour cette typologie. Ce sont les seules matrices de confusion reproduites ici par souci de lisibilité. Cependant, toutes les matrices de confusion, normalisées ou non, sont disponibles dans le dépôt du projet ([algorithmes classiques](#), [apprentissage par transfert](#)). Les matrices normalisées se trouvent pour chaque modèle en suivant les liens correspondants : [SGDClassifier](#), [distilbert-base-multilingual-cased](#), [bert-base-multilingual-cased](#), [bert-base-multilingual-uncased](#), [camembert-base](#), [distilcamembert](#), [SetFit](#).

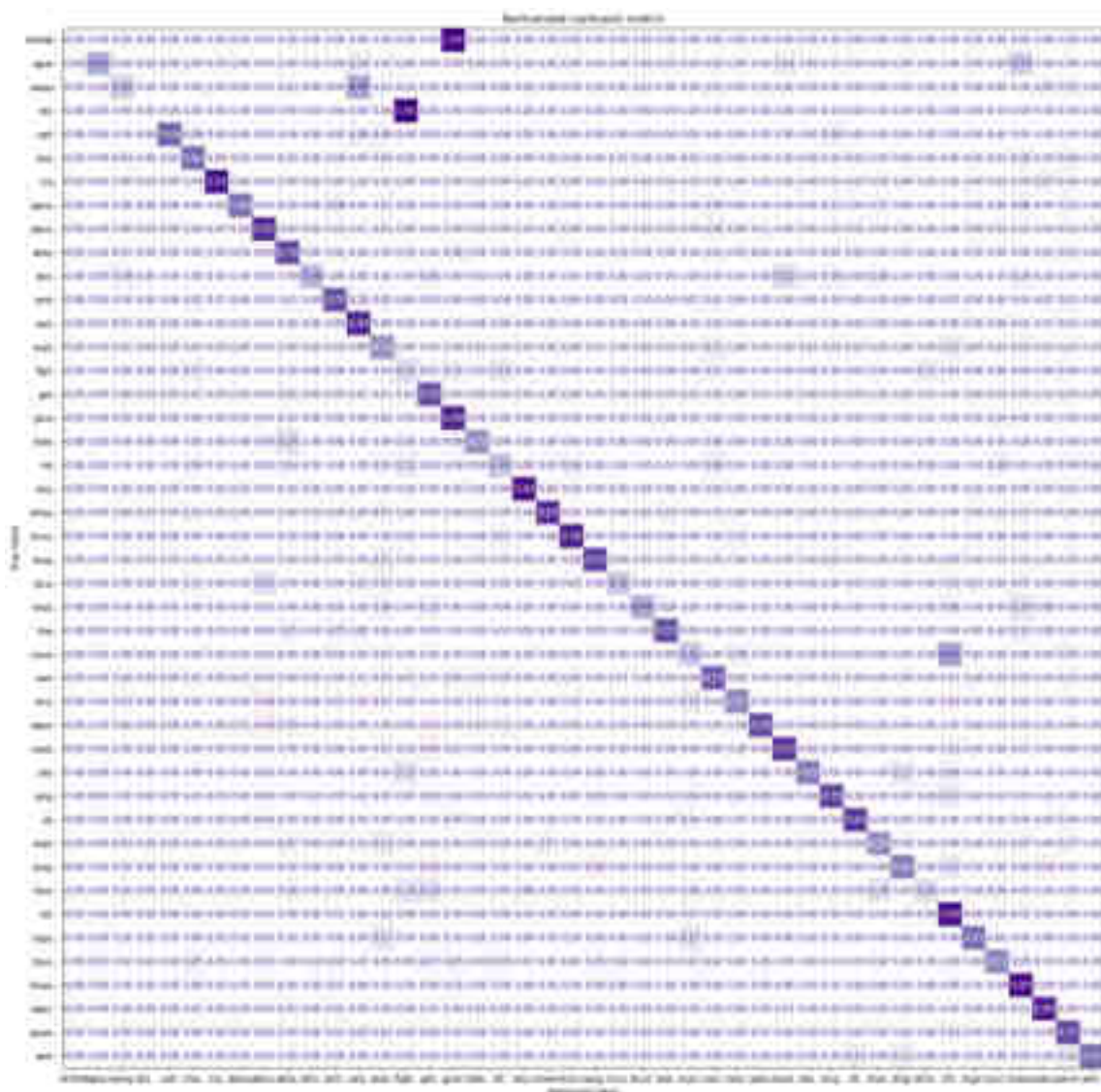


Figure 10 Matrice de confusion SVM linéaire (SGD entraîné) pour la typologie baseline

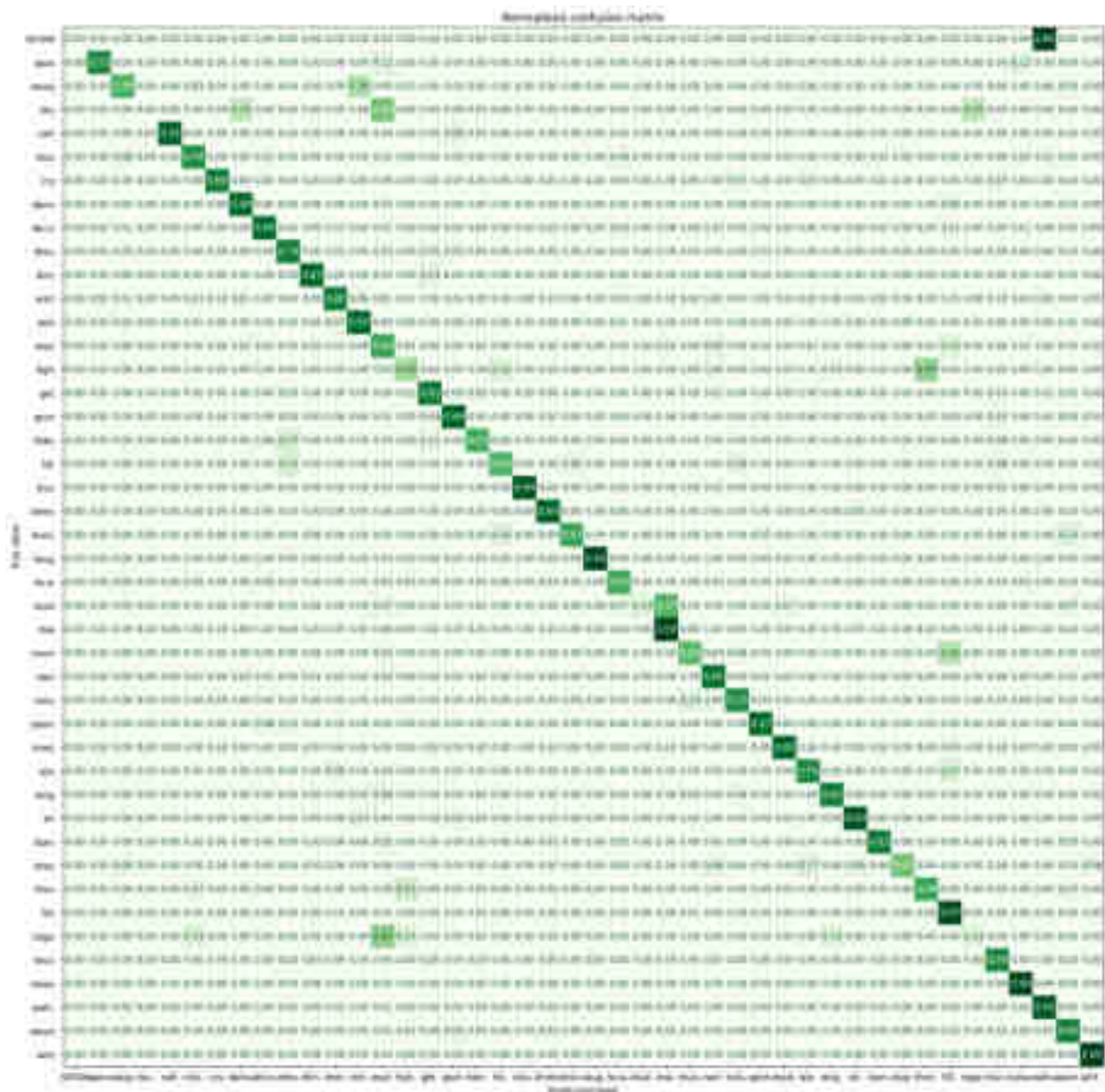


Figure 11 Matrice de confusion distilcamembert-base pour la typologie baseline

De manière générale, cette typologie fait émerger des groupes de similarité sémantique lors des classifications. Dans les matrices de confusion des deux modèles comparés en détail (Figure 10, Figure 11), il apparaît que *threat* est confondu avec *fight* dans 18% et 21% des cas ; *away* avec *exit* dans 45% et 30% des cas ; *together* avec *explicit* dans 17% et 43% des cas et enfin *music* avec *title* dans 47% et 30% des cas⁸⁸.

Ces groupes font apparaître la logique qui a permis le regroupement lors de la création du jeu de données de la typologie principale. En effet, *threat* et *fight* ont été placés dans *aggression*, *away* et *exit* réunis en *exit* et *music* et *title* réunis en *music*.

⁸⁸ Tous les pourcentages représentent la vraie classe attribuée à la fausse classe en donnant d'abord les prédictions du SVM linéaire puis de distilcamembert-base.

