Make text look like speech: disfluency generation using sequence-to-sequence neural networks Domain

Henri Lasselin

Master research Internship

Internship report

Make text look like speech: disfluency generation using sequence-to-sequence neural networks

Domain: Document and Text Processing - Machine Learning

Author:
Henri Lasselin

Supervisor:
Gwénolé Lecorvé
Expression - Irisa

ENSSAT
LANNION

Abstract: La synthèse de discours spontanés naturels est un défi à relever. Une manière de s'en approcher est de produire des discours disfluents. Dans ce document, nous présentons le travail réalisé lors d'un stage de master. Nous expérimentons plusieurs modèles neuronaux séquence-à-séquence ayant pour but de transformer un texte fluide en texte disfluent. Nous proposons également plusieurs métriques évaluant cette tâche. Les résultats obtenus tendent à montrer que cette tâche n'est pas aisée pour des modèles neuronaux.

Table des matières

1 Introduction

Ces dernières années, de nombreuses avancées ont été faites en synthèse de la parole, notamment grâce aux progrès en apprentissage automatique. Celles-ci permettent de produire des discours clairs et fluides comme lors de discours préparés ou de lectures de texte. Cependant, la production de parole spontanée reste un défi à relever car ce type de parole est moins régulier et plus expressif. En particulier, il contient souvent des disfluences, sujet d'étude de ce stage.

Les disfluences sont des phénomènes qui interrompent le flux de la parole, sans ajouter de contenu propositionnel [Tree, 1995]. Longtemps, elles ont été considérées comme étant des éléments perturbant le discours, c'est pourquoi des recherches ont été faites sur la suppression de celles-ci [Hassan et al., 2014]. Cependant, plusieurs études ont montré que les disfluences présentent plusieurs intérêts pour la communication. Elles permettent de prévenir les auditeurs de la complexité du discours à suivre, les aident à en comprendre la structure et rendent possible la correction d'erreurs faites précédemment [Tree, 2001, Rose, 1998] (cités par [Adell et al., 2012]). Les disfluences sont donc très importantes pour produire une parole spontanée et naturelle.

Récemment, des travaux ont été réalisés pour générer des textes disfluents [Qader et al., 2014, Qader, 2017]. Ces travaux proposent une implémentation fondée sur des modèles de langage et des champs aléatoires conditionnels. Notre objectif est de comprendre le domaine et de réaliser cette même tâche mais grâce à des réseaux de neurones et sous l'angle d'un modèle séquence-à-séquence. En effet, la génération de textes disfluents peut être vue comme la transformation d'un texte fluide (une séquence de mots) en un texte disfluent (une autre séquence de mots). Notre travail étant exploratoire, notre objectif secondaire est de formaliser le problème de production de disfluences. Notons que la synthèse audio des discours disfluents, bien qu'étant la motivation de fond, reste en dehors du cadre du stage. Nous nous concentrerons uniquement sur la production de textes disfluents.

Ce document présente le travail que j'ai réalisé en ce sens lors d'un stage de master. Afin de suivre les pistes les plus adéquates, nous présenterons tout d'abord une étude bibliographique sur les disfluences ainsi que sur les modèles séquence-à-séquence. Puis, nous formaliserons le problème à traiter et proposerons des métriques pour évaluer des séquences disfluentes. Nous décrirons ensuite les corpus sur lesquels nous avons travaillé. Nous présenterons alors différents modèles neuronaux que nous avons expérimentés. Enfin, nous discuterons des résultats obtenus afin de conclure sur le travail réalisé et sur les perspectives du domaine.

2 État de l'art

Cette partie est une étude bibliographique préliminaire à notre travail. Nous étudierons tout d'abord les disfluences, leurs caractéristiques et la manière dont leur production est actuellement traitée. Puis, nous présenterons les réseaux de neurones et en particulier, leur utilisation pour des tâches séquence-à-séquence.

2.1 Étude des disfluences

Nous commençons par donner la façon dont les disfluences sont structurées. Nous présentons ensuite différentes catégories de disfluences avant de nous intéresser à leur composition. Enfin, nous verrons comment les travaux traitent actuellement de ce phénomène dans la littérature.

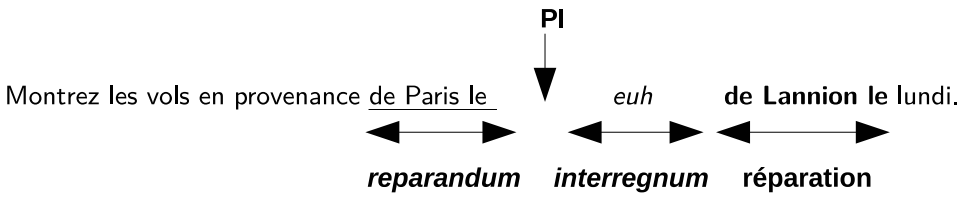

FIGURE 2.1 – Structure des disfluences proposée par [Shriberg, 1994]

2.1.1 Structure des disfluences

Une disfluence est une discontinuité dans un énoncé. Cette discontinuité peut introduire des modifications de l'énoncé autour de son point central. [Shriberg, 1994] a révélé des régularités dans la structure des disfluences. La structure apportée dans ce travail est présentée dans la figure 2.1. Elle permet de représenter tous les types de disfluences comme une suite de sections, chacune ayant un rôle dans la disfluence. Ces sections s'articulent autour d'un point central, dit point d'interruption (PI), qui marque une coupure dans le discours : c'est ici que le locuteur se rend compte d'une erreur. Avant ce PI, le *reparandum* correspond à la partie erronée du discours. Selon les personnes, le *reparandum* est uniquement le mot contenant l'erreur ou bien la zone complète qui contient ce mot (c'est cette dernière version qui est montrée dans la figure). Ensuite, juste après le PI, l'*interregnum* indique l'interruption de la parole par un intervalle entre l'erreur et la réparation de celle-ci. Enfin, la réparation rectifie l'erreur faite dans le *reparandum*. Par la suite, nous utiliserons le même format que sur la figure 2.1 : le *reparandum* sera souligné, l'*interregnum* sera en italique et, enfin, la réparation sera en gras. De plus, le PI sera indiqué entre parenthèses.

Il existe plusieurs types de disfluences, tous conformes à ce modèle mais avec des particularités, et chacun ayant des origines, des rôles et des impacts différents. On peut relever les disfluences suivantes : les allongements, les répétitions, les révisions, les faux départs ainsi que les pauses. Parmi ces dernières, on peut distinguer les pauses silencieuses et les pauses remplies. On peut noter que la terminologie peut varier en fonction des articles.

Les PI étant les éléments centraux des disfluences, il est capital de prédire leur fréquence et leurs emplacements au sein du discours afin de produire ces dernières. On peut tout d'abord relever que la fréquence des disfluences dépend de beaucoup d'éléments [Shriberg, 1994]. Par exemple, elle varie en fonction du genre : les femmes feraient globalement moins de disfluences que les hommes. Les disfluences surviennent plus fréquemment lorsque le discours n'est pas préparé car le locuteur doit anticiper ce qu'il va dire tout en parlant. Il existe également des positions où insérer les PI. Par exemple, il y a davantage de disfluences entre deux mots qui n'ont pas l'habitude d'être juxtaposés. Les disfluences se produisant plus souvent lorsque le locuteur réfléchit à la suite de sa phrase, près de la moitié de celles-ci survient avant des mots-outils [Maclay and Osgood, 1959, Blankenship and Kay, 1964] (cités par [Shriberg, 1994]). La position des disfluences est importante car [Dall et al., 2014] montre qu'il existe des endroits propices aux disfluences et qu'à l'inverse, la parole est perçue comme peu naturelle lorsqu'elles sont insérées à de mauvais endroits.

Cependant, ces articles s'intéressent aux caractéristiques des disfluences et non à leurs origines. Pourtant, les disfluences dépendent de l'intention du locuteur et de multiples autres facteurs liés à son état mental (énervé, stressé...) ou encore son expérience personnelle. Par exemple, le locuteur est en train de parler et le mot « voiture » vient dans la conversation. Il va alors peut-être se rappeler

de sa propre voiture et peut produire une disfluence en conséquence.

Il existe plusieurs manières de catégoriser les disfluences : selon leur complexité ou selon leur rôle par exemple. Nous avons choisi d'étudier chaque type de disfluences en les regroupant selon leur influence sur différents niveaux d'abstraction du langage, à savoir la prosodie, la syntaxe et la sémantique.

2.1.2 Niveau prosodique

Certaines disfluences influent nettement sur la prosodie, c'est-à-dire qu'elles modifient le rythme et l'intonation du discours. On retrouve dans cette catégorie les pauses silencieuses, les allongements et les troncatures.

2.1.2.1 Pauses silencieuses

Les pauses sont des interruptions dans le discours. Elles ne possèdent ni *reparandum*, ni réparation. Dans le cas des pauses silencieuses, aucun son n'est prononcé dans l'*interregnum* : il n'y a qu'un silence (*cf.* l'exemple suivant).

Montrez-moi les vols en provenance de (PI) **silence** Lannion le lundi.

Il est tout d'abord nécessaire de mentionner que toutes les pauses silencieuses ne sont pas des disfluences et il peut être compliqué de distinguer les deux [Tree, 1995]. Par exemple, les pauses dues à la ponctuation sont inhérentes au langage.

Les pauses silencieuses sont les disfluences les plus fréquentes [Betz et al., 2015]. Elles surviennent aussi bien en anticipation d'une difficulté que lorsque celle-ci est déjà survenue. Cette étude montre que la durée de ces pauses est très variable : elle peut atteindre 800ms sans être perçue comme non naturelle. [Shriberg, 1994] note que les pauses silencieuses peuvent aider les interlocuteurs à discerner la partie réparation dans le discours.

2.1.2.2 Allongements

Les allongements sont des disfluences particulières. En effet, un allongement seul contient un *reparandum* sans réparation associée, il ne possède pas d'*interregnum* non plus : il consiste en le prolongement de la syllabe précédant le PI. Tout comme les pauses silencieuses, les allongements augmentent la durée du discours et permettent au locuteur de réfléchir à ce qu'il va dire après. Ils sont complexes à étudier car il est difficile de déterminer de manière absolue à partir de quelle durée une syllabe est considérée comme allongée, cette durée étant propre à chaque corpus. Cependant, [Betz et al., 2015] a montré que dans un contexte donné (même personne, même corpus), une syllabe allongée dure généralement deux fois plus longtemps qu'une syllabe normale.

2.1.2.3 Troncatures

Les troncatures sont des disfluences pour lesquelles le PI se situe au milieu d'un mot. Cela survient lorsque le locuteur perçoit son erreur alors qu'il prononce un mot.

Selon les corpus, le taux de troncatures par disfluence dans la parole spontanée varie énormément (de 22 à 60%) [Shriberg, 1994]. Pour le corpus utilisé dans [Betz et al., 2015], 15% des disfluences possèdent une troncature. Il n'y a pas d'explication claire à cette variation. Ce dernier article montre

également que les troncatures sont peu appréciées dans les communications humaines mais qu'elles restent nécessaires car elles facilitent la correction d'erreur.

2.1.3 Niveau syntaxique

Les disfluences peuvent également modifier la syntaxe en introduisant de nouveaux syntagmes. Ces disfluences changent la structure de la phrase, sans en modifier le sens. Dans cette catégorie, on retrouve les pauses remplies et les répétitions.