Dans le détail, si certaines confusions ne sont pas surprenantes, comme *distilcamembert-base* qui confond *stop* avec *silence* à 18%, parce qu'elles semblent explicables sur le plan sémantique, d'autres, comme *drink* associé à 23% à *read* par l'algorithme SVM, ne donnent pas à voir d'explication par la seule lecture des étiquettes.

Comme mentionné dans la section 3.1.3.2, la classification effectuée à partir d'un prompt sans exemple semble être influencée par l'intitulé du type. Les modèles génératifs tendent à chercher une correspondance entre le contenu strictement thématique de la didascalie et son type, ce qui laisse de côté certains types dont l'identification repose sur des critères non strictement sémantiques comme *narration* ou *explicit*, que l'on comprend comme étant des didascalies à forte valeur descriptive et/ou paratextuelle.

On se référera à la section 2.1.4.3 de la présentation de la typologie principale pour cerner les spécificités du type *narration* dans la typologie baseline. Le type *explicit* de la typologie baseline a des enjeux similaires, puisqu'il s'agit soit de didascalies descriptives comme dans l'exemple suivant :

1. Après les glorieuses fatigues et les exploits victorieux de notre auguste monarque, il est bien juste que tous ceux qui se mêlent d'écrire travaillent ou à ses louanges, ou à son divertissement. C'est ce qu'ici l'on a voulu faire, et ce prologue est un essai des louanges de ce grand prince, qui donne entrée à la comédie du Malade imaginaire, dont le projet a été fait pour le délasser de ses nobles travaux.

Mais il inclut aussi des didascalies bien plus brèves avec une valeur paratextuelle d'annonce du personnage énonciateur de la réplique comme dans l'exemple suivant :

2. « ENSEMBLE. »

On voit bien à travers ces deux exemples que le type *explicit* ne réfère pas à un contenu de nature « explicite », mais plutôt au fait qu'il s'agisse d'un commentaire en marge du texte ou d'une « explicitation » de la situation d'énonciation. Autrement dit, ces didascalies font appel à deux niveaux de compréhension du texte théâtral : le premier comprend des didascalies qui complètent la fiction, et en ce sens appartiennent, quoiqu'indirectement, à la diégèse, et le second, commente la fiction, et en ce sens ne relève pas de la diégèse. Il semblerait donc que les modèles génératifs aient rencontré des difficultés à cerner ces deux niveaux de lecture dans le cadre d'une tâche de classification.

3.2.3. Perspectives d'amélioration pour les modèles génératifs

Les expérimentations avec ces modèles ont été opérées à titre exploratoire, afin d'établir de quelle manière ces modèles, dont l'utilisation s'est largement répandue au cours des derniers mois, peuvent s'intégrer à une stratégie d'annotation. Force est de constater que la logique intrinsèque aux modèles génératifs nous oblige à changer de perspective pour la tâche de classification : il conviendrait peut-être de trouver un prompt qui permette d'intégrer un guide d'annotation, ou de fine-tuner un modèle génératif à partir de ce guide d'annotation. Pour des raisons de temps, aucun modèle génératif n'a été affiné.

3.2.3.1. Disparité entre les classes

Une des limites majeures de notre corpus, à savoir la disparité entre les classes représentées, pourrait être dépassée à l'aide de *data augmentation* telle que décrite par Guo & Chen (2024). La « data augmentation » à partir de prompt de génération conditionnelle consiste à générer du texte selon des attributs définis, dans notre cas une didascalie qui correspond à une classe donnée selon une série d'exemples ou d'instructions limités. Cette stratégie permettrait d'étoffer les classes les moins dotées de notre corpus tout en limitant le travail d'annotation manuelle.

3.2.3.2. Terminologie

La comparaison entre la typologie principale et la typologie baseline a révélé l'influence de l'intitulé même des classes lors de la classification par des modèles génératifs, avec ou sans l'ajout d'exemple(s) (voir 3.1.3.2 et 3.2.2.2). Dans la perspective de l'emploi de ce genre de modèle pour l'annotation automatique, il convient de prêter une attention particulière au choix terminologique lors de l'établissement de la typologie. Ce mémoire ébauche une comparaison de l'impact de ces intitulés à l'échelle de quelques types en coïncidence forte d'un jeu de données à un autre (*narration* et *location*), mais une étude complète et rigoureuse permettrait de mesurer l'influence réelle des intitulés de type sur la classification générée par des LLMs. Un sujet de recherche possible serait, à partir d'un même jeu de donnée, de comparer les classifications générées à partir de deux ensembles d'étiquettes, le premier ensemble étant une réelle typologie, et le second ayant des intitulés « neutre », sans rapport avec le contenu décrit comme « catégorie 1, catégorie 2 », etc.

Conclusion

En conclusion, ce projet a exploré l'annotation automatique du type de didascalie en français à partir d'une stratégie d'annotation simplifiée sur un corpus déjà annoté manuellement.

La première partie de ce mémoire, concernant l'état de l'art, comprend deux sous-parties. La première s'est concentrée sur la définition de la didascalie dans le contexte des Humanités Numériques et a cherché à extraire de quelques typologies issues de la théorie littéraire les éléments utiles à l'élaboration d'une typologie pertinente pour les HN actuelles. La deuxième s'est concentrée sur les pratiques d'annotation, avec une réflexion sur l'approche méthodologique des pratiques d'annotation et des pratiques de classification automatique ainsi que sur les outils nécessaires à la phase expérimentale du projet de mémoire.

La deuxième partie de ce mémoire, parcourant la méthodologie du projet, a présenté le guide d'annotation d'une typologie générique qui a ensuite été appliquée à partir de l'annotation préexistante de FreDraCor aux quelques 38 000 didascalies de ce corpus. Un deuxième jeu de données a été élaboré à des fins de comparaison en tant qu'alternative simplifiée à la conception d'un jeu de données complet à partir de ce même corpus, FreDraCor, sur la base d'une extraction des types représentant au moins cent didascalies dans ce corpus. La typologie qui en a découlé fait état de 44 types, contrairement à la typologie principale qui a réuni en 13 types les didascalies du corpus représentant au moins 50 didascalies. Cette partie présente aussi les outils de TAL nécessaires au projet ainsi que les paramétrages des classifieurs automatiques comparés, à savoir quatre algorithmes de ML classique, des modèles d'apprentissage par transfert basés sur BERT, un modèle d'apprentissage contrastif et enfin trois techniques d'apprentissage en contexte (*in-context learning*) à savoir le *zero-shot*, *one-shot* et *two-shots* pour deux modèles génératifs.

La troisième partie, faisant état des résultats, s'est concentrée sur l'évaluation qualitative et quantitative des classifications produites par ces différents modèles et a mis au jour les capacités et les limites des modèles dans une analyse globale de leur performance, mais a aussi permis de comprendre ce qui, dans les typologies proposées, limite les modèles dans leur prédiction. Au terme de cette analyse, il apparaît que les modèles d'apprentissage par transfert affinés sur une typologie générique sont les plus adaptés à la tâche de classification supervisée pour les données hétérogènes que sont des didascalies en langue française.

Références bibliographiques

- Albert Jiang, Alexandre Sabayrolles, & Arthur Mensch. (2023, septembre 27). *Mistral 7B*.
<https://mistral.ai/news/announcing-mistral-7b/>
- Andresen, M., Krautter, B., Pagel, J., & Reiter, N. (2022). Who Knows What in German Drama? A Composite Annotation Scheme for Knowledge Transfer. Annotation, Evaluation, and Analysis. *Journal of Computational Literary Studies*, 1(1), Article 1.
<https://doi.org/10.48694/jcls.107>
- Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). *Enriching Word Vectors with Subword Information* (arXiv:1607.04606). arXiv.
<https://doi.org/10.48550/arXiv.1607.04606>
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). *Language Models are Few-Shot Learners* (arXiv:2005.14165). arXiv.
<https://doi.org/10.48550/arXiv.2005.14165>
- Carlson, M. (1991). The Status of Stage Directions. *Studies in the Literary Imagination*, 24(2), 37-48.
- Daria Maximova & Frank Fischer. (2019). Using Machine Learning for the Automated Classification of Stage Directions in TEI-Encoded Drama Corpus. *Book of Abstracts*, 123. <https://graz-2019.tei-c.org/files/BoATEI2019.pdf#page=123>
- Delestre, C., & Amar, A. (2022, juillet). *DistilCamemBERT: une distillation du modèle français CamemBERT*. CAp (Conférence sur l'Apprentissage automatique), Vannes, France.
- Dennerlein, Dr. K. (2015, mars). *Configuration density as a measurement for differences between comedies and tragedies*. Workshop Computer-based analysis of drama and

its uses for literary criticism and historiography, Munich.

<https://comedy.hypotheses.org/12>

Dennerlein, Dr. K. (2016, juin 26). *Digitally assisted analysis of drama | History of Comedy*.

<https://comedy.hypotheses.org/category/digitally-assisted-analysis-of-drama>

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171-4186. <https://doi.org/10.18653/v1/N19-1423>

Dompeyre, S. (1992). Étude des fonctions et du fonctionnement des didascalies. *Pratiques*, 74(1), Article 1. <https://doi.org/10.3406/prati.1992.1665>

Douguet, M. (2015). *La composition dramatique* [Theses, Université Paris 8].

<https://hal.archives-ouvertes.fr/tel-01504340>

Douguet, M. (2017). L'extraction automatique des motifs dramaturgiques dans les séquences de deux scènes. *Digital Humanities 2017: Conference Abstracts*. Digital Humanities, Montréal, Canada. dh2017.adho.org

Douguet, Marc. (2017). MAnaDram : Un moteur d'analyse pour l'extraction et le repérage des motifs dramaturgiques. *Études théâtrales et humanités numériques | Revue d'Historiographie du Théâtre No4*.

<https://sht.asso.fr/revue/etudes-theatrales-et-humanites-numeriques/>

Escobar Varela, M. (2021). Words as Data. In *Theater as Data* (p. 75-93). University of Michigan Press. <https://www.jstor.org/stable/10.3998/mpub.11667458.8>

Fina Polat, Paul Groth, & Ilaria Tiddi. (2023). *Testing Prompt Engineering Methods for Knowledge Extraction from Text* (3606-4820). <https://github.com/FinaPolat/Prompt-Engineering-for-KE>. <https://www.semantic-web-journal.net/system/files/swj3606.pdf>

- Fischer, F., Börner, I., Göbel, M., Hechtel, A., Kittel, C., Milling, C., & Trilcke, P. (2019). Programmable Corpus : Introducing DraCor, an Infrastructure for the Research on European Drama. *Proceedings of DH2019: « Complexities »*,. DH2019, Utrecht University. <https://doi.org/doi:10.5281/zenodo.4284002>.
- Fischer, F., Kittel Christopher, Dazord Gilles, Göbel Mathias, & Trilcke Peer. (2017). Le drame comme réseau de relations. Une application de l'analyse automatisée pour l'histoire littéraire du théâtre. *Études théâtrales et humanités numériques | Revue d'Historiographie du Théâtre No4*. <https://sht.asso.fr/revue/etudes-theatrales-et-humanites-numeriques/>
- Flaux, N., & Zaragoza, G. (2002). L'inversion du sujet dans les didascalies. *L'information grammaticale*, 95(1), 31-37. <https://doi.org/10.3406/igram.2002.2655>
- François Chollet. (2017). *Deep Learning with Python*. <https://www.manning.com/books/deep-learning-with-python>
- Galleron, I. (2018). *Quel encodage pour les didascalies internes ?* JE « Corpus et textes de représentation », Avignon.
- Galleron, I. (2021). Pour un balisage sémantique des textes de théâtre : Le cas des didascalies. *Sens public*, 1-23. <https://doi.org/10.7202/1089589ar>
- Glorieux, F. (2017). Les confidents du théâtre classique Essai d'enquête quantitative. *Études théâtrales et humanités numériques | Revue d'Historiographie du Théâtre No4*. <https://sht.asso.fr/revue/etudes-theatrales-et-humanites-numeriques/>
- Guo, X., & Chen, Y. (2024). *Generative AI for Synthetic Data Generation : Methods, Challenges and the Future* (arXiv:2403.04190). arXiv. <http://arxiv.org/abs/2403.04190>
- Hastie, T., Friedman, J., & Tibshirani, R. (2009). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction* (Springer).

- Ingarden, R. (1973). *The Literary Work of Art : An Investigation on the Borderlines of Ontology, Logic, and Theory of Literature : with an Appendix on the Functions of Language in the Theater*. Northwestern University Press.
- Issacharoff, M. (1981). Texte théâtral et didascalecture. *MLN*, 96(4), 809-823.
<https://doi.org/10.2307/2905838>
- Jockers, M. L. (2013). *Macroanalysis : Digital Methods and Literary History*. University of Illinois Press.
- Krautter, B., Pagel, J., Reiter, N., & Willand, M. (2022). Properties of Dramatic Characters : Automatically Detecting Gender, Age, and Social Status. In *Properties of Dramatic Characters : Automatically Detecting Gender, Age, and Social Status* (p. 179-202). De Gruyter. <https://doi.org/10.1515/9783110781502-010>
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). *Pre-train, Prompt, and Predict : A Systematic Survey of Prompting Methods in Natural Language Processing* (arXiv:2107.13586). arXiv. <https://doi.org/10.48550/arXiv.2107.13586>
- Manako, Ô. (2006). LE ROMAN DU THÉÂTRE : La dernière bande et le « reste » didascalique. *Samuel Beckett Today / Aujourd'hui*, 17, 351-364.
- Marcus, S. (1970). *Poetica matematica*. Editura Academiei Republicii Socialiste România.
- Martin, L., Muller, B., Suárez, P. J. O., Dupont, Y., Romary, L., de la Clergerie, É. V., Seddah, D., & Sagot, B. (2020). CamemBERT : A Tasty French Language Model. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7203-7219. <https://doi.org/10.18653/v1/2020.acl-main.645>
- Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2024). Prompt Engineering in Large Language Models. In I. J. Jacob, S. Piramuthu, & P. Falkowski-Gilski (Éds.), *Data Intelligence and Cognitive Informatics* (p. 387-402). Springer Nature.
https://doi.org/10.1007/978-981-99-7962-2_30

- Moretti, F. (2000). Conjectures on World Literature. *New Left Review*, 1, 54-68.
- Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python : A Guide for Data Scientists*. O'Reilly Media, Inc.
- Munkelt, M. (1987). *Stage Directions as Part of the Text*. 19, 253-272.
- Ortiz Suárez, P. J., Sagot, B., & Romary, L. (2019). *Asynchronous pipelines for processing huge corpus on medium to low resource infrastructures*. 9-16.
<https://doi.org/10.14618/ids-pub-9021>
- Pagel, J., Sihag, N., & Reiter, N. (2021, novembre). Predicting Structural Elements in German Drama. *Proceedings of the Second Conference on Computational Humanities Research (CHR2021)*. (CHR2021), online.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn : Machine Learning in Python. *Journal of Machine Learning Research*, 12(85), 2825-2830.
- Pennington, J., Socher, R., & Manning, C. (2014). Glove : Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543. <https://doi.org/10.3115/v1/D14-1162>
- Pfister, M. (1988). *The Theory and Analysis of Drama*. Cambridge University Press.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer* (arXiv:1910.10683; Version 1). arXiv.
<https://doi.org/10.48550/arXiv.1910.10683>
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT : Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in*

Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 3980-3990.

<https://doi.org/10.18653/v1/D19-1410>

Ruiz Fabo, P., Bernhard, D., Briand, A., & Werner, C. (2024). Computational drama analysis from almost zero electronic text : The case of Alsatian theater. *Computational Drama Analysis: Reflecting Methods and Interpretations*.

<https://univoak.eu/islandora/object/islandora%3A157880/>

Ruiz Fabo, P., Boisnard, F., & Schneider, A. (2024, août). *Comparative metadata for popular theater in Alsatian, German and French*. ADHO - Alliance of Digital Humanities Organizations. <https://hal.science/hal-04520875>

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). *DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter* (arXiv:1910.01108). arXiv.

<https://doi.org/10.48550/arXiv.1910.01108>

Schmidt, T., Burghardt, M., Dennerlein, K., & Wolff, C. (2019). Katharsis – A Tool for Computational Drametrics. *Book of Abstracts, Digital Humanities Conference 2019 (DH 2019)*, 1-5.

Schneider, A. (2023, novembre 29). *Classification automatique du type de didascalie*.

<https://doi.org/10.5281/zenodo.10644192>

Schneider, A., & Nugues, L. (2023, novembre 29). *Détection automatiquement des airs de vaudeville*. <https://doi.org/10.5281/zenodo.10702988>

Schneider, A., & Ruiz, P. (2024). Stage Direction Classification in French Theater : Transfer Learning Experiments. *Proceedings of LaTeCH-CLfL 2024*, 280.

The pandas development team. (2024). *pandas-dev/pandas : Pandas (v2.2.2)* [Logiciel].

Zenodo. <https://doi.org/10.5281/zenodo.10957263>

Tunstall, L., Reimers, N., Jo, U. E. S., Bates, L., Korat, D., Wasserblat, M., & Pereg, O.

(2022). *Efficient Few-Shot Learning Without Prompts*. <http://arxiv.org/abs/2209.11055>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need* (arXiv:1706.03762; Version 1). arXiv. <https://doi.org/10.48550/arXiv.1706.03762>

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). *ReAct : Synergizing Reasoning and Acting in Language Models*. <https://doi.org/10.48550/arXiv.2210.03629>

Yvon, F. (2023, novembre 19). *Building and Evaluating Massively Multilingual Language Models*. Nancy, journées du GDR LIFT, november 19th, 2023, Nancy.