2.1.3.1 Pauses remplies

Les pauses remplies sont similaires aux pauses silencieuses dans le sens où elles ne sont composées que d'un *interregnum*. Cependant, celui-ci ne contient pas un silence mais contient des éléments qui n'ajoutent aucune information. En anglais, ces pauses sont *« uh »* et *« um »* (resp. pauses courtes et longues), les éléments correspondants en français sont « euh » et « hmmm ».

Certaines études considèrerent que les marqueurs de discours tels que « enfin », « voilà » et « en fait » (en anglais *« I mean »*, *« well »* et *« you know »*) font également partie de cette catégorie. La phrase suivante est un exemple de pause remplie :

Montrez-moi les vols en provenance de (PI) *euh* Lannion le lundi.

[Shriberg, 1994] relève que contrairement aux autres disfluences, le taux de pauses remplies est stable, peu importe la complexité de la tâche. Ceci peut expliquer pourquoi la plupart des systèmes de génération de discours disfluents se concentraient sur cette famille de disfluences. Pour [Maclay and Osgood, 1959], celles-ci reflètent un besoin de pauses combiné à un besoin de continuer de parler, c'est-à-dire d'éviter les silences. De même, en s'intéressant à la gestuelle du locuteur, on remarque que les gestes et les pauses remplies ne coexistent pas. Ainsi, si on interdit au locuteur de faire des gestes en s'exprimant, il aura davantage tendance à faire des pauses remplies : les gestes et les pauses remplies semblent occuper le même rôle dans la parole, bien que celui-ci ne soit pas clairement établi [Christenfeld et al., 1991].

Une autre étude a également montré que la production de ces disfluences détériorait le naturel de la synthèse de la parole [Betz et al., 2015]. Cependant, ce manque de naturel était davantage dû à la mauvaise qualité de synthèse de ces disfluences qu'à leur présence.

2.1.3.2 Répétitions

Les répétitions consistent en la répétition, sans modification, d'une partie de la phrase. Elles n'ont pas d'*interregnum* et la partie réparation est l'exacte recopie du *reparandum*, comme dans l'exemple suivant :

Montrez-moi les vols en provenance de Lannion le (PI) **le** lundi.

[Tree, 1995] montre que, dans un discours spontané, les répétitions permettent d'augmenter l'attention de l'auditeur sur le mot suivant celle-ci. En effet, le temps de réaction pour détecter le mot suivant est plus court avec répétition, que sans. Cependant, il relève qu'une répétition générée artificiellement n'avait pas d'influence sur le temps de réaction, sûrement à cause de la faible qualité audio des répétitions ajoutées. Les répétitions n'ont donc, au pire, aucune influence sur la

parole spontanée et, au mieux, aident à la compréhension de celle-ci : ce sont donc des disfluences intéressantes à produire. Elles peuvent permettre au locuteur de gagner du temps pour réfléchir à ses prochains mots. Elles peuvent aussi aider les interlocuteurs à se recentrer sur le discours après une longue pause [Shriberg, 1994].

2.1.4 Niveau sémantique

Les disfluences peuvent avoir un impact sur la sémantique en introduisant des variations de sens du discours. Cette catégorie comprend les révisions et les faux-départs.

2.1.4.1 Révisions

Les révisions ont la même structure que les répétitions, mais la partie réparation est différente du *reparandum*. L'exemple suivant montre le rôle principal des révisions : celui de corriger une erreur faite précédemment.

Montrez-moi les vols en provenance de Paris (PI) **de Lannion** le lundi.

Cet exemple montre également un phénomène relevé par [Shriberg, 1994] : la répétition d'une partie de la phrase précédant le mot corrigé (« de » dans l'exemple). Ce phénomène survient moins souvent lorsque l'erreur est phonétique comme dans la phrase suivante :

Montrez-moi les vols en provence (PI) **provenance** de Lannion le lundi.

Cela peut s'expliquer par le fait que l'interlocuteur comprend immédiatement la correction d'une erreur phonétique (il l'assimile à une répétition).

Dans certains cas, la partie réparation ne contredit pas le *reparandum* [Shriberg, 1994]. En effet, les révisions peuvent également servir à apporter des précisions, comme par exemple dans la phrase suivante :

Montrez-moi les vols (PI) **les dix premiers vols** en provenance de Lannion le lundi.

Ce dernier cas est sujet à discussion car toutes les précisions ne sont pas des révisions. Cela dépend de l'intention du locuteur : s'il avait planifié une précision, ce n'est pas une disfluence. À l'inverse, si de son point de vue il s'est trompé, on considère la précision comme une disfluence.

2.1.4.2 Faux départs

Les faux départs sont des disfluences pour lesquelles le locuteur abandonne la phrase qu'il énonçait et en commence une nouvelle. Ils peuvent donc être vus comme un cas particulier de révisions. L'exemple suivant montre un faux départ :

Je souhaite (PI) **Montrez-moi les vols en provenance de Lannion le lundi.**

[Tree, 1995] a étudié les faux départs. Cette étude montre que ces derniers perturbent les interlocuteurs. En effet, les interlocuteurs se forment une image mentale du discours qu'ils écoutent. Après un faux départ, ils doivent alors détruire cette image et en construire une nouvelle. Ces travaux montrent également que cette gêne est la même quelque soit la position du faux départ dans la phrase.

2.1.5 Compositions de disfluences

Au delà des archétypes présentés pour chaque type de disfluences, il est fréquent que plusieurs disfluences coexistent dans un énoncé. [Betz et al., 2015] propose deux scénarios. Le premier survient lorsque le locuteur se rend compte tardivement d'un changement dans le plan qu'il s'était fixé. Il interrompt immédiatement la parole (ce qui peut produire une troncature) puis, va chercher un nouveau plan de discours. Tant qu'il ne le trouve pas, il va produire d'autres disfluences en commençant par des pauses silencieuses, puis des pauses remplies. Le deuxième scénario est similaire au premier à l'exception près que l'information d'un changement de plan arrive tôt. Le locuteur va alors commencer par allonger la syllabe qu'il prononce avant de produire les autres disfluences.

[Betz et al., 2015] a également remarqué qu'une troncature seule (qui génère un fragment de mot) semblait abrupte et peu naturelle. Il suggère que cela vient du fait que, dans le langage naturel, le locuteur change d'intonation et/ou allonge la syllabe avant de couper le mot. De plus, le locuteur coupe rarement un mot sans effectuer une révision ensuite. Par exemple, on peut supposer qu'il a mal prononcé un mot, s'en rend compte et se corrige immédiatement :

Montrez-moi les vols en provenance de Lon (PI) *enfin* **Lannion** le lundi.

Un marqueur de discours est également présent dans cet exemple. En effet, les pauses (et donc les marqueurs de discours) servent à ralentir le discours. Elles sont donc très utiles en combinaison des disfluences influant sur la sémantique. En outre, cela permet aux interlocuteurs de comprendre qu'une correction du discours va avoir lieu. Ce rôle peut aussi être rempli par les répétitions [Tree, 1995].

2.1.6 Production de disfluences

Les disfluences ont longtemps été perçues comme des éléments perturbant le discours. Cependant, dans l'objectif de générer une parole spontanée naturelle, plusieurs travaux ont récemment été réalisés afin de générer des discours disfluents. Toutefois, ces travaux ne se concentrent que sur certains types de disfluences. Rappelons que notre perspective de travail se concentre sur la production de disfluences, et non leur synthèse.

Parmi les travaux de l'état de l'art, [Adell et al., 2007] n'étudie que la production de pauses remplies « uh » et « um ». La méthode consite à trouver tout d'abord le point d'interruption (PI), puis à produire la pause. La recherche de PI se fait grâce à un modèle de langage et à un arbre de décision. Ce dernier classifie chaque mot du texte en deux catégories : ceux qu'il faut faire suivre d'une disfluence et les autres. Afin de faire cette classification, l'arbre de décision s'appuie sur des probabilités du modèle de langage et des étiquettes morphosyntaxiques (nom, verbe, déterminant...).

Récemment, des travaux ont été faits sur la production de plusieurs types de disfluences : les répétitions et les pauses [Qader, 2017]. Le modèle théorique proposé permet néanmoins de produire des révisions et donc des faux-départs.

Le principe de cette méthode est de voir une disfluence comme le résultat d'une fonction qui transforme une phrase fluide. Ce processus de transformation est décomposé : il y a ainsi une fonction de transformation f_T par type de disfluences T. Ces fonctions prennent en paramètre une séquence de mots sous forme de vecteur. Elles donnent en résultat une autre séquence de mots ayant la disfluence associée. En se basant sur ce processus, plusieurs disfluences peuvent être générées en composant les fonctions comme le montre la figure 2.2. Chaque disfluence peut être générée 0, 1 ou plusieurs fois. L'ordre de composition est fixé : les révisions sont générées en premier car elles sont

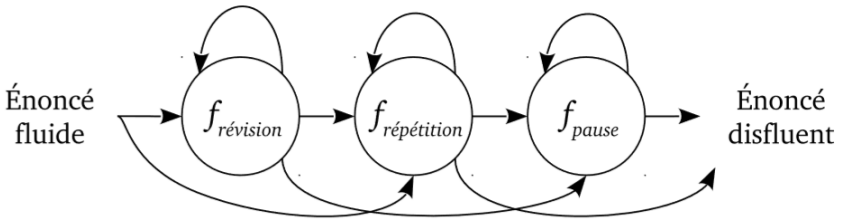

FIGURE 2.2 – Processus de production de disfluences (Source : [Qader et al., 2014])

.

complexes. Les répétitions viennent ensuite. En effet, si elles étaient produites avant les révisions, ces dernières pourraient interrompre une répétition. Les pauses sont produites en dernier car elles peuvent être insérées au milieu des autres disfluences et elles sont plus faciles à produire.

[Qader, 2017] propose une implémentation de ce principe. Celle-ci peut produire des répétitions et des pauses. Pour chaque disfluence, la génération est décomposée en la prédiction de PI puis en l'insertion de la disfluence. La prédiction de PI utilise un CRF afin d'étiqueter les mots (l'étiquette indique si le mot est suivi d'une disfluence ou non). Afin de faire cette prédiction, le texte fluide est au préalable annoté avec des descripteurs (POS, fréquence du mot, nombre de syllabes, position du mot...). Pour l'insertion de disfluences, l'algorithme génère un ensemble de phrases candidates selon la disfluence. Ensuite, celles-ci sont évaluées à l'aide d'un modèle de langage et la phrase la plus probable est conservée. Un critère d'arrêt existe pour chaque disfluence. Il permet de choisir le degré de disfluence de la phrase.