Annexes

A. Rapports de classification

A.1. SVM linéaire (SGD entraîné)

Typologie principale :

	precision	recall	f1-score	support
action	0.7656	0.8164	0.7902	512
aggression	0.5645	0.5000	0.5303	70
aparte	0.9091	0.6250	0.7407	16
delivery	0.7120	0.7158	0.7139	190
entrance	0.7822	0.6172	0.6900	128
exit	0.8431	0.8750	0.8587	264
interaction	0.7474	0.6455	0.6927	110
movement	0.6778	0.5126	0.5837	119
music	0.9068	0.9364	0.9214	582
narration	0.6250	0.6796	0.6512	103
object	0.6622	0.7129	0.6866	209
setting	0.7744	0.7135	0.7427	178
toward	0.9246	0.9434	0.9339	442
accuracy			0.8036	2923
macro avg	0.7611	0.7149	0.7335	2923
weighted avg	0.8019	0.8036	0.8011	2923

Typologie baseline :

	precision	recall	f1-score	support
SHOW	0.0000	0.0000	0.0000	1
aparte	0.6000	0.4286	0.5000	7
away	0.5455	0.3000	0.3871	40
bis	0.0000	0.0000	0.0000	1
call	0.9000	0.6923	0.7826	13
closer	0.6481	0.6604	0.6542	53
cry	0.9333	0.9333	0.9333	15
danse	0.9200	0.6571	0.7667	35
decor	0.8675	0.8471	0.8571	170
dress	0.6333	0.7600	0.6909	50
drink	1.0000	0.3846	0.5556	13
entrance	0.7432	0.7534	0.7483	146
exit	0.8157	0.8776	0.8455	237
explicit	0.4198	0.5312	0.4690	64
fight	0.2857	0.2143	0.2449	28
get	0.6875	0.7458	0.7154	59
give	0.7077	0.8846	0.7863	52
hide	0.6923	0.5294	0.6000	17
hit	0.4211	0.3333	0.3721	24
kiss	0.9600	0.9412	0.9505	51
kneel	0.8333	0.8333	0.8333	30
knock	0.8182	0.9000	0.8571	20
laugh	0.8824	0.7895	0.8333	19
location	0.6842	0.3095	0.4262	42

loud	0.8750	0.4375	0.5833	16
low	0.8824	0.7500	0.8108	20
music	0.4000	0.3158	0.3529	38
narration	0.5765	0.7206	0.6405	68
noise	0.7407	0.5714	0.6452	35
open	0.8387	0.7429	0.7879	35
read	0.8108	0.8333	0.8219	36
silence	0.8889	0.6154	0.7273	13
sing	0.6207	0.7826	0.6923	23
sit	0.8148	0.8148	0.8148	27
stand	0.6400	0.5333	0.5818	30
stop	0.7273	0.6154	0.6667	13
threat	0.3636	0.2353	0.2857	17
title	0.9141	0.9485	0.9310	505
together	0.6667	0.6667	0.6667	6
touch	0.7059	0.5714	0.6316	42
toward	0.9068	0.9534	0.9295	408
watch	0.8958	0.9021	0.8990	143
wave	0.6216	0.7667	0.6866	60
write	0.8571	0.6667	0.7500	9
accuracy			0.7979	2731
macro avg	0.6988	0.6307	0.6526	2731
weighted avg	0.7959	0.7979	0.7918	2731

A.2. camembert-base

Typologie principale :

	precision	recall	f1-score	support
action	0.9366	0.8210	0.8750	486
aggression	0.7024	0.7867	0.7421	75
aparte	0.0000	0.0000	0.0000	14
delivery	0.8612	0.8451	0.8531	213
entrance	0.8095	0.7969	0.8031	128
exit	0.8549	0.9008	0.8773	242
interaction	0.7544	0.8431	0.7963	102
movement	0.6639	0.6639	0.6639	119
music	0.9775	0.9775	0.9775	577
narration	0.7934	0.8000	0.7967	120
object	0.8045	0.8510	0.8271	208
setting	0.8309	0.9053	0.8665	190
toward	0.9591	0.9933	0.9759	449
accuracy			0.8820	2923
macro avg	0.7653	0.7834	0.7734	2923
weighted avg	0.8802	0.8820	0.8801	2923

Typologie Baseline :

Classification report:

precision	recall	f1-score	support
-----------	--------	----------	---------

SHOW	0.0000	0.0000	0.0000	1
aparte	0.0000	0.0000	0.0000	8
away	0.4737	0.4865	0.4800	37
bis	0.0000	0.0000	0.0000	4
call	1.0000	0.9091	0.9524	11
closer	0.7755	0.7170	0.7451	53
cry	0.8000	0.8000	0.8000	15
danse	0.6923	0.8438	0.7606	32
decor	0.9172	0.8986	0.9078	148
dress	0.6957	0.8205	0.7529	39
drink	0.8333	0.6667	0.7407	15
entrance	0.8255	0.8311	0.8283	148
exit	0.8805	0.8633	0.8718	256
explicit	0.6703	0.6854	0.6778	89
fight	0.4583	0.4400	0.4490	25
get	0.7778	0.8448	0.8099	58
give	0.8933	0.8816	0.8874	76
hide	1.0000	0.6111	0.7586	18
hit	0.4211	0.6667	0.5161	12
kiss	0.9362	0.9362	0.9362	47
kneel	0.9048	1.0000	0.9500	19
knock	0.7143	0.7143	0.7143	14
laugh	0.9474	1.0000	0.9730	18
location	0.6333	0.5000	0.5588	38
loud	0.0000	0.0000	0.0000	14
low	0.3500	0.9333	0.5091	15
music	0.6296	0.5152	0.5667	33
narration	0.7683	0.8077	0.7875	78
noise	0.8462	0.7857	0.8148	28
open	0.8485	0.9032	0.8750	31
read	0.9355	0.8286	0.8788	35
silence	0.6000	0.7500	0.6667	12
sing	0.9091	0.7692	0.8333	26
sit	0.9333	0.9333	0.9333	30
stand	0.9048	0.9048	0.9048	21
stop	1.0000	0.5833	0.7368	12
threat	0.3333	0.2143	0.2609	14
title	0.9435	0.9691	0.9561	551
together	0.0000	0.0000	0.0000	7
touch	0.7000	0.7568	0.7273	37
toward	0.9791	0.9836	0.9814	428
watch	0.9224	0.9554	0.9386	112
wave	0.7778	0.8305	0.8033	59
write	1.0000	0.7143	0.8333	7
accuracy			0.8583	2731
macro avg	0.6962	0.6876	0.6836	2731
weighted avg	0.8533	0.8583	0.8535	2731

A.3. distilcamembert-base

Typologie principale :

	precision	recall	f1-score	support
action	0.9047	0.8395	0.8709	486
aggression	0.7045	0.8267	0.7607	75
aparte	1.0000	0.2857	0.4444	14
delivery	0.8641	0.8357	0.8496	213
entrance	0.8532	0.7266	0.7848	128
exit	0.8139	0.9215	0.8643	242
interaction	0.7719	0.8627	0.8148	102
movement	0.7222	0.6555	0.6872	119
music	0.9637	0.9671	0.9654	577
narration	0.7463	0.8333	0.7874	120
object	0.8271	0.8510	0.8389	208
setting	0.8490	0.8579	0.8534	190
toward	0.9800	0.9822	0.9811	449
accuracy			0.8803	2923
macro avg	0.8462	0.8035	0.8079	2923
weighted avg	0.8828	0.8803	0.8795	2923

Typologie baseline :

Classification report:

	precision	recall	f1-score	support
SHOW	0.0000	0.0000	0.0000	1
aparte	1.0000	0.7500	0.8571	8
away	0.5238	0.5946	0.5570	37
bis	0.0000	0.0000	0.0000	4
call	1.0000	0.9091	0.9524	11
closer	0.7500	0.7358	0.7429	53
cry	0.9231	0.8000	0.8571	15
danse	0.8235	0.8750	0.8485	32
decor	0.9220	0.8784	0.8997	148
dress	0.7381	0.7949	0.7654	39
drink	0.8667	0.8667	0.8667	15
entrance	0.8027	0.7973	0.8000	148
exit	0.9102	0.8711	0.8902	256
explicit	0.6404	0.6404	0.6404	89
fight	0.4737	0.3600	0.4091	25
get	0.7705	0.8103	0.7899	58
give	0.9067	0.8947	0.9007	76
hide	0.7333	0.6111	0.6667	18
hit	0.5385	0.5833	0.5600	12
kiss	1.0000	0.9362	0.9670	47
kneel	0.8095	0.8947	0.8500	19
knock	0.7500	0.6429	0.6923	14
laugh	0.9474	1.0000	0.9730	18
location	0.5455	0.6316	0.5854	38
loud	1.0000	0.1429	0.2500	14

low	0.4839	1.0000	0.6522	15
music	0.6923	0.5455	0.6102	33
narration	0.7500	0.8462	0.7952	78
noise	0.8077	0.7500	0.7778	28
open	0.9000	0.8710	0.8852	31
read	0.9688	0.8857	0.9254	35
silence	0.6923	0.7500	0.7200	12
sing	0.8077	0.8077	0.8077	26
sit	0.9333	0.9333	0.9333	30
stand	0.9444	0.8095	0.8718	21
stop	0.8571	0.5000	0.6316	12
threat	0.4737	0.6429	0.5455	14
title	0.9383	0.9655	0.9517	551
together	0.3333	0.1429	0.2000	7
touch	0.7000	0.7568	0.7273	37
toward	0.9791	0.9836	0.9814	428
watch	0.9375	0.9375	0.9375	112
wave	0.7344	0.7966	0.7642	59
write	0.8750	1.0000	0.9333	7
accuracy			0.8620	2731
macro avg	0.7542	0.7260	0.7266	2731
weighted avg	0.8645	0.8620	0.8599	2731

A.4. bert-base-multilingual-uncased

Typologie principale :

	precision	recall	f1-score	support
action	0.8621	0.8107	0.8356	486
aggression	0.7424	0.6533	0.6950	75
aparte	1.0000	0.9286	0.9630	14
delivery	0.8587	0.7418	0.7960	213
entrance	0.8110	0.8047	0.8078	128
exit	0.8583	0.9008	0.8790	242
interaction	0.6250	0.8333	0.7143	102
movement	0.7048	0.6218	0.6607	119
music	0.9672	0.9723	0.9697	577
narration	0.6304	0.7250	0.6744	120
object	0.7773	0.7885	0.7828	208
setting	0.7843	0.8421	0.8122	190
toward	0.9799	0.9777	0.9788	449
accuracy			0.8570	2923
macro avg	0.8155	0.8154	0.8130	2923
weighted avg	0.8605	0.8570	0.8573	2923

Typologie baseline :

	precision	recall	f1-score	support
SHOW	0.0000	0.0000	0.0000	1
aparte	1.0000	0.7500	0.8571	8

away	0.4694	0.6216	0.5349	37
bis	0.0000	0.0000	0.0000	4
call	0.8182	0.8182	0.8182	11
closer	0.6333	0.7170	0.6726	53
cry	0.6842	0.8667	0.7647	15
danse	0.7179	0.8750	0.7887	32
decor	0.9394	0.8378	0.8857	148
dress	0.5366	0.5641	0.5500	39
drink	0.8333	0.6667	0.7407	15
entrance	0.8276	0.8108	0.8191	148
exit	0.9121	0.8516	0.8808	256
explicit	0.7093	0.6854	0.6971	89
fight	0.6875	0.4400	0.5366	25
get	0.7000	0.8448	0.7656	58
give	0.9429	0.8684	0.9041	76
hide	0.6000	0.5000	0.5455	18
hit	0.4615	0.5000	0.4800	12
kiss	0.9773	0.9149	0.9451	47
kneel	0.8636	1.0000	0.9268	19
knock	0.5789	0.7857	0.6667	14
laugh	0.8947	0.9444	0.9189	18
location	0.4773	0.5526	0.5122	38
loud	0.6250	0.3571	0.4545	14
low	0.5833	0.9333	0.7179	15
music	0.6818	0.4545	0.5455	33
narration	0.8571	0.6923	0.7660	78
noise	0.7778	0.7500	0.7636	28
open	0.7500	0.7742	0.7619	31
read	0.9310	0.7714	0.8438	35
silence	0.5455	0.5000	0.5217	12
sing	0.7692	0.7692	0.7692	26
sit	0.6923	0.9000	0.7826	30
stand	0.9412	0.7619	0.8421	21
stop	0.8750	0.5833	0.7000	12
threat	0.5000	0.4286	0.4615	14
title	0.9402	0.9710	0.9554	551
together	0.5000	0.1429	0.2222	7
touch	0.6410	0.6757	0.6579	37
toward	0.9722	0.9813	0.9767	428
watch	0.9130	0.9375	0.9251	112
wave	0.6026	0.7966	0.6861	59
write	1.0000	1.0000	1.0000	7
accuracy			0.8444	2731
macro avg	0.7128	0.6954	0.6947	2731
weighted avg	0.8490	0.8444	0.8433	2731

A.5. bert-base-multilingual-cased

Typologie principale :

precision recall f1-score support

action	0.8881	0.8169	0.8510	486
aggression	0.6429	0.7200	0.6792	75
aparte	1.0000	0.7143	0.8333	14
delivery	0.8199	0.8122	0.8160	213
entrance	0.7612	0.7969	0.7786	128
exit	0.8555	0.9050	0.8795	242
interaction	0.6377	0.8627	0.7333	102
movement	0.6260	0.6891	0.6560	119
music	0.9787	0.9549	0.9667	577
narration	0.7838	0.7250	0.7532	120
object	0.8265	0.7788	0.8020	208
setting	0.8586	0.8632	0.8609	190
toward	0.9800	0.9844	0.9822	449
accuracy			0.8659	2923
macro avg	0.8199	0.8172	0.8148	2923
weighted avg	0.8711	0.8659	0.8672	2923

Typologie baseline :

	precision	recall	f1-score	support
SHOW	0.0000	0.0000	0.0000	1
aparte	0.6250	0.6250	0.6250	8
away	0.5000	0.5135	0.5067	37
bis	0.0000	0.0000	0.0000	4
call	1.0000	0.7273	0.8421	11
closer	0.6667	0.6415	0.6538	53
cry	0.7647	0.8667	0.8125	15
danse	0.8000	0.8750	0.8358	32
decor	0.9328	0.8446	0.8865	148
dress	0.5870	0.6923	0.6353	39
drink	0.8333	0.6667	0.7407	15
entrance	0.7879	0.7027	0.7429	148
exit	0.8506	0.8672	0.8588	256
explicit	0.6667	0.5843	0.6228	89
fight	0.4286	0.2400	0.3077	25
get	0.8113	0.7414	0.7748	58
give	0.8800	0.8684	0.8742	76
hide	0.5500	0.6111	0.5789	18
hit	0.2917	0.5833	0.3889	12
kiss	0.9565	0.9362	0.9462	47
kneel	0.8500	0.8947	0.8718	19
knock	0.7143	0.7143	0.7143	14
laugh	0.9000	1.0000	0.9474	18
location	0.4773	0.5526	0.5122	38
loud	0.4545	0.3571	0.4000	14
low	0.5556	1.0000	0.7143	15
music	0.6364	0.4242	0.5091	33
narration	0.7234	0.8718	0.7907	78
noise	0.7188	0.8214	0.7667	28
open	0.7879	0.8387	0.8125	31
read	0.8485	0.8000	0.8235	35
silence	0.5455	0.5000	0.5217	12
sing	0.8182	0.6923	0.7500	26
sit	0.8750	0.9333	0.9032	30
stand	0.8947	0.8095	0.8500	21
stop	0.6364	0.5833	0.6087	12

threat	0.5294	0.6429	0.5806	14
title	0.9371	0.9728	0.9546	551
together	1.0000	0.1429	0.2500	7
touch	0.7059	0.6486	0.6761	37
toward	0.9766	0.9743	0.9754	428
watch	0.9231	0.9643	0.9432	112
wave	0.8103	0.7966	0.8034	59
write	0.8750	1.0000	0.9333	7
accuracy			0.8400	2731
macro avg	0.7074	0.6937	0.6874	2731
weighted avg	0.8409	0.8400	0.8374	2731

A.6. distilbert-base-uncased

Typologie principale :

	precision	recall	f1-score	support
action	0.8691	0.7922	0.8288	486
aggression	0.6842	0.5200	0.5909	75
aparte	0.8462	0.7857	0.8148	14
delivery	0.7874	0.7653	0.7762	213
entrance	0.8475	0.7812	0.8130	128
exit	0.8492	0.8843	0.8664	242
interaction	0.7339	0.7843	0.7583	102
movement	0.7234	0.5714	0.6385	119
music	0.9504	0.9636	0.9570	577
narration	0.6260	0.6833	0.6534	120
object	0.7154	0.8462	0.7753	208
setting	0.7824	0.8895	0.8325	190
toward	0.9779	0.9844	0.9811	449
accuracy			0.8502	2923
macro avg	0.7995	0.7886	0.7913	2923
weighted avg	0.8513	0.8502	0.8490	2923

Typologie baseline :

	precision	recall	f1-score	support
SHOW	0.0000	0.0000	0.0000	1
aparte	1.0000	0.6250	0.7692	8
away	0.5135	0.5135	0.5135	37
bis	0.0000	0.0000	0.0000	4
call	0.9000	0.8182	0.8571	11
closer	0.7200	0.6792	0.6990	53
cry	0.8667	0.8667	0.8667	15
danse	0.8667	0.8125	0.8387	32
decor	0.9179	0.8311	0.8723	148
dress	0.6136	0.6923	0.6506	39
drink	0.8462	0.7333	0.7857	15
entrance	0.7800	0.7905	0.7852	148
exit	0.8835	0.8594	0.8713	256
explicit	0.6506	0.6067	0.6279	89

fight	0.7059	0.4800	0.5714	25
get	0.6769	0.7586	0.7154	58
give	0.9254	0.8158	0.8671	76
hide	0.7778	0.3889	0.5185	18
hit	0.5000	0.6667	0.5714	12
kiss	0.9565	0.9362	0.9462	47
kneel	0.8636	1.0000	0.9268	19
knock	0.7778	0.5000	0.6087	14
laugh	0.9444	0.9444	0.9444	18
location	0.4651	0.5263	0.4938	38
loud	0.4286	0.2143	0.2857	14
low	0.5652	0.8667	0.6842	15
music	0.5833	0.4242	0.4912	33
narration	0.7442	0.8205	0.7805	78
noise	0.5952	0.8929	0.7143	28
open	0.8710	0.8710	0.8710	31
read	0.9118	0.8857	0.8986	35
silence	0.6667	0.6667	0.6667	12
sing	0.7600	0.7308	0.7451	26
sit	0.8750	0.9333	0.9032	30
stand	0.6400	0.7619	0.6957	21
stop	0.5455	0.5000	0.5217	12
threat	0.4118	0.5000	0.4516	14
title	0.9379	0.9601	0.9489	551
together	0.2500	0.1429	0.1818	7
touch	0.6410	0.6757	0.6579	37
toward	0.9790	0.9813	0.9802	428
watch	0.9043	0.9286	0.9163	112
wave	0.6528	0.7966	0.7176	59
write	1.0000	1.0000	1.0000	7
accuracy			0.8400	2731
macro avg	0.7072	0.6909	0.6912	2731
weighted avg	0.8411	0.8400	0.8380	2731