Enfin, l'évaluation des discours disfluents produits n'est pas un sujet d'étude dans la littérature mais est un aspect primordial qu'il faut considérer avec attention. Une évaluation objective peut être réalisée en s'intéressant aux statistiques des systèmes comme le rappel ou la précision. Cependant, une telle méthode n'est pas suffisante pour évaluer la production de disfluences. En effet, plusieurs positions de disfluences sont acceptables pour chaque énoncé et, pour chacune de ces positions, plusieurs types de disfluences sont également possibles [Dall et al., 2014]. On peut alors se tourner vers une évaluation subjective : les énoncés disfluents sont présentés à des testeurs qui en évaluent la qualité. Ces énoncés peuvent être présentés sous forme écrite ou sonore. Dans le premier cas, les testeurs devront imaginer que le texte est prononcé. Dans le second cas, les énoncés peuvent être prononcés par des humains ou par un système de synthèse. D'un côté, faire prononcer les énoncés par des humains coûte cher et prend du temps. D'un autre côté, synthétiser les énoncés est plus abordable, mais les résultats de l'évaluation dépendent énormément de la qualité du système de synthèse qui est généralement faible pour des textes disfluents car les systèmes ne sont pas prévus pour ce type de texte. Une fois ce premier travail effectué, des questions sont posées aux testeurs, comme par exemple « Pensez-vous que les pauses remplies rendent le texte (plus/autant/moins) naturel ? ». Ces questions sont à choisir minutieusement car les résultats peuvent être influencés par celles-ci [Dall et al., 2014]. Malgré son coût et le temps requis par les testeurs, l'évaluation subjective reste cependant la méthode la plus pertinente pour évaluer la production de disfluences.

Nous avons donc étudié les différentes disfluences que l'on peut être amené à produire. La production de disfluences pouvant être vue comme le passage d'une séquence de mots fluide en une séquence de mots disfluente, nous allons maintenant nous intéresser aux modèles séquence-à-séquence.

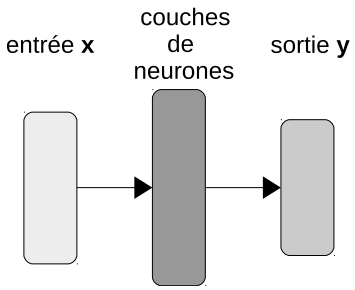

FIGURE 2.3 – Schéma d'un réseau de neurones.

2.2 Modèles séquence-à-séquence

Dans cette partie, nous allons étudier les modèles séquence-à-séquence. En effet, ces modèles ont pour but de générer une séquence d'éléments à partir d'une autre séquence et peuvent donc nous permettre de produire un énoncé disfluent à partir d'un énoncé fluide. Nous allons commencer par présenter différents modèles neuronaux d'apprentissage automatique pouvant réaliser des tâches séquence-à-séquence. Enfin, nous regarderons quelles applications ont ces réseaux de neurones séquence-à-séquence.

2.2.1 Réseaux de neurones

Les réseaux de neurones sont des modèles capables d'apprendre des relations très complexes à partir de données. Comme le montre la figure 2.3, ce sont des modèles qui prennent un vecteur d'entrée et le transforme successivement grâce à des couches de neurones jusqu'à produire une sortie, elle aussi sous forme vectorielle. Les neurones sont des cellules qui reçoivent des valeurs numériques en entrée, les transforment via une fonction f, et transmettent le résultat aux neurones de la couche suivante. La transformation réalisée par un neurone consiste tout d'abord à faire une somme du vecteur d'entrée pondérée par un vecteur de poids $\boldsymbol{\theta}$, puis à appliquer une fonction d'activation (par exemple la fonction sigmoïde).

Le but de la phase d'apprentissage est de trouver la pondération $\boldsymbol{\theta}$ qui minimise l'erreur entre la sortie du réseau et celle souhaitée. Pour cela, la technique la plus utilisée est celle de la rétropropagation du gradient qui consiste à corriger les erreurs en fonction de l'importance des éléments qui y ont participé.

On peut représenter un réseau de neurones qui admet $\mathbf{x} \in \mathbb{R}^n$ en entrée par :

$$\mathbf{y} = f_{\boldsymbol{\theta}}(\mathbf{x}), \tag{2.1}$$

avec $\mathbf{y} \in \mathbb{R}^m$ la sortie et $\boldsymbol{\theta} \in \mathbb{R}^{n \times m}$ les paramètres du réseau. Il existe de très nombreuses architectures de réseaux de neurones qui diffèrent selon la manière dont les couches sont organisées, les connexions entre les neurones ou encore le fonctionnement interne des neurones. Les prochaines sections listent les plus pertinentes d'entre elles pour notre problème.

2.2.1.1 Réseaux de neurones récurrents

Les réseaux de neurones récurrents (RNN pour *Recurrent Neural Network*) sont des réseaux de neurones destinés à traiter des données séquentielles, c'est-à-dire produire une séquence $[\mathbf{y}_1, \ldots, \mathbf{y}_T]$ à partir d'une séquence $[\mathbf{x}_1, \ldots, \mathbf{x}_T]$ d'entrées.

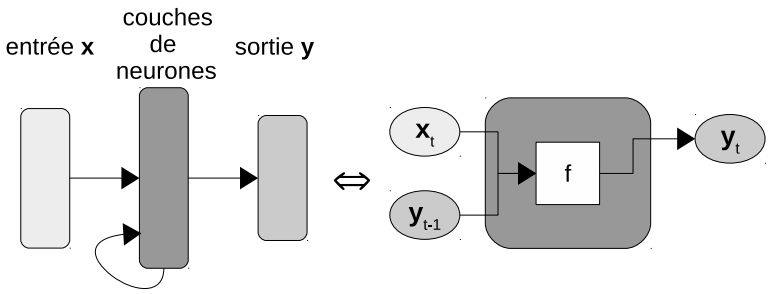

FIGURE 2.4 – Schéma d'un bloc RNN

La structure des RNN est semblable à celle des réseaux de neurones classiques. Cependant, la sortie de la couche de neurones correspondant au $(t-1)$-ième élément de la séquence est réinjectée en tant qu'entrée de la couche pour l'élément t (*cf.* figure 2.4). Ainsi, la t-ième sortie $\mathbf{y}_t$ d'un bloc RNN se calcule comme :

$$\mathbf{y}_t = f_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{y}_{t-1}), \tag{2.2}$$

avec $\mathbf{y}_{t-1}$ la $(t-1)$-ième sortie, $\mathbf{x}_t$ la t-ième entrée et $\boldsymbol{\theta}$ les paramètres du réseau. Ces réseaux permettent donc de prendre en compte une dépendance entre les entrées d'une séquence.

Ils sont beaucoup utilisés dans les travaux de traitement du langage naturel pour plusieurs raisons [Young et al., 2017]. Premièrement, dans le langage naturel, le sens d'un mot dépend en partie des mots précédents. De plus, les phrases n'ont pas toutes le même nombre de mots, or, les réseaux de neurones ont un vecteur d'entrées de taille fixe. Le jeu de chaînage de l'équation (2.2) permet d'apporter une solution à ces deux problèmes.

La technique utilisée pour l'apprentissage est une variante de celle utilisée pour les réseaux de neurones classiques : la rétropropagation du gradient à travers le temps. Cependant, les RNN connaissent le problème de disparition du gradient [Pascanu et al., 2013]. À cause de ce dernier, les systèmes ont du mal à apprendre de longues dépendances. D'autres réseaux de neurones, comme les *Long Short-Term Memory*, permettent de contourner ce phénomène.

2.2.1.2 Long Short-Term Memory

Les *Long Short-Term Memory* (LSTM) sont un type particulier de RNN beaucoup plus complexes que ces derniers [Greff et al., 2017]. Comme le représente le système d'équations (2.3), la sortie $\mathbf{y}_t$ dépend toujours de la sortie $\mathbf{y}_{t-1}$ et de l'entrée $\mathbf{x}_t$. Toutefois, elle dépend également d'une nouvelle information récurrente $\mathbf{i}_{t-1}$, appelée mémoire, qui saisit les dépendances à long terme. Pour produire la valeur $\mathbf{i}_t$ à l'instant t, le bloc LSTM est composé de plusieurs « portes » qui ont pour objectif de conserver, supprimer ou modifier de l'information (voir la figure 2.5 pour plus de détails).

$$\begin{cases} \mathbf{y}_t = f_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{y}_{t-1}, \mathbf{i}_{t-1}) \\ \mathbf{i}_t = g_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{y}_{t-1}, \mathbf{i}_{t-1}) \end{cases}, \tag{2.3}$$

avec $\boldsymbol{\theta}$ les paramètres du réseau. Grâce à cette structure complexe, les LSTM sont efficaces pour capturer les dépendances à long terme, contrairement aux RNN classiques. De plus, ils possèdent leurs

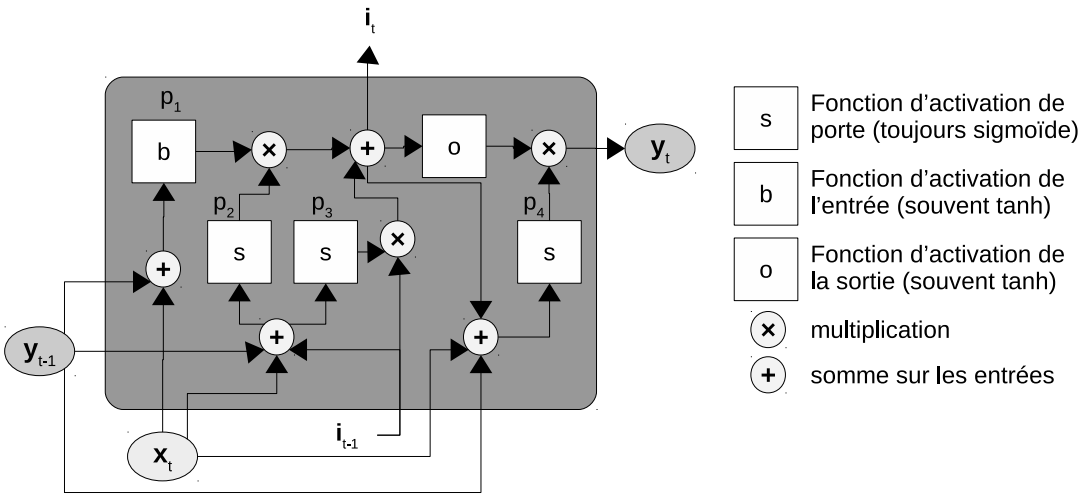

FIGURE 2.5 – Schéma d'un bloc LSTM. p_1 est appelé bloc d'entrée, p_2 porte d'entrée, p_3 porte d'oubli et p_4 porte de sortie. (D'après [Greff et al., 2017])

avantages (dépendances à court-terme, séquences de longueur variable). Cependant, l'apprentissage est plus coûteux à cause de ses nombreuses connexions et donc de ses paramètres à estimer.

2.2.1.3 Gated Recurrents Units

Les *Gated Recurrents Unit* (GRU) sont une autre variante de RNN, introduite récemment par [Cho et al., 2014]. Ces réseaux sont plus simples que les LSTM car ils possèdent une porte en moins.

Une étude a comparé les GRU, les LSTM et les RNN classiques [Chung et al., 2014]. Elle montre que, pour des tâches autres que le traitement du langage naturel (l'étude ne s'intéressant pas à ce domaine), les réseaux GRU et LSTM sont plus performants que les RNN classiques. Toutefois, cette étude ne parvient pas à déterminer lequel des deux premiers est le meilleur. Par conséquent, de par leur plus grande simplicité, les réseaux GRU seront privilégiés lorsque la puissance de calcul disponible est limitée [Greff et al., 2017].

2.2.1.4 Generative Adversarial Networks

Les Generative Adversarial Networks (GAN), proposés par [Goodfellow et al., 2014] font partie de la catégorie des modèles génératifs, c'est-à-dire ayant pour but de créer des données. Comme en témoigne la figure 2.6, le principe des GAN est de mettre en concurrence deux modèles. Le premier est un réseau générateur qui est chargé de produire des données à partir d'un vecteur de bruit. Le second est un réseau discriminateur qui apprend à différencier les données générées de données réelles fournies par l'utilisateur. Ce modèle peut être utilisé pour produire toutes sortes de données, aussi bien des images que des textes ou des signaux audio.