A.7. SetFit

Typologie principale :

	precision	recall	f1-score	support
action	0.8581	0.7963	0.8260	486
aggression	0.6389	0.6133	0.6259	75
aparte	0.8235	1.0000	0.9032	14
delivery	0.8071	0.7465	0.7756	213
entrance	0.6828	0.7734	0.7253	128
exit	0.8023	0.8554	0.8280	242
interaction	0.6771	0.6373	0.6566	102
movement	0.6638	0.6471	0.6553	119
music	0.9583	0.9549	0.9566	577
narration	0.6992	0.7167	0.7078	120
object	0.6864	0.7788	0.7297	208
setting	0.8306	0.8000	0.8150	190

toward	0.9626	0.9733	0.9679	449
accuracy			0.8354	2923
macro avg	0.7762	0.7918	0.7825	2923
weighted avg	0.8374	0.8354	0.8356	2923

Typologie baseline :

	precision	recall	f1-score	support
SHOW	0.0000	0.0000	0.0000	1
aparte	0.4667	0.8750	0.6087	8
away	0.4048	0.4595	0.4304	37
bis	0.1429	0.2500	0.1818	4
call	1.0000	1.0000	1.0000	11
closer	0.8333	0.5660	0.6742	53
cry	0.9286	0.8667	0.8966	15
danse	0.6389	0.7188	0.6765	32
decor	0.9118	0.8378	0.8732	148
dress	0.6667	0.5641	0.6111	39
drink	0.4333	0.8667	0.5778	15
entrance	0.7568	0.7568	0.7568	148
exit	0.9136	0.7852	0.8445	256
explicit	0.6081	0.5056	0.5521	89
fight	0.5417	0.5200	0.5306	25
get	0.7222	0.6724	0.6964	58
give	0.9091	0.7895	0.8451	76
hide	0.7333	0.6111	0.6667	18
hit	0.4000	0.5000	0.4444	12
kiss	0.9565	0.9362	0.9462	47
kneel	0.7500	0.9474	0.8372	19
knock	0.6429	0.6429	0.6429	14
laugh	1.0000	0.9444	0.9714	18
location	0.3860	0.5789	0.4632	38
loud	0.4444	0.5714	0.5000	14
low	0.5217	0.8000	0.6316	15
music	0.1897	0.3333	0.2418	33
narration	0.7303	0.8333	0.7784	78
noise	0.6429	0.6429	0.6429	28
open	0.8333	0.8065	0.8197	31
read	0.9231	0.6857	0.7869	35
silence	0.7500	0.7500	0.7500	12
sing	0.7407	0.7692	0.7547	26
sit	0.8438	0.9000	0.8710	30
stand	0.7308	0.9048	0.8085	21
stop	0.7143	0.4167	0.5263	12
threat	0.5000	0.6429	0.5625	14
title	0.9306	0.8766	0.9028	551
together	0.0571	0.2857	0.0952	7
touch	0.5778	0.7027	0.6341	37
toward	0.9718	0.9673	0.9696	428
watch	0.9286	0.9286	0.9286	112
wave	0.8070	0.7797	0.7931	59

write	0.7778	1.0000	0.8750	7
accuracy			0.8026	2731
macro avg	0.6673	0.6998	0.6727	2731
weighted avg	0.8294	0.8026	0.8120	2731

A.8. gpt-3.5-turbo-0125

Typologie principale, zero-shot, méthode 1er mot :

	precision	recall	f1-score	support
action	0.34	0.19	0.24	512
aggression	0.45	0.07	0.12	70
aparte	0.10	0.75	0.18	16
appearance	0.00	0.00	0.00	0
costumes	0.00	0.00	0.00	0
delivery	0.32	0.17	0.22	190
emotion	0.00	0.00	0.00	0
entrance	0.27	0.82	0.40	128
exit	0.80	0.70	0.75	264
interaction	0.11	0.66	0.19	110
movement	0.15	0.61	0.24	119
music	0.73	0.21	0.33	582
narration	0.11	0.06	0.08	103
object	0.32	0.11	0.16	209
observation	0.00	0.00	0.00	0
reaction	0.00	0.00	0.00	0
setting	0.45	0.78	0.57	178
sound	0.00	0.00	0.00	0
toward	0.67	0.00	0.01	442
transformation	0.00	0.00	0.00	0
accuracy			0.30	2923
macro avg	0.24	0.26	0.17	2923
weighted avg	0.49	0.30	0.28	2923

Typologie principale, zero-shot, méthode n'importe quel mot :

	precision	recall	f1-score	support
action	0.34	0.19	0.24	512
aggression	0.45	0.07	0.12	70
aparte	0.10	0.75	0.18	16
delivery	0.32	0.17	0.22	190
entrance	0.27	0.82	0.40	128
exit	0.80	0.70	0.75	264
interaction	0.11	0.66	0.19	110
movement	0.15	0.61	0.24	119
music	0.73	0.21	0.33	582
narration	0.11	0.06	0.08	103
object	0.32	0.11	0.16	209
setting	0.45	0.78	0.57	178
toward	0.67	0.00	0.01	442
unknown	0.00	0.00	0.00	0
accuracy			0.30	2923
macro avg	0.34	0.37	0.25	2923
weighted avg	0.49	0.30	0.28	2923

Typologie baseline, zero-shot, méthode 1er mot :

	precision	recall	f1-score	support
SHOW	0.00	0.00	0.00	1
air	0.00	0.00	0.00	0
aparte	0.06	0.71	0.10	7
approach	0.00	0.00	0.00	0
away	0.27	0.20	0.23	40
bis	0.00	0.00	0.00	1
call	0.52	0.92	0.67	13
category	0.00	0.00	0.00	0
closer	0.92	0.23	0.36	53
combat	0.00	0.00	0.00	0
cry	0.48	0.93	0.64	15
dance	0.00	0.00	0.00	0
danse	0.23	0.86	0.36	35
decor	0.78	0.75	0.77	170
dialogue	0.00	0.00	0.00	0
dress	0.71	0.78	0.74	50
drink	0.52	0.92	0.67	13
drop	0.00	0.00	0.00	0
entrance	0.65	0.84	0.73	146
exit	0.78	0.78	0.78	237
explicit	0.42	0.08	0.13	64
fall	0.00	0.00	0.00	0
fight	0.38	0.64	0.48	28
get	0.63	0.61	0.62	59
give	0.64	0.90	0.75	52
hide	0.57	0.76	0.65	17
hit	0.46	0.50	0.48	24
implicit	0.00	0.00	0.00	0
kiss	0.89	0.61	0.72	51
kneel	0.90	0.63	0.75	30
knock	0.86	0.60	0.71	20
label	0.00	0.00	0.00	0
laugh	0.88	0.79	0.83	19
location	0.07	0.36	0.11	42
loud	1.00	0.44	0.61	16
low	0.91	0.50	0.65	20
move	0.00	0.00	0.00	0
music	0.43	0.53	0.48	38
narration	0.10	0.06	0.07	68
noise	0.74	0.74	0.74	35
open	0.86	0.54	0.67	35
poetry	0.00	0.00	0.00	0
read	0.50	0.75	0.60	36
show	0.00	0.00	0.00	0
shrug	0.00	0.00	0.00	0
silence	0.58	0.85	0.69	13
silent	0.00	0.00	0.00	0
sing	0.28	0.83	0.41	23
sit	0.77	0.85	0.81	27
speech	0.00	0.00	0.00	0
stand	0.60	0.70	0.65	30
stop	0.60	0.46	0.52	13
suspension	0.00	0.00	0.00	0
taste	0.00	0.00	0.00	0
thought	0.00	0.00	0.00	0

threat	0.50	0.18	0.26	17
title	0.33	0.00	0.00	505
together	0.03	0.67	0.05	6
touch	0.39	0.74	0.51	42
toward	0.40	0.06	0.10	408
wait	0.00	0.00	0.00	0
watch	0.89	0.48	0.62	143
wave	1.00	0.03	0.06	60
write	0.89	0.89	0.89	9
accuracy			0.41	2731
macro avg	0.38	0.39	0.34	2731
weighted avg	0.56	0.41	0.40	2731

Typologie baseline zero-shot, méthode n'importe quel mot :

	precision	recall	f1-score	support
SHOW	0.00	0.00	0.00	1
aparte	0.06	0.71	0.10	7
away	0.27	0.20	0.23	40
bis	0.00	0.00	0.00	1
call	0.52	0.92	0.67	13
closer	0.92	0.23	0.36	53
cry	0.48	0.93	0.64	15
danse	0.23	0.86	0.36	35
decor	0.78	0.75	0.77	170
dress	0.71	0.78	0.74	50
drink	0.52	0.92	0.67	13
entrance	0.65	0.84	0.73	146
exit	0.78	0.78	0.78	237
explicit	0.42	0.08	0.13	64
fight	0.38	0.64	0.48	28
get	0.63	0.61	0.62	59
give	0.64	0.90	0.75	52
hide	0.57	0.76	0.65	17
hit	0.46	0.50	0.48	24
kiss	0.89	0.61	0.72	51
kneel	0.90	0.63	0.75	30
knock	0.86	0.60	0.71	20
laugh	0.89	0.84	0.86	19
location	0.07	0.36	0.11	42
loud	1.00	0.44	0.61	16
low	0.91	0.50	0.65	20
music	0.43	0.53	0.48	38
narration	0.10	0.06	0.07	68
noise	0.74	0.74	0.74	35
open	0.86	0.54	0.67	35
read	0.50	0.75	0.60	36
silence	0.58	0.85	0.69	13
sing	0.27	0.83	0.41	23
sit	0.77	0.85	0.81	27
stand	0.60	0.70	0.65	30
stop	0.64	0.54	0.58	13
threat	0.50	0.18	0.26	17
title	0.33	0.00	0.00	505
together	0.03	0.67	0.05	6
touch	0.39	0.74	0.51	42
toward	0.40	0.06	0.10	408
unknown	0.00	0.00	0.00	0
watch	0.89	0.48	0.62	143
wave	1.00	0.03	0.06	60

write	0.89	0.89	0.89	9
accuracy			0.41	2731
macro avg	0.54	0.55	0.48	2731
weighted avg	0.56	0.41	0.40	2731

Typologie principale, one-shot, méthode 1er mot :

	precision	recall	f1-score	support
action	0.44	0.56	0.49	512
aggression	0.65	0.19	0.29	70
aparte	0.88	0.44	0.58	16
costumes	0.00	0.00	0.00	0
delivery	0.53	0.17	0.26	190
entrance	0.45	0.80	0.57	128
exit	0.86	0.75	0.80	264
gesture	0.00	0.00	0.00	0
interaction	0.19	0.47	0.27	110
movement	0.28	0.61	0.38	119
music	0.92	0.78	0.84	582
narration	0.48	0.15	0.22	103
object	0.38	0.08	0.13	209
setting	0.46	0.81	0.59	178
sound	0.00	0.00	0.00	0
toward	0.91	0.61	0.73	442
accuracy			0.57	2923
macro avg	0.46	0.40	0.39	2923
weighted avg	0.64	0.57	0.57	2923

Typologie principale, one-shot, méthode n'importe quel mot :

	precision	recall	f1-score	support
action	0.44	0.56	0.49	512
aggression	0.65	0.19	0.29	70
aparte	0.88	0.44	0.58	16
delivery	0.53	0.17	0.26	190
entrance	0.45	0.80	0.57	128
exit	0.86	0.75	0.80	264
interaction	0.19	0.47	0.27	110
movement	0.28	0.61	0.38	119
music	0.92	0.78	0.84	582
narration	0.48	0.15	0.22	103
object	0.38	0.08	0.13	209
setting	0.46	0.81	0.59	178
toward	0.91	0.61	0.73	442
unknown	0.00	0.00	0.00	0
accuracy			0.57	2923
macro avg	0.53	0.46	0.44	2923
weighted avg	0.64	0.57	0.57	2923

Typologie principale, two-shots, méthode 1er mot :

	precision	recall	f1-score	support
action	0.46	0.49	0.47	512
aggression	0.65	0.21	0.32	70
aparte	0.83	0.31	0.45	16

costumes	0.00	0.00	0.00	0
delivery	0.66	0.19	0.30	190
entrance	0.46	0.83	0.59	128
exit	0.86	0.74	0.80	264
gesture	0.00	0.00	0.00	0
interaction	0.21	0.57	0.31	110
movement	0.24	0.62	0.35	119
music	0.92	0.83	0.88	582
narration	0.49	0.17	0.25	103
object	0.50	0.09	0.15	209
setting	0.45	0.82	0.58	178
sound	0.00	0.00	0.00	0
toward	0.90	0.62	0.73	442
accuracy			0.58	2923
macro avg	0.48	0.41	0.39	2923
weighted avg	0.66	0.58	0.58	2923

Typologie principal, two-shots, méthode n'importe quel mot :

	precision	recall	f1-score	support
action	0.46	0.49	0.47	512
aggression	0.65	0.21	0.32	70
aparte	0.83	0.31	0.45	16
delivery	0.66	0.19	0.30	190
entrance	0.46	0.83	0.59	128
exit	0.86	0.74	0.80	264
interaction	0.21	0.57	0.31	110
movement	0.24	0.62	0.35	119
music	0.92	0.83	0.88	582
narration	0.49	0.17	0.25	103
object	0.50	0.09	0.15	209
setting	0.45	0.82	0.58	178
toward	0.90	0.62	0.73	442
unknown	0.00	0.00	0.00	0
accuracy			0.58	2923
macro avg	0.55	0.46	0.44	2923
weighted avg	0.66	0.58	0.58	2923

A.9. open-mixtral-8x7b

Typologie principale, zero-shot, méthode 1er mot :

	precision	recall	f1-score	support
action	0.00	0.00	0.00	486
aggression	1.00	0.05	0.10	75
aparte	0.71	0.86	0.77	14
delivery	0.67	0.23	0.34	213
emotion	0.00	0.00	0.00	0
ensemble	0.00	0.00	0.00	0
entrance	0.36	0.84	0.50	128
exit	0.90	0.57	0.70	242
interaction	0.12	0.64	0.20	102
marriage	0.00	0.00	0.00	0
movement	0.12	0.86	0.21	119
music	0.71	0.24	0.35	577

narration	0.05	0.05	0.05	120
object	0.44	0.56	0.49	208
patience	0.00	0.00	0.00	0
setting	0.39	0.81	0.52	190
stage	0.00	0.00	0.00	0
the	0.00	0.00	0.00	0
toward	0.00	0.00	0.00	449
transformation	0.00	0.00	0.00	0
accuracy			0.30	2923
macro avg	0.27	0.28	0.21	2923
weighted avg	0.38	0.30	0.27	2923

Typologie principale, zero-shot, méthode n'importe quel mot :

	precision	recall	f1-score	support
action	0.00	0.00	0.00	486
aggression	1.00	0.05	0.10	75
aparte	0.71	0.86	0.77	14
delivery	0.65	0.23	0.33	213
entrance	0.37	0.79	0.51	128
exit	0.90	0.54	0.67	242
interaction	0.11	0.75	0.20	102
movement	0.12	0.71	0.20	119
music	0.70	0.23	0.35	577
narration	0.05	0.05	0.05	120
object	0.42	0.53	0.47	208
setting	0.38	0.81	0.52	190
toward	0.55	0.04	0.07	449
unknown	0.00	0.00	0.00	0
accuracy			0.30	2923
macro avg	0.43	0.40	0.30	2923
weighted avg	0.46	0.30	0.27	2923

Typologie baseline, zero-shot, méthode 1er mot :

	precision	recall	f1-score	support
SHOW	0.00	0.00	0.00	1
act	0.00	0.00	0.00	0
address	0.00	0.00	0.00	0
air	0.00	0.00	0.00	0
alone	0.00	0.00	0.00	0
animals	0.00	0.00	0.00	0
announcement	0.00	0.00	0.00	0
aparte	0.14	0.71	0.24	7
appearance	0.00	0.00	0.00	0
aside	0.00	0.00	0.00	0
audience	0.00	0.00	0.00	0
away	0.01	0.03	0.02	40
awkwardly	0.00	0.00	0.00	0
backstage	0.00	0.00	0.00	0
based	0.00	0.00	0.00	0
bis	0.00	0.00	0.00	1
call	0.31	0.92	0.46	13
character	0.00	0.00	0.00	0
closer	0.85	0.21	0.33	53
come	0.00	0.00	0.00	0

comparison	0.00	0.00	0.00	0
cover	0.00	0.00	0.00	0
cry	0.87	0.87	0.87	15
dance	0.00	0.00	0.00	0
danse	0.50	0.74	0.60	35
death	0.00	0.00	0.00	0
decor	0.54	0.52	0.53	170
defend	0.00	0.00	0.00	0
dialog	0.00	0.00	0.00	0
dialogue	0.00	0.00	0.00	0
die	0.00	0.00	0.00	0
dress	0.65	0.60	0.62	50
drink	0.57	1.00	0.72	13
eat	0.00	0.00	0.00	0
embrace	0.00	0.00	0.00	0
emotion	0.00	0.00	0.00	0
emportement	0.00	0.00	0.00	0
ensemble	0.00	0.00	0.00	0
entrance	0.53	0.88	0.66	146
entryentrance	0.00	0.00	0.00	0
epode	0.00	0.00	0.00	0
examine	0.00	0.00	0.00	0
exit	0.73	0.73	0.73	237
explicit	0.00	0.00	0.00	64
expression	0.00	0.00	0.00	0
faith	0.00	0.00	0.00	0
fight	0.42	0.79	0.54	28
general	0.00	0.00	0.00	0
gesture	0.00	0.00	0.00	0
get	0.57	0.46	0.51	59
give	0.64	0.87	0.74	52
hide	0.71	0.59	0.65	17
hit	0.53	0.33	0.41	24
humour	0.00	0.00	0.00	0
incrédulité	0.00	0.00	0.00	0
kiss	0.91	0.82	0.87	51
kneel	0.90	0.93	0.92	30
knock	0.81	0.65	0.72	20
laugh	0.74	0.89	0.81	19
length	0.00	0.00	0.00	0
light	0.00	0.00	0.00	0
location	0.07	0.43	0.12	42
look	0.00	0.00	0.00	0
loud	0.86	0.38	0.52	16
love	0.00	0.00	0.00	0
low	0.71	0.60	0.65	20
lower	0.00	0.00	0.00	0
modesty	0.00	0.00	0.00	0
move	0.00	0.00	0.00	0
movement	0.00	0.00	0.00	0
music	0.17	0.61	0.27	38
narration	0.08	0.07	0.08	68
nod	0.00	0.00	0.00	0
noise	0.43	0.74	0.54	35
off	0.00	0.00	0.00	0
open	0.88	0.43	0.58	35
pain	0.00	0.00	0.00	0
pick	0.00	0.00	0.00	0
place	0.00	0.00	0.00	0
please	0.00	0.00	0.00	0
prepare	0.00	0.00	0.00	0