Dans le cas de la production de disfluences, nous pourrions supposer que les deux réseaux seraient des réseaux récurrents tels que des LSTM. Le réseau discriminateur recevrait en entrée des séquences de mots disfluents, soit créées par le réseau générateur, soit provenant d'un corpus. Ces séquences seraient fournies mot par mot au réseau qui devra prédire si celles-ci sont réelles ou

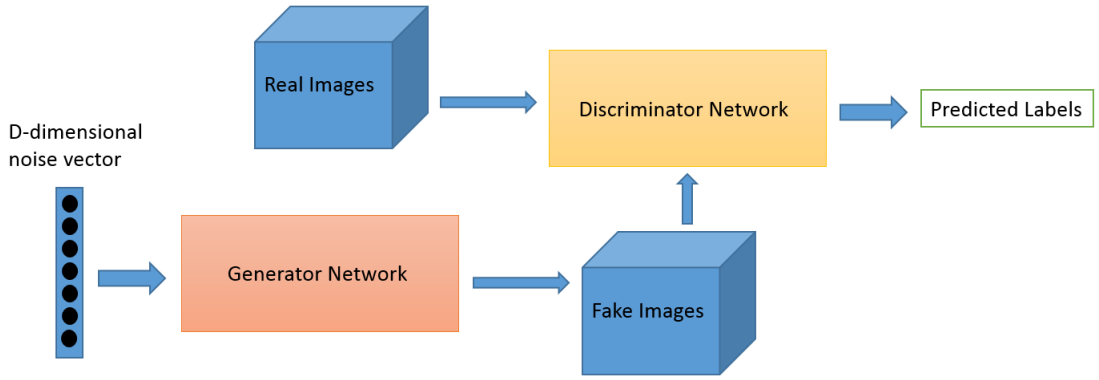

FIGURE 2.6 – Principe des GAN (Source : [Bruner and Deshpande, 2017])

.

non. Le modèle générateur recevrait en entrée des séquences de bruit. Toutefois, nous souhaitons transformer des séquences fluides en séquences disfluentes, pas les créer à partir de rien. C'est pourquoi, nous pouvons imaginer que nous ajouterions une seconde entrée à ce réseau. Cette entrée serait une séquence de mots fluides. Le modèle générateur devrait donc apprendre à transformer la séquence fluide en séquence disfluente. Tant que le réseau discriminateur arrivera à faire la différence entre les séquences du corpus et les séquences générées, le réseau générateur va modifier ses poids. En ne conservant que le modèle générateur, nous devrions être capable de transformer des séquences fluides en séquences disfluentes. Cependant, il est possible que celui-ci ne conserve pas totalement la séquence fluide originale si elle ne ressemble pas à celles présentes dans le corpus.

2.2.1.5 Mécanisme d'attention

Un réseau de neurones récurrent réalise des prédictions à partir de données séquentielles. Chaque élément apporte une part d'information que le réseau utilise pour produire une sortie. Toutefois, il est possible que certains éléments d'une séquence contiennent davantage d'informations utiles à la prédiction. Le mécanisme d'attention a pour objectif de repérer quels sont les éléments les plus utiles.

Par exemple, [Bahdanau et al., 2014] propose une architecture RNN de type encodeur-décodeur avec un mécanisme d'attention. Un encodeur-décodeur est un modèle en deux composantes qui permet de faire des correspondances entre une séquence d'entrée et une séquence de sortie. La partie encodeur est chargée de représenter la séquence d'entrée en un vecteur de taille fixée, tandis que la partie décodeur utilise cette représentation afin de générer une séquence en sortie. Cependant, tous les éléments de la séquence d'entrée n'ont pas la même importance pour obtenir la séquence de sortie. Le rôle du mécanisme d'attention est donc de donner de l'importance à la position des éléments au sein de la séquence d'entrée. Pour cela, on attribue un poids à chaque élément encodé en entrée. Ces poids sont ensuite mis à jour après chaque sortie produite. Le décodeur calcule alors un vecteur de contexte en faisant une somme, pondérée par ces poids, des éléments encodés. Ce vecteur de contexte permet au décodeur de prêter davantage attention à certains éléments pour produire sa sortie. À chaque sortie produite par le décodeur, le vecteur de contexte est recalculé.

2.2.2 Applications des réseaux de neurones séquence-à-séquence

Dans cette section, nous étudierons divers travaux réalisant des tâches séquence-à-séquence en traitement automatique du langage naturel à l'aide de réseaux de neurones. De cette façon, nous pourrons nous en inspirer en conservant leurs atouts.

Tout d'abord, les réseaux de neurones sont très utiles dans ce domaine car ils parviennent à capturer le sens des mots. [Mikolov et al., 2013] a mis en avant ce phénomène. Dans cet article, des phrases (séquence de mots) étaient mises en entrée d'un RNN classique. Les mots étaient encodés en *« one-hot »*, c'est-à-dire que le mot est représenté par un vecteur de dimension égale à la taille du dictionnaire. Ce vecteur contient des 0 partout et un 1 au niveau de la dimension correspondant au mot encodé. Cette représentation est souvent utilisée pour ce type de travaux. Le réseau de neurones essayait de prédire une distribution de probabilités des mots. En observant la sortie d'une couche intermédiaire, on obtient une représentation vectorielle de l'entrée. Les auteurs ont alors montré que cette représentation, appelée *« embedding »* (ou plongement), modélise les informations sémantiques et syntaxiques des mots. En effet, on remarque que le décalage (distance et orientation) entre le vecteur du mot *« man »* et celui de *« woman »* est le même qu'entre les mots *« king »* et *« queen »*. Les mêmes constatations sont faites pour les pluriels ou encore les conjugaisons.

Dans [Sutskever et al., 2014], les réseaux de neurones sont utilisés pour l'apprentissage séquence-à-séquence. Cet article propose en effet un modèle pour faire de la traduction automatique. Ce modèle a pour but d'estimer la probabilité qu'une séquence de mots soit la traduction correcte de la séquence de mots en entrée. Pour cela, il contient deux composantes. La première est un réseau LSTM avec quatre couches cachées qui prend en entrée la séquence de mots source et produit un vecteur représentant la séquence. La seconde est un LSTM simple qui prédit une traduction à partir de ce dernier vecteur. On retrouve donc la structure d'encodeur-décodeur. Le système commence par lire la séquence d'entrée mot par mot. Celle-ci se termine par un symbole qui en indique la fin. Une fois ce symbole lu, le système produit la séquence de sortie également mot par mot, en la terminant par un symbole de fin. Les auteurs de cet article ont remarqué que donner la séquence d'entrée à l'envers améliorait la qualité des traductions : au lieu de prédire que la séquence $\langle A,B,C \rangle$ se traduit par $\langle W,X,Y,Z \rangle$, le système était entraîné à prédire que $\langle C,B,A \rangle$ se traduit par $\langle W,X,Y,Z \rangle$.

Dans ce domaine, une grande partie des réseaux de neurones sont des LSTM. Toutefois, des variantes sont parfois utilisées. On peut évoquer les LSTM bidirectionnels (BLSTM) qui permettent de lire l'entrée en entier avant de produire une sortie. Pour cela, pendant qu'un LSTM lit les données (de gauche à droite), un autre LSTM les parcourt dans l'autre sens. La sortie produite peut être la concaténation des sorties de ces deux réseaux. Dans [Rao et al., 2015], un modèle complexe a été utilisé pour produire une conversion graphème (plus petite unité d'écriture) vers phonème (plus petite unité sonore permettant de distinguer deux mots dans une langue). Ce modèle utilise un LSTM et un BLSTM en parallèle, puis un second LSTM reçoit les sorties de ces deux réseaux pour prédire les phonèmes.

Dans [Hermann et al., 2015], plusieurs modèles ont été mis en compétition afin de réaliser une tâche de compréhension écrite. Chaque système recevait en entrée un texte (le contexte) ainsi qu'une question. Il devait prédire la réponse à la question grâce au contexte associé. Ces sytèmes renvoient la réponse qui maximise la probabilité que la réponse soit exacte sachant le contexte et la question. Les modèles les plus performants ont été « le lecteur attentif » et « le lecteur impatient ». Ces derniers utilisent un mécanisme d'attention pour repérer quelle partie du contexte est la plus à même de répondre à la question. Le « lecteur attentif » est un modèle comportant deux composants. Le premier composant est un LSTM bidirectionnel chargé d'encoder le contexte. Le second

est un autre LSTM bidirectionnel qui encode la question. Ensuite, une représentation de la paire contexte/question est produite à partir d'une somme pondérée des encodages et permet au système de prédire la réponse. Le « lecteur impatient » est semblable au précédent. Cependant, tandis que les deux encodages sont indépendants pour ce dernier, le « lecteur impatient » relit le contexte à la lecture de chaque élément de la question afin de produire un encodage propre à chacun de ceux-ci.

Comme le montre ces différents travaux, les réseaux de neurones sont utilisés avec succès pour traiter de nombreux problèmes de traitement automatique du langage naturel. Ces réseaux sont de différentes complexités et de différentes formes. On remarque toutefois que ce ne sont pas les seuls facteurs ayant un rôle dans la qualité des systèmes. On peut par exemple citer l'astuce de [Sutskever et al., 2014] qui consiste à inverser la séquence en entrée. Comme nous avons fait pour les disfluences, une bonne compréhension du phénomène que l'on veut modéliser peut en effet contribuer aux performances des systèmes.

2.3 Pistes envisagées

Dans cette revue bibliographique, nous avons cherché à mettre au jour les éléments utiles à la production automatique de disfluences. Nous avons ainsi pu voir que les disfluences possèdent des caractéristiques importantes pour la synthèse de discours spontanés naturels et qu'il est donc nécessaire d'apprendre à produire des textes disfluents. La production de textes disfluents pouvant être assimilée au passage d'une séquence de mots fluide à une séquence de mots disfluente, nous avons également étudié différents modèles séquence-à-séquence. De plus, nous avons porté un intérêt particulier aux réseaux de neurones car de nombreux problèmes issus du domaine du traitement automatique du langage naturel sont traités avec succès par ces modèles. Cependant, il n'existe pas encore de réseaux de neurones chargés de produire des textes disfluents. Cette tâche est partiellement traitée (uniquement pour certaines disfluences) par des modèles de langages et des modèles probabilistes. La méthode récemment proposée au sein de l'IRISA [Qader, 2017] utilise par exemple des modèles de langage et des CRF.

Le sujet de mon stage intervient dans ce contexte. Les réseaux de neurones étant efficaces pour d'autres problèmes proches, il est intéressant de les utiliser pour produire des textes disfluents. Un premier travail à effectuer durant ce stage sera de choisir le type précis, ainsi que la topologie du modèle neuronal à utiliser. À cause des dépendances entre les mots, l'utilisation de LSTM semble tout indiquée. Le travail de production de textes disfluents peut être vu comme un problème de traduction « texte fluide » (langage source) vers « texte disfluent » (langage cible). C'est pourquoi les modèles dédiés à la traduction automatique seront une bonne source d'inspiration. Il faudra aussi s'intéresser au jeu de données dont nous disposerons. En effet, il y aura sûrement un travail de préparation des données à réaliser. Selon ces dernières, il faudra peut-être les nettoyer de toutes disfluences afin de pouvoir entraîner le modèle neuronal. Cette tâche peut s'avérer compliquée en cas d'imbrications de disfluences. D'autres questions, comme la stratégie de production des disfluences, seront à prendre en considération. Il faudra en effet se demander s'il faut proposer un modèle par disfluence ou un modèle global, ou encore s'il faut se restreindre à certaines disfluences.

Dans la suite, la section 3 formalisera le problème de production de disfluences. La section 4 présentera les jeux de données dont nous disposons. Nous décrirons nos modèles dans la section 5. Enfin, les expériences réalisées et leurs résultats seront présentés dans la section 6.

3 Formalisation du problème

Afin de formaliser le problème, nous en présentons tout d'abord une modélisation. Cette modélisation décrit un cadre mathématique et théorique du problème. Nous proposons ensuite des métriques pour évaluer objectivement des séquences disfluentes.

3.1 Modélisation

Pour nos travaux, nous assimilons la production de disfluences à un problème de traduction automatique depuis un langage fluide vers un langage disfluent.

Nous notons $\mathcal{F}$ le langage fluide, c'est-à-dire l'ensemble des énoncés possibles ne contenant pas de disfluence, et $\mathcal{D}$ le langage disfluent. Nous nous concentrons sur trois types de disfluences : les pauses (P), les répétitions (R) et les révisions (RV). Nous décidons de regrouper les pauses silencieuses, les pauses remplies et les marqueurs de discours sous le terme générique de pauses. De même, nous considérons ici que les faux-départs sont des cas particuliers de révisions.

Nous avons vu précédemment que les trois types de disfluences qui nous intéressent ont chacune des caractéristiques particulières. C'est pourquoi, il peut être intéressant d'étudier la production de chaque type de disfluences séparément. Nous définissons donc des sous-langages de $\mathcal{D}$:

- nous appellons $\mathcal{D}_X$ un langage disfluent uniquement en terme de X, $X \in \{P, R, RV\}$, c'est-à-dire qu'il ne contient aucune disfluence de type autre que X.
- nous posons $\mathcal{D}_{X_1 \times X_2}$ avec $X_1, X_2 \in \{P, R, RV\}$ comme étant l'ensemble des séquences ne contenant aucune disfluence de type autre que X_1 ou X_2.
- nous définissons $\mathcal{D}_{X_1 - X_2}$ tel que $s \in \mathcal{D}_{X_1 - X_2} \Leftrightarrow s \in \mathcal{D}_{X_1} \wedge s \notin \mathcal{D}_{X_2}$.
- nous pouvons noter que $\mathcal{F} = \mathcal{D}_\emptyset$ et que $\forall X \in \{P, R, RV\}, \mathcal{F} \subset \mathcal{D}_X$.

Le problème de traduction correspond à la transformation d'une séquence $(a_i)_{i \in [\![1,n]\!]}$ de mots fluides en une autre séquence $(b_j)_{j \in [\![1,m]\!]}$ de mots disfluents en termes de X de longueur supérieure (les disfluences considérées ne font qu'ajouter du contenu). Cette transformation, que nous nommons Δ_X, est modélisée par :

$$\begin{aligned} \Delta_X : \quad & \mathcal{F} \to \mathcal{D}_X \\ & (a_i)_{i \in [\![1,n]\!]} \mapsto (b_j)_{j \in [\![1,m]\!]} \end{aligned} \tag{3.1}$$

avec n la longueur de la séquence fluide, m celle de la séquence disfluente. Cette transformation doit respecter plusieurs propriétés :

- préservation : la séquence disfluente produite est associée à la séquence fluide en entrée. Par exemple, on peut avoir :

  Le chat regarde le chien. $\overset{\Delta_{P+R}}{\to}$ Le chat regarde le *euh* le chien.

  On remarque que la séquence de mots initiale se retrouve dans celle produite. Pour toute séquence $B = (b_j)_{j \in [\![1,m]\!]} \in \mathcal{D}_X$, il existe une séquence non vide $A = (a_i)_{i \in [\![1,n]\!]}$ telle que A est une sous-séquence de B et A ne contient pas de disfluence.
  Formellement, on a :

$$\begin{aligned} \forall (i_1, i_2) \text{ tel que } i_1 < i_2, \exists (j_1, j_2) \text{ tel que } j_1 < j_2 \text{ et } b_{j_1} = a_{i_1}, b_{j_2} = a_{i_2} \\ \text{avec } n \leq m, A \in \mathcal{F} \text{ et } B \in \mathcal{D}_X \end{aligned} \tag{3.2}$$

- conformité : les disfluences produites doivent respecter la structure et les propriétés de disfluences que nous avons présentées dans l'étude bibliographique (section 2.1.1). Ces propriétés

prennent des formes différentes selon le type de disfluence : les pauses produites doivent appartenir au vocabulaire des pauses (certains mots ne peuvent pas constituer une pause), les répétitions doivent avoir un *reparandum* identique à la réparation et enfin, le *reparandum* d'une révision doit corriger sa réparation (ces deux parties doivent être différentes).

En plus de ces propriétés, nous émettons des hypothèses de crédibilité :

— il existe des positions de préférences pour insérer les disfluences. Ces positions dépendent du type de disfluence et de son contexte.

— les disfluences insérées doivent avoir un sens dans leur contexte. Le contexte d'une disfluence joue un rôle sur le contenu de la disfluence.

Afin de représenter un mot d'une séquence, nous associons à chaque mot son type t de disfluence. Celui-ci est tel que $t \in \{-, P_B, P_I, R_B, R_I, RV_B, RV_I\}$ avec « $-$ » désignant un mot fluide, P, R, RV désignant respectivement un mot faisant partie d'une pause, d'une répétition ou d'une révision. Nous utilisons le codage BIO (pour *Beginning-Inside-Outside*) : les types indicés B marquent le début de la disfluence et ceux indicés I indiquent que le mot est dans la disfluence. Par exemple, on a l'annotation suivante :

Le	le	chat	le	chat	regarde	un	euh	le	chien.
R_B	R_B	R_I	$-$	$-$	$-$	RV_B	P_B	$-$	$-$

De plus, nous décidons d'intégrer l'étiquette morpho-syntaxique (POS) à la représentation d'un mot en entrée. Cela fournit ainsi une information supplémentaire pour traiter le problème. Nous choisissons également de traiter un problème multi-tâches : nous souhaitons que les mots produits contiennent, en plus du type de disfluence, le POS associé au mot. Nous avons vu que certains modèles neuronaux produisaient de meilleurs résultats lorsqu'ils étaient entraînés à prédire plusieurs informations complémentaires. Nous pouvons donc compléter l'équation 3.1 par :

$$\begin{aligned} \Delta_X : \quad & \mathcal{F} \to \mathcal{D}_X \\ & (w_i, t_i, p_i)_{i \in [\![1,n]\!]} \mapsto (w'_j, t'_j, p'_j)_{j \in [\![1,m]\!]} \end{aligned} \tag{3.3}$$

avec w_i (resp. w'_j) les mots de la séquence en entrée (resp. de la séquence produite), t_i (resp. t'_j) les types de disfluences et p_i (resp. p'_j) les POS associés aux mots.

3.2 Métriques

La production de disfluences étant une tâche relativement peu étudiée, il n'existe pas de mesure d'évaluation spécifique. Nous définissons donc des métriques pour tenter d'analyser et comparer nos résultats.

Nous distinguons trois objectifs de nos modèles que nous souhaitons évaluer. Ceux-ci font échos aux propriétés et aux hypothèses présentées dans la section 3.1 : nos modèles doivent être capable de conserver la séquence initiale dans la séquence produite, de produire des disfluences grammaticalement correctes et crédibles.

Avant de détailler ces trois points, nous allons présenter des fonctions outils :

— $\Phi_Y = \Delta_Y^{-1}$ est la réciproque de Δ_Y. Cette fonction enlève les disfluences de type Y d'une séquence $a\delta_Y b \in \mathcal{D}_X$ telle que $\mathcal{D}_Y \subset \mathcal{D}_X$. La transformation est la suivante :

$$\begin{aligned} \Phi_Y : \mathcal{D}_X &\to \mathcal{D}_{X-Y} \\ a\delta_Y b &\mapsto \Phi_Y(a)\Phi_Y(b) \end{aligned} \tag{3.4}$$

avec δ_Y une disfluence de type Y, a et b sont des séquences de mots pouvant, elles-aussi, être disfluentes. Par exemple :

Le	le	chat	regarde	un	un	le	chien.	$\overset{\Phi_{RV}}{\rightarrow}$	Le	le	chat	regarde	le	chien.
R_B	–	–	–	R_B	RV_B	–	–		R_B	–	–	–	–	–

Il est important de noter qu'en supprimant la révision, la répétition qui la précédait a été supprimée également. En effet, la partie répétée ayant été enlevée, la répétition n'a plus de sens.

— Π_Y, qui retourne la liste des points d'interruption de type Y d'une séquence.

— T, qui remplace les disfluences d'une séquence par un symbole correspondant à son type. Par exemple, T(« Le chat chat dort. ») donne « Le <R> chat dort. ».

3.2.1 Conservation de l'information initiale

Nous souhaitons rendre des phrases disfluentes, pas en produire à partir de rien. Comme nous l'avons remarqué dans la section 3.1, les modèles doivent conserver l'information initiale (c'est-à-dire la séquence de mots en entrée du modèle) dans la séquence produite. Nous proposons deux mesures pour évaluer cette capacité. Ces mesures sont fondées sur le dénombrement des différences entre deux séquences x et y. Nous définissons trois fonctions de dénombrement :

— $D(x, y)$ compte le nombre de mots de y que l'on ne retrouve pas dans x (mots supprimés).

— $S(x, y)$ compte le nombre de mots de y qui ont été modifiés dans x.

— $I(x, y)$ compte le nombre de mots de x qui n'étaient pas dans y (mots ajoutés).

Ces trois fonctions se calculent après alignement des séquences x et y. Notons que nous n'effectuons qu'un seul alignement pour toutes ces fonctions.

La première mesure se nomme « préservation ». Comme son nom l'indique, elle évalue à quel point la séquence initiale est conservée dans la séquence finale. On mesure le taux de mots de la séquence source qui ne sont ni supprimés, ni modifiés dans la séquence produite. Ainsi, pour un couple de séquences (p, s) tel que $p = \Delta_X(s)$ avec $p \in \mathcal{D}_X, s \in \mathcal{F}$, la préservation se calcule selon :

$$\begin{array}{rccl} \text{préservation :} & \mathcal{D}_X \times \mathcal{F} & \rightarrow & [0,1] \\ & (p,s) & \mapsto & 1 - \frac{S(\Phi_X(p),s)+D(\Phi_X(p),s)}{\text{Card}(s)} \end{array} \tag{3.5}$$

La préservation est idéale lorsqu'elle vaut 1. Les séquences sont nettoyées de leurs disfluences car cette mesure n'évalue pas les disfluences produites.

La seconde mesure est la « distorsion ». Cette mesure évalue la dégradation de l'information initiale, c'est-à-dire à quel point celle-ci est perdue ou modifiée. On mesure le taux de mots de la séquence source qui sont supprimés, modifiés ou ajoutés dans la séquence produite. Pour un couple de séquences (p, s) tel que $p = \Delta_X(s)$ avec $p \in \mathcal{D}_X, s \in \mathcal{F}$, la distorsion se calcule par :

$$\begin{array}{rccl} \text{distorsion :} & \mathcal{D}_X \times \mathcal{F} & \rightarrow & [0,+\infty] \\ & (p,s) & \mapsto & \frac{S(\Phi_X(p),s)+D(\Phi_X(p),s)+I(\Phi_X(p),s)}{\text{Card}(s)} \end{array} \tag{3.6}$$

Contrairement à la préservation, nous prenons en compte les mots qui ont été ajoutés par le modèle. Il est à noter que la valeur obtenue n'est pas bornée : si le modèle ajoute énormément de mot, la distorsion peut dépasser 1. La distorsion idéale est égale à 0. Cette mesure correspond au *WER* (pour *Word Error Rate*) et est souvent utilisée en reconnaissance de la parole.

La préservation et la distorsion sont corrélées. Plus la préservation est faible, plus la distorsion est élevée ($1 -$ préservation $\leq$ distorsion). Toutefois, il est possible d'avoir une préservation parfaite et une distorsion élevée lorsque le modèle génère des mots fluides en plus de ceux de la séquence initiale.

3.2.2 Conformité des disfluences

Les modèles doivent produire des disfluences grammaticalement correctes, c'est-à-dire correctement formées. En effet, les pauses produites doivent appartenir au vocabulaire des pauses, le *reparandum* des répétitions doit répéter la réparation et le *reparandum* des révisions doit corriger la réparation. Pour évaluer si une disfluence est correctement formée, nous proposons la mesure « conformité ». Soit une disfluence δ contenue dans la séquence $A = a\delta b$, avec $\delta \in \mathcal{D}_X, A \in \mathcal{D}_Y$ et $\mathcal{D}_X \subset \mathcal{D}_Y$. La conformité de δ est donnée par :

$$\begin{aligned} \text{conformité} : \quad & \mathcal{D}_X \times \mathcal{D}_Y \to \{0,1\} \\ & \delta, a\delta b \mapsto \begin{cases} 1 & \text{si } X = P \text{ et } \delta \text{ appartient au vocabulaire des pauses,} \\ & \text{ou si } X = R \text{ et } \forall i, \delta_i = \Phi_P(b)_i, \\ & \text{ou si } X = RV \text{ et } \exists i, \delta_i \neq \Phi_P(b)_i, \\ 0 & \text{sinon.} \end{cases} \end{aligned} \tag{3.7}$$

Le résultat de cette mesure est binaire (soit la disfluence est bien formée, soit elle ne l'est pas). De cette façon, tous les types de disfluences sont évalués de manière identique. Pour les révisions, l'évaluation est très permissive : il suffit que la révision ne soit pas une répétition pour être considérée comme correcte. Il est en effet très difficile d'évaluer objectivement si une révision est correcte car le *reparandum* d'une révision peut ne pas avoir la même structure syntaxique que sa réparation et pourtant être correcte (cas des faux-départs par exemple). C'est pour des cas comme celui-là que les évaluations subjectives sont particulièrement pertinentes.

3.2.3 Crédibilité des disfluences

Les disfluences prédites par les modèles doivent être crédibles. Nous distinguons plusieurs niveaux de crédibilité :

— crédibilité quantitative : le nombre de disfluences par mot fluide est crédible
— crédibilité positionnelle : les disfluences sont insérées au bon endroit
— crédibilité sémantique : le contenu des disfluences doit être cohérent avec le reste de la séquence

3.2.3.1 Crédibilité quantitative

Nous voulons que le nombre de disfluences par mot fluide soit crédible, c'est-à-dire qu'il corresponde à celui dans la référence. Soient X le type de disfluences que nous souhaitons produire, $A = (a_1, \ldots, a_n)$ l'ensemble des séquences prédites par le modèle que l'on évalue et $B = (b_1, \ldots, b_n)$ l'ensemble des séquences servant de références. On a $\forall i \in [\![1, n]\!], a_i \in \mathcal{D}_X$ et $b_i \in \mathcal{D}_X$. Nous posons la fonction $\Theta_X(A)$ qui compte le nombre de points d'interruption par mot dans l'ensemble de

séquences A de la manière suivante :

$$\Theta_X(A) = \frac{\sum_{i=1}^{n} card(\Pi_X(a_i))}{\sum_{i=1}^{n} card(\Phi_X(a_i))} \tag{3.8}$$

Nous définissons alors la mesure IRR (pour *Interruption Rate Ratio*) qui fait le rapport entre le nombre de points d'interruption par mot de A et le nombre de points d'interruption par mot de B. L'IRR se calcule donc comme :

$$\mathrm{IRR}_X(A, B) = \frac{\Theta_X(A)}{\Theta_X(B)} \tag{3.9}$$

Notons que $\Theta_X(\mathbf{B})$ est toujours strictement positif car la référence contient au moins une disfluence. L'IRR indique si la quantité de disfluences dans les séquences prédites est semblable à celle dans les séquences de référence. Un modèle sera bon si cette mesure est proche de 1.

3.2.3.2 Crédibilité positionnelle

Nous cherchons maintenant à évaluer si les disfluences sont insérées au bon endroit. Pour cela, nous alignons automatiquement les mots fluides des séquences prédites et ceux des références et comparons la position des points d'interruption : nous désignons par « PI vrais positifs » le nombre de PI prédits à la même position que dans la référence, par « PI positifs » le nombre de PI prédits et par « PI vrais » le nombre de PI de la référence. Le **rappel** mesure le nombre de vrais positifs (éléments prédits vrais qui sont vrais) divisé par le nombre de positifs (éléments prédits vrais) tandis que la **précision** mesure le nombre de vrais positifs divisé par le nombre d'éléments qui sont vrais (qu'ils aient été prédits comme tels ou pas). Nous proposons d'utiliser la mesure F_1 qui prend compte à la fois du rappel et de la précision de la façon suivante :

$$F_1 = \frac{2 \times \text{précision} \times \text{rappel}}{\text{précision} + \text{rappel}} \tag{3.10}$$

Toutefois, cette évaluation est naïve car nous considérons qu'il n'existe qu'une seule position valable pour insérer les disfluences alors que, lors de notre étude bibliographique, nous avons vu que cela était faux. Soit δ une disfluence contenue dans la séquence $A = a\delta b$ avec $\delta \in \mathcal{D}_X, A \in \mathcal{D}_Y$ et $\mathcal{D}_X \subset \mathcal{D}_Y$, a et b pouvant également contenir des disfluences. Nous avons réfléchi à une série de mesures qui a pour but d'évaluer la crédibilité de δ.

— la mesure CP (pour Crédibilité Positionnelle) évalue la crédibilité que le type de la disfluence δ soit à un endroit précis au sein de la séquence A en prenant en compte la probabilité de la séquence sans cette disfluence :

$$\mathrm{CP}(A, \delta) = \frac{\mathbb{P}(T(a\delta b))}{\mathbb{P}(T(ab))} \tag{3.11}$$

— la mesure $\widehat{\mathrm{CP}}$ quant à elle, prend en compte la probabilité du type de la disfluence évaluée. De cette façon, nous enlevons le biais qu'il peut y avoir si un type de disfluence est surreprésenté :

$$\widehat{\mathrm{CP}}(A, \delta) = \frac{\mathrm{CP}(A, \delta)}{\mathbb{P}(T(\delta))} \tag{3.12}$$

— les mesures CPF (pour Crédibilité Positionnelle Fluide) et $\widehat{\text{CPF}}$ sont des variantes des deux mesures précédantes. Avec ces mesures, nous évaluons les probabilités comme si la disfluence évaluée était la seule de la séquence, afin de limiter le biais qu'il y aurait si les autres disfluences de la séquence étaient peu crédibles :

$$\text{CPF}(A, \delta) = \frac{\mathbb{P}(\Phi_\Omega(a)T(\delta)\Phi_\Omega(b))}{\mathbb{P}(\Phi_\Omega(ab))} \tag{3.13}$$

$$\widehat{\text{CPF}}(A, \delta) = \frac{\text{CPF}(A, \delta)}{\mathbb{P}(T(\delta))} \tag{3.14}$$

Il est difficile de donner une interprétation aux valeurs que ces mesures fournissent. C'est pourquoi nous les utiliserons en comparaison avec les références : nous ferons le ratio de chaque mesure sur les disfluences produites par chaque mesure sur les disfluences références. Nous ne pouvons pas dire à partir de quel ratio un modèle est acceptable, nous nous servirons de ces mesures uniquement pour comparer les modèles entre eux.

3.2.3.3 Crédibilité sémantique

Enfin, nous voulons savoir si les disfluences ont du sens par rapport à leur contexte, c'est-à-dire si leur contenu est cohérent avec le reste de la séquence. Nous faisons la même remarque que précédemment : évaluer la sémantique objectivement n'est pas évident, le mieux pour cette tâche étant l'évaluation subjective. Toutefois, nous tentons d'avoir un aperçu de cette crédibilité en proposant des adaptations des quatre mesures présentées auparavant. Ces adaptations consistent à évaluer la séquence produite directement, sans remplacer la disfluence par un symbole correspondant à son type. Pour δ une disfluence contenue dans la séquence $A = a\delta b$ avec $\delta \in \mathcal{D}_X, A \in \mathcal{D}_Y$ et $\mathcal{D}_X \subset \mathcal{D}_Y$, a et b pouvant également contenir des disfluences, nous avons donc :

— les mesures CS (pour Crédibilité Sémantique) et $\widehat{\text{CS}}$, qui sont les équivalents de CP et $\widehat{\text{CP}}$:

$$\text{CS}(A, \delta) = \frac{\mathbb{P}(a\delta b)}{\mathbb{P}(ab)} \tag{3.15}$$

$$\widehat{\text{CS}}(A, \delta) = \frac{\text{CS}(A, \delta)}{\mathbb{P}(\delta)} \tag{3.16}$$

— les mesures CSF (pour Crédibilité Sémantique Fluide) et $\widehat{\text{CSF}}$, qui sont les équivalents de CPF et $\widehat{\text{CPF}}$:

$$\text{CSF}(A, \delta) = \frac{\mathbb{P}(\Phi_\Omega(a)\delta\Phi_\Omega(b))}{\mathbb{P}(\Phi_\Omega(ab))} \tag{3.17}$$

$$\widehat{\text{CSF}}(A, \delta) = \frac{\text{CSF}(A, \delta)}{\mathbb{P}(\delta)} \tag{3.18}$$

Bien que ces mesures aient pour but d'évaluer si le contenu des disfluences est cohérent avec leur contexte, elles évaluent également la position des disfluences, ce qui constituent un biais. Nous pouvons également relever que, tout comme pour les précédentes mesures, les résultats absolus sont difficiles à interpréter.

Soit la séquence $A =$ « Le chat joue avec le *euh* avec le chien. ». L'évaluation de la disfluence $\delta =$ « *euh* » pour chacune de ces mesures est donnée pour exemple dans la table 3.1.

Mesure	Calcul
CP	$\mathbb{P}$(Le chat joue <R> <P> avec le chien.)/$\mathbb{P}$(Le chat joue <R> avec le chien.)
$\widehat{\text{CP}}$	$\mathbb{P}$(Le chat joue <R> <P> avec le chien.)/$\mathbb{P}$(Le chat joue <R> avec le chien.)$\mathbb{P}$(<P>)
CPF	$\mathbb{P}$(Le chat joue <P> avec le chien.)/$\mathbb{P}$(Le chat joue avec le chien.)
$\widehat{\text{CPF}}$	$\mathbb{P}$(Le chat joue <P> avec le chien.)/$\mathbb{P}$(Le chat joue avec le chien.)$\mathbb{P}$(<P>)
CS	$\mathbb{P}$(Le chat joue avec le euh avec le chien.)/$\mathbb{P}$(Le chat joue avec le avec le chien.)
$\widehat{\text{CS}}$	$\mathbb{P}$(Le chat joue avec le euh avec le chien.)/$\mathbb{P}$(Le chat joue avec le avec le chien.)$\mathbb{P}$(euh)
CSF	$\mathbb{P}$(Le chat joue euh avec le chien.)/$\mathbb{P}$(Le chat joue avec le chien.)
$\widehat{\text{CSF}}$	$\mathbb{P}$(Le chat joue euh avec le chien.)/$\mathbb{P}$(Le chat joue avec le chien.)$\mathbb{P}$(euh)

TABLE 3.1 – Calculs effectués par les mesures de crédibilité pour un exemple.

4 Corpus utilisés

Dans cette section, nous décrirons les deux corpus que nous avons à notre disposition. Tout d'abord, nous présentons un corpus réel : RATP-DECODA. Ensuite, nous présenterons des données artificielles qui nous seront utiles pour paramétrer nos modèles et pour les évaluer sur des données plus simples. L'objectif des modèles neuronaux sera de reproduire les disfluences présentes dans ces corpus.

4.1 Données RATP-DECODA

Le corpus RATP-DECODA est constitué de plus de 60 heures de parole retranscrites manuellement. Celle-ci est découpée en 1487 conversations téléphoniques collectées par le centre d'appel de la RATP à Paris. Ce sont donc des données issues de parole spontanée naturelle : en particulier, elles contiennent des disfluences. Le nombre total de locuteurs n'est pas connu mais plusieurs interlocuteurs interviennent pour chaque conversation (de 2 à 4). Ces conversations mettent en scène un client demandant de l'aide à son interlocuteur et selon l'aide demandée, ce dernier peut faire appel à un ou plusieurs collègues. Les sujets traités sont divers. Ils vont de la demande d'itinéraires ou d'informations sur le trafic jusqu'à la recherche d'un objet perdu. Chaque conversation est anonymisée et divisée en séquences de mots selon deux critères : le changement de locuteur et lorsque la durée entre deux mots consécutifs est strictement supérieure à 0.01 seconde. Notons que les séquences n'ont pas de ponctuation. Chaque mot possède de multiples annotations (certaines manuelles, d'autres automatiques) :

— le type de disfluence (marqueur de discours, greffe, répétition, troncature, fluide).
— l'étiquette morpho-syntaxique (POS).
— l'entité nommée associée au mot (au format BIO).
— lien de dépendance, c'est-à-dire à quel autre mot de la séquence ce mot est relié.
— le lemme correspondant au mot.
— diverses informations temporelles (temps de début, de fin,...).

— le locuteur, décrit par un identifiant propre à la conversation. Un identifiant identique dans deux conversations ne signifie pas que la même personne parle dans ces deux conversations.

Nous avons décidé de conserver uniquement le mot, le POS et le type de disfluence. En effet, nous ne nous intéressons pas à la synthèse audio des disfluences donc les informations temporelles ne semblent pas pertinentes. Les autres annotations peuvent être envisagées mais nous souhaitons aborder le problème le plus simplement possible.

Plusieurs pré-traitements ont été réalisés sur ces données. Tout d'abord, nous avons normalisé le corpus car les retranscriptions n'étaient pas cohérentes. Par exemple, on retrouve les séquences « c'est avec un b comme Bernard » et « ça s'écrit B comme Béatrice ». Le mot « b » est écrit en minuscule dans la première séquence et en majuscule dans le seconde. Nous avons donc converti toutes les majuscules en minuscules afin d'uniformiser les retranscriptions. Il y avait un autre type d'incohérence concernant les mots composés. L'expression « est-ce que » est quelques fois retranscrite en une seule expression et d'autres fois décomposée en trois mots (« est »,« ce » et « que »). Nous avons donc choisi de séparer tous les mots composés. De plus, cela résout un problème de mots composés lié, cette fois, aux disfluences : les répétitions partielles de mots composés. Par exemple, dans la séquence « la rue Ledru Ledru-Rollin », le mot « Ledru » ne répète pas l'expression « Ledru-Rollin » et serait classé en répétition erronée. En séparant les mots composés, on détecte bien que le mot « Ledru » est répété.

D'autres pré-traitements concernent la découpe des séquences. Notre objectif est de faire coïncider au maximum les séquences avec les phrases prononcées. Dans certaines conversations, les interlocuteurs parlent en même temps. À cause de cela, certains segments de phrases prononcés par l'un des interlocuteurs peuvent être retranscrits au milieu de la phrase d'un autre interlocuteur. Pour éviter ces situations, nous choisissons de supprimer les séquences multi-locuteurs. Un deuxième problème est la durée à partir de laquelle deux mots consécutifs sont considérés comme appartenant à deux séquences différentes. Celle-ci est de 0.01 seconde et est trop stricte. En effet, une phrase peut être découpée en plusieurs séquences car le locuteur produisait une pause silencieuse. Nous avons donc recollé ces séquences en ajoutant une pause au milieu, identifiée par un symbole *<silence>*. Enfin, il arrive que plusieurs phrases soient regroupées en une seule séquence. En nous aidant des annotations, nous avons séparé les sous-séquences n'ayant aucun lien de dépendance entre elles.

Les derniers pré-traitements portent sur les disfluences. Nous nous concentrons uniquement sur les pauses, les répétitions et les révisions. Nous avons donc supprimé les troncatures. De plus, nous avons remarqué que les mots annotés en tant que « greffe » sont en fait des marqueurs de discours. C'est pourquoi, nous avons changé l'étiquette des greffes, ainsi que des marqueurs de discours en pauses. Nous avons également étiqueté les disfluences avec le format BIO. Le corpus n'est pas annoté en révision. Toutefois, lorsque l'on étudie les répétitions de corpus, nous remarquons que certaines d'entre elles sont erronées : le *reparandum* ne répète pas la réparation mais le modifie, ce sont donc des révisions. Nous avons annoté automatiquement les répétitions du corpus et comparé nos annotations à celles du corpus. Nous avons alors appliqué le processus suivant :

— les mots annotés en répétition de base, confirmés par notre annotation, conservent leur annotation ;
— les mots annotés en répétition de base, que notre annotation ne repère pas, sont annotés en révisions ;
— les mots que l'on annote en répétition mais qui n'étaient pas déjà annotés en tant que tel, ne sont pas annotés (pour éviter de confondre les phrases telles que *« vous vous êtes perdus »* avec une répétition).

Nombre de séquences	35 679
Taille du vocabulaire	7 466
Longueur moyenne des séquences	12.00
Nombre de mots	428 199
Nombre de mots fluides	372 542
Nombre de sections annotées en pauses*	42 669
Nombre de sections annotées en répétitions*	6 419
Nombre de sections annotées en révisions*	1 933

TABLE 4.1 – Caractéristiques du corpus RATP-DECODA. * : une section disfluente peut contenir plusieurs mots

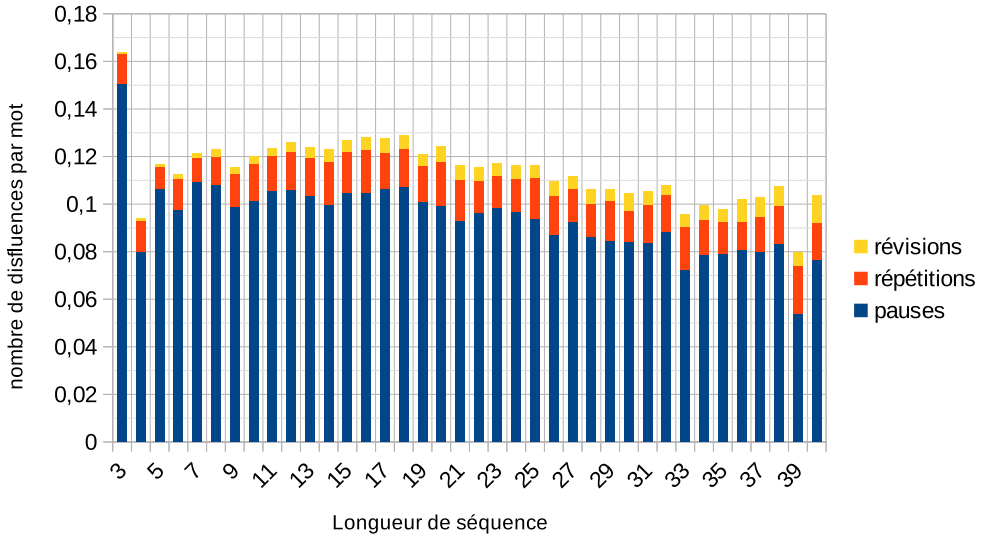

FIGURE 4.1 – Répartition des disfluences dans le corpus RATP-DECODA.

Cette solution n'est pas parfaite car beaucoup de révisions ne sont pas annotées. Toutefois, réaliser l'annotation à la main aurait été trop chronophage. Ne souhaitant pas étudier des séquences trop longues, nous avons conservé uniquement les séquences disfluentes de moins de 40 mots (115 séquences contenaient plus de 40 mots). De même, nous n'avons pas gardé les séquences qui, une fois les disfluences enlevées, avaient une longueur nulle.

Au final, 35 679 séquences disfluentes sont considéres, avec les caractéristiques décrites dans la table 4.1. Le vocabulaire contient plus de 7 000 mots différents, soit 1/5 du nombre de séquences, ce qui est conséquent. Les disfluences ne sont pas présentes à parts égales. En effet, les pauses sont sur-représentées, alors que les répétitions et les révisions sont moins présentes. Cela est confirmé par la figure 4.1. Cette dernière montre également que le taux de disfluences par mot est relativement constant en fonction de la longueur des séquences. Il y a en moyenne 0.11 disfluences par mot fluide et donc les séquences de plus de $n \times 10$ mots comportent souvent au moins n disfluences. En étudiant la répartition des séquences donnée par la figure 4.2, nous relevons que la majorité des séquences sont de longueurs courtes. Celles-ci proviennent en majorité de l'assistant de la RATP qui demande des précisions. En effet, les séquences de l'appelant sont plus longues car celui-ci expose

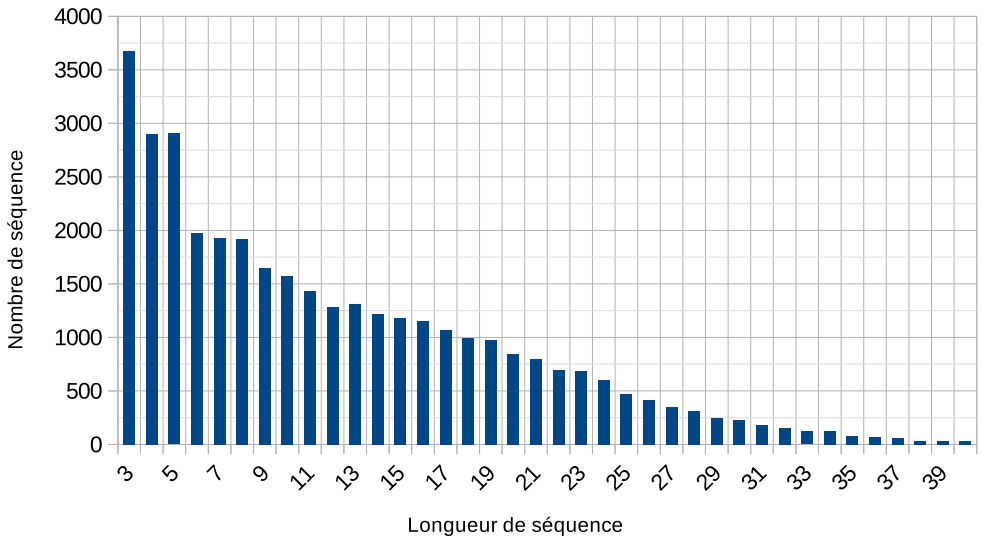

FIGURE 4.2 – Répartition des séquences issues du corpus RATP-DECODA.

Version disfluente après pré-traitements	**Version fluide**
de à la station volontaire où j' habite rue RV_B dargue <*silence*> et sur l' estimation euh on P_B P_B est déjà à neuf minutes	à la station volontaire où j' habite rue dargue et sur l' estimation on est déjà à neuf minutes
c' est c' est jusque on ne sait pas R_B R_I à quelle quand se termine le marathon	c' est jusque on ne sait pas à quelle quand se termine le marathon

TABLE 4.2 – Exemples de séquences du corpus RATP-DECODA.

son problème, la raison de son appel.

Ces séquences constituent donc nos données disfluentes. Afin de réaliser l'apprentissage, nous avons besoin d'une version fluide de ce corpus. Pour cela, nous utilisons la fonction Φ_Y présentée dans la section 3.2. Nous en profitons également pour produire des versions du corpus comportant uniquement certains types de disfluences. Des exemples de séquences disfluentes et leurs équivalents fluides sont présentées dans la table 4.2. Le deuxième exemple confirme que beaucoup de révisions ne sont pas annotées. En effet, « c'est jusque » est un faux-départ et « à quelle » est une révision de « quand ».

4.2 Données artificielles

Nous souhaitons disposer de données plus simples pour paramétrer nos modèles et pouvoir évaluer les résultats manuellement. Pour cela, nous avons créé des données artificielles, moins complexes que les données réelles.

Nous avons commencé par définir une grammaire hors contexte probabiliste avec un vocabulaire significativement plus petit (taille 22). Cette grammaire est définie par la figure 4.3, les probabi-

```
S     →  NP VP [0.60]  |  S1 S2 [0.40]
S1    →  NP VP [0.97]  |  S1 S2 [0.03]
S2    →  LIA S1 [1.0]
NP    →  PropN [0.6]   |  GN [0.4]
GN    →  DET N [0.7]   |  DET AN [0.3]
AN    →  A N [1.0]
VP    →  VS [0.3]      |  V NP [0.7]
V     →  VS [0.4]      |  VC PREP  [0.6]
PropN →  'Pierre' [0.4] |  'Paul' [0.3]      |  'Jacques' [0.3]
DET   →  'le' [0.4]     |  'un' [0.4]        |  'son' [0.2]
N     →  'chat' [0.3]   |  'chien' [0.3]     |  'lezard' [0.2]      |  'dragon' [0.2]
A     →  'petit' [0.7]  |  'gentil' [0.3]
VS    →  'regarde' [0.4] |  'aime' [0.3]     |  'protege' [0.3]
VC    →  'joue' [0.3]   |  'mange' [0.3]     |  'travaille' [0.4]
PREP  →  'avec' [1.0]
LIA   →  'car' [0.35]   |  'lorsque' [0.35]  |  'tandis que' [0.3]
```

FIGURE 4.3 – Grammaire hors contexte utilisée pour la génération des données artificielles.

Version fluide	**Version disfluente**
Pierre regarde Jacques car Paul – – – – – regarde Jacques – –	Pierre protège enfin regarde Jacques car – $_{RV_B}$ $_{P_B}$ – – – Paul regarde Jacques regarde Jacques – $_{R_B}$ $_{R_I}$ – –
le chat protège le chien – – – – –	le chat protège euh le chien – – – $_{P_B}$ – –

TABLE 4.3 – Exemples de séquences du corpus Arti.

lités sont écrites entre crochets. Elle respecte la structure classique du français SUJET-VERBE [-COMPLÉMENT] (le complément est optionnel) et il y a une probabilité pour qu'une fois la séquence générée, on génère à nouveau un SUJET-VERBE [-COMPLÉMENT] en ajoutant un mot de liaison entre les deux. Cela permet de produire des séquences de longueur variable.

Nous avons généré ainsi 50 000 séquences fluides. Afin de réaliser l'apprentissage, nous avons produit une version disfluente de ces séquences. Pour cela, à chaque mot fluide, nous avons associé une probabilité pour chaque type de disfluence. Certains mots avaient une chance élevée (20%) d'avoir un certain type de disfluence (mots déclencheurs), d'autres une probabilité quasi-nulle (0.5%) et d'autres une probabilité moyenne (5%). Les pauses et les révisions générées ne comportent qu'un seul mot, mais les répétitions sont de longueur variable.

De cette manière, nous avons créé un corpus que nous nommons « Arti ». Des exemples de séquences fluides du corpus Arti, ainsi que leurs versions disfluentes, sont présentés dans la table 4.3. Nous souhaitons également évaluer comment le taux de disfluences du corpus peut influencer l'apprentissage. Nous avons donc produit une seconde version de ce corpus, nommée « Arti+ », qui contient davantage de disfluences. Les caractéristiques des versions disfluentes de ces corpus sont décrites dans la table 4.4. La taille du vocabulaire est de 25 (la différence avec le vocabulaire fluide provient du vocabulaire des pauses que l'on produit). Le nombre de mots fluides est similaire à celui des données réelles mais le nombre de disfluences est différent. En effet, les deux corpus artificiels

	Arti	**Arti+**
Nombre de séquences	50 000	50 000
Taille du vocabulaire	25	25
Longueur moyenne des séquences	7.50	11.16
Nombre de mots	374 913	557 807
Nombre de mots fluides	306 231	306 231
Nombre de sections annotées en pauses*	14 531	73 856
Nombre de sections annotées en répétitions*	15 729	50 124
Nombre de sections annotées en révisions*	17 174	59 842

TABLE 4.4 – Caractéristiques des corpus artificiels. * : une section disfluente peut comporter plusieurs mots

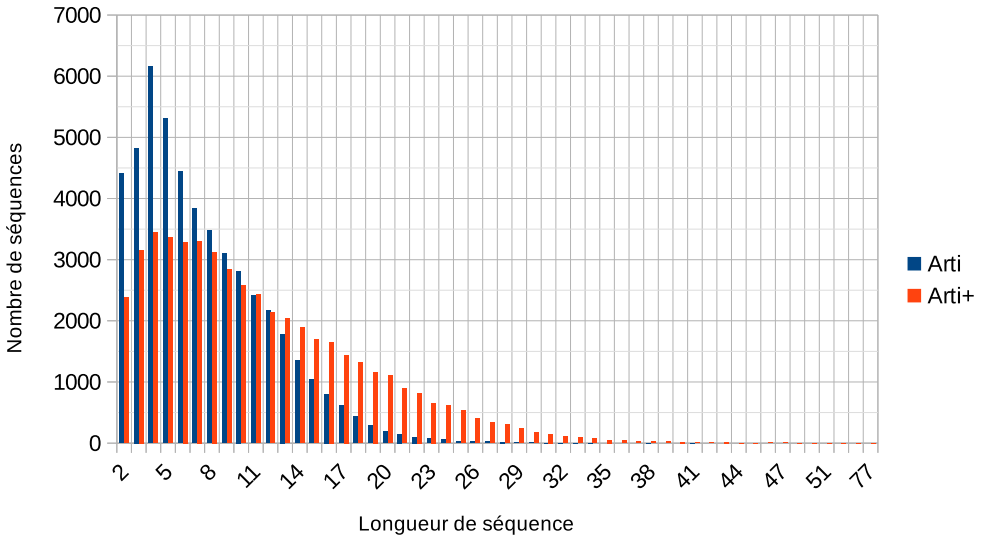

FIGURE 4.4 – Répartition des séquences issues des corpus artificiels.

possèdent davantage de disfluences de chaque type, excepté Arti qui contient moins de pauses que RATP-DECODA. Contrairement aux données réelles, ces corpus ont une répartition des types de disfluences équilibrée. La répartition des séquences est donnée par la figure 4.4. Tout comme pour RATP-DECODA, les séquences courtes sont majoritaires. Par construction, le taux de disfluences par mots n'évolue pas en fonction de la longueur des séquences.

Lors de nos expériences, nous utilisons le corpus Arti pour mesurer l'impact des différents paramètres de nos modèles et pour évaluer les résultats produits. Le corpus Arti+ sert à montrer l'impact du taux de disfluences et le corpus RATP-DECODA permet de mesurer les performances de nos modèles sur des données complexes dotées d'un vocabulaire beaucoup plus grand.

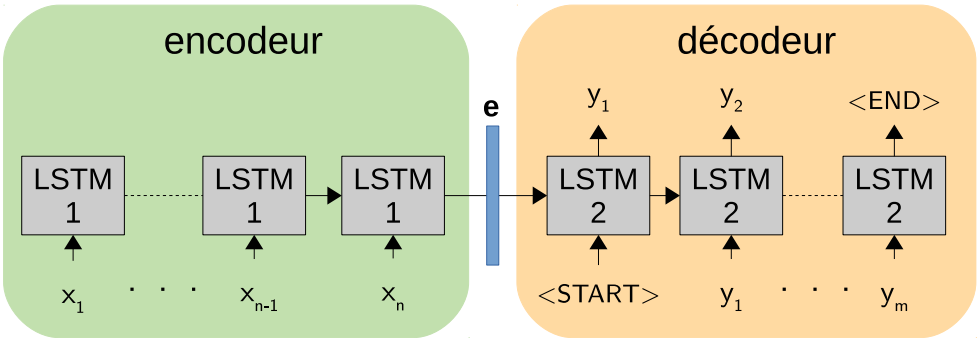

FIGURE 5.1 – Structure générale utilisée (encodeur-décodeur). $\mathbf{e}$: représentation de la séquence $(\mathrm{x}_i)_{i\in[\![1,n]\!]}$ de dimension latente l.

5 Modèles proposés

Dans cette section, nous allons présenter différents modèles séquence-à-séquence que nous avons expérimentés. Tout d'abord, nous détaillerons différentes modèles de type encodeur-décodeur. Ensuite, nous exposerons un modèle aléatoire qui nous servira de comparaison avec nos modèles neuronaux.

5.1 Modèles neuronaux de type encodeur-décodeur

Les modèles neuronaux que nous présentons sont grandement inspirés des modèles que l'on retrouve en traduction automatique et plus particulièrement de [Sutskever et al., 2014]. En effet, nous utiliserons la structure d'encodeur-décodeur comme base de nos modèles. Ce type de modèle est souvent utilisé dans le traitement des séquences, comme dans [Le and Mikolov, 2014] où cette structure est utilisée pour créer des représentations vectorielles de documents (*doc2vec*).

Comme le montre la figure 5.1, la partie encodeur lit la séquence d'entrées $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, élément par élément, afin de créer une représentation $\mathbf{e}$ de celle-ci. Cette représentation