prop	0.00	0.00	0.00	0
provisioning	0.00	0.00	0.00	0
put	0.00	0.00	0.00	0
read	0.52	0.78	0.62	36
repeat	0.00	0.00	0.00	0
same	0.00	0.00	0.00	0
say	0.00	0.00	0.00	0
serve	0.00	0.00	0.00	0
shoulder_shrug	0.00	0.00	0.00	0
show	0.00	0.00	0.00	0
silence	0.55	0.92	0.69	13
sing	0.24	0.43	0.31	23
sit	0.73	0.89	0.80	27
sleep	0.00	0.00	0.00	0
slowly	0.00	0.00	0.00	0
song	0.00	0.00	0.00	0
speak	0.00	0.00	0.00	0
speech	0.00	0.00	0.00	0
stage	0.00	0.00	0.00	0
stand	0.48	0.80	0.60	30
staying	0.00	0.00	0.00	0
stop	1.00	0.38	0.56	13
striking	0.00	0.00	0.00	0
suspension	0.00	0.00	0.00	0
talking	0.00	0.00	0.00	0
the	0.00	0.00	0.00	0
theme	0.00	0.00	0.00	0
threat	0.27	0.18	0.21	17
title	0.53	0.02	0.03	505
to	0.00	0.00	0.00	0
together	0.06	0.83	0.11	6
tone	0.00	0.00	0.00	0
touch	0.54	0.67	0.60	42
toward	0.62	0.28	0.39	408
troc	0.00	0.00	0.00	0
try	0.00	0.00	0.00	0
up	0.00	0.00	0.00	0
upside	0.00	0.00	0.00	0
vers	0.00	0.00	0.00	0
verse	0.00	0.00	0.00	0
voice	0.00	0.00	0.00	0
voicespeech	0.00	0.00	0.00	0
wake	0.00	0.00	0.00	0
walk	0.00	0.00	0.00	0
watch	0.90	0.71	0.79	143
watchtowardhide	0.00	0.00	0.00	0
wave	0.67	0.03	0.06	60
weight	0.00	0.00	0.00	0
whisper	0.00	0.00	0.00	0
write	0.53	0.89	0.67	9
梦幻西游	0.00	0.00	0.00	0
accuracy			0.44	2731
macro avg	0.17	0.19	0.16	2731
weighted avg	0.57	0.44	0.42	2731

Typologie baseline, zero-shot, méthode n'importe quel mot :

	precision	recall	f1-score	support
SHOW	0.00	0.00	0.00	1

aparte	0.10	0.71	0.18	7
away	0.04	0.03	0.03	40
bis	0.00	0.00	0.00	1
call	0.41	0.85	0.55	13
closer	0.85	0.21	0.33	53
cry	0.86	0.80	0.83	15
danse	0.45	0.74	0.56	35
decor	0.57	0.39	0.47	170
dress	0.54	0.52	0.53	50
drink	0.55	0.92	0.69	13
entrance	0.56	0.85	0.67	146
exit	0.72	0.71	0.72	237
explicit	0.00	0.00	0.00	64
fight	0.35	0.61	0.45	28
get	0.58	0.42	0.49	59
give	0.45	0.81	0.58	52
hide	0.77	0.59	0.67	17
hit	0.50	0.33	0.40	24
interaction	0.00	0.00	0.00	0
kiss	0.93	0.73	0.81	51
kneel	0.92	0.80	0.86	30
knock	0.78	0.70	0.74	20
laugh	0.74	0.89	0.81	19
location	0.06	0.40	0.10	42
loud	0.80	0.50	0.62	16
low	0.38	0.55	0.45	20
music	0.19	0.61	0.29	38
narration	0.07	0.07	0.07	68
noise	0.38	0.69	0.48	35
open	0.54	0.37	0.44	35
read	0.46	0.81	0.59	36
silence	0.48	0.92	0.63	13
sing	0.13	0.70	0.22	23
sit	0.42	0.70	0.53	27
stand	0.47	0.80	0.59	30
stop	0.71	0.38	0.50	13
threat	0.27	0.18	0.21	17
title	0.64	0.03	0.05	505
together	0.05	0.83	0.10	6
touch	0.41	0.57	0.48	42
toward	0.63	0.36	0.46	408
unknown	0.00	0.00	0.00	0
watch	0.85	0.73	0.79	143
wave	0.75	0.05	0.09	60
write	0.47	1.00	0.64	9
accuracy			0.43	2731
macro avg	0.45	0.52	0.43	2731
weighted avg	0.57	0.43	0.42	2731

Typologie principale, one-shot, méthode 1er mot (seuls 4 tokens maximum donc une stratégie d'extraction seulement) :

	precision	recall	f1-score	support
action	0.60	0.20	0.30	474
aggression	0.70	0.38	0.49	69
aparte	0.75	0.92	0.83	13
category	0.00	0.00	0.00	0
clameurs	0.00	0.00	0.00	0
delivery	0.51	0.48	0.50	205
entrance	0.50	0.86	0.63	125
exit	0.88	0.66	0.75	225
hearingoverhe	0.00	0.00	0.00	0
interaction	0.29	0.73	0.41	101
movement	0.25	0.45	0.32	116
music	0.00	0.76	0.83	555
narration	0.12	0.08	0.10	114
object	0.47	0.63	0.54	197
observation	0.00	0.00	0.00	0
preparation	0.00	0.00	0.00	0
returning	0.00	0.00	0.00	0
seating	0.00	0.00	0.00	0
setting	0.60	0.83	0.70	184
sound	0.00	0.00	0.00	0
source	0.00	0.00	0.00	0
the	0.00	0.00	0.00	0
toward	0.78	0.79	0.78	430
accuracy			0.59	2000
macro avg	0.32	0.34	0.31	2000
weighted avg	0.65	0.59	0.59	2000

Typologie principale, two-shots, méthode 1er mot :

	precision	recall	f1-score	support
action	0.45	0.07	0.11	486
aggression	0.73	0.40	0.52	75
aparte	0.85	0.79	0.81	14
delivery	0.47	0.54	0.50	213
entrance	0.55	0.88	0.67	128
exit	0.88	0.66	0.75	242
exitobject	0.00	0.00	0.00	0
interaction	0.26	0.69	0.37	102
movement	0.20	0.61	0.30	119
music	0.92	0.85	0.89	577
narration	0.18	0.10	0.13	120
object	0.50	0.61	0.55	208
observation	0.00	0.00	0.00	0
setting	0.59	0.83	0.69	190
sound	0.00	0.00	0.00	0
the	0.00	0.00	0.00	0
toward	0.92	0.76	0.83	449
transformation	0.00	0.00	0.00	0
accuracy			0.59	2923

macro avg	0.42	0.43	0.40	2923
weighted avg	0.65	0.59	0.58	2923

Typologie principale, two-shots, méthode n'importe quel mot :

	precision	recall	f1-score	support
action	0.44	0.06	0.10	486
aggression	0.74	0.35	0.47	75
aparte	0.69	0.79	0.73	14
delivery	0.49	0.54	0.51	213
entrance	0.65	0.81	0.72	128
exit	0.88	0.60	0.71	242
interaction	0.20	0.72	0.31	102
movement	0.22	0.54	0.31	119
music	0.92	0.84	0.88	577
narration	0.18	0.12	0.14	120
object	0.49	0.64	0.56	208
setting	0.60	0.80	0.69	190
toward	0.83	0.85	0.84	449
unknown	0.00	0.00	0.00	0
accuracy			0.59	2923
macro avg	0.52	0.55	0.50	2923
weighted avg	0.64	0.59	0.58	2923

B. Exemples de sorties de Mistral-7B avec le prompt ReAct.

Didascalie : En montrant la ville.

Etiquette réelle : action

Réponse : "Thought: This stage direction seems to be in French, and it might be related to the setting or scenery of the scene.

Action: Translate the stage direction to understand its meaning.

Observation: After translating, ""En montrant la ville"" means ""Showing the city.""

Thought: I now know the final answer.

Final Answer: setting. This stage direction indicates the setting or scenery of the scene, specifically pointing out that the city should be shown or displayed on stage."

Didascalie : Alfonse et Marc Antoine les frappent chacun le sien.

Etiquette attendue : aggression

Réponse : "Thought: This stage direction seems to be written in French, and it describes some sort of action being taken by two characters, Alfonse and Marc Antoine, against others, possibly other characters. I need to understand what exactly they are doing before I can classify it.

Action: Translate the stage direction to understand the context.

Observation: The translated stage direction is: ""Alfonse and Marc Antoine each strike their own.""

Thought: Now that I understand the context, I can classify this stage direction.

Final Answer: action"

Il frappe trois fois.,action,"Thought: This stage direction seems to involve some sort of action, but I need to consider if it fits into any of the other categories more specifically.

Action: I'll think about the context and meaning of ""Il frappe trois fois.""

Observation: It is French for ""He knocks three times."" It seems to be an action, but I should consider if it could be classified as aggression or interaction.

Thought: I now know the final answer.

Final Answer: action

Note: While it's possible that the knocking could be interpreted as aggressive or intimidating, the stage direction itself doesn't necessarily imply that, so I'm sticking with the 'action' category."

C. Productions scientifiques reliées au mémoire

Schneider, A. (2023). *Classification automatique du type de didascalie*. Journée d'étude : Données de théâtre, petit et grand jeu : Le bon, le brut et le futé. Strasbourg. MISHA. <https://doi.org/10.5281/zenodo.10644192>

Schneider, A., & Ruiz, P. (2024). Stage Direction Classification in French Theater : Transfer Learning Experiments. *Proceedings of LaTeCH-CLfL 2024*, 280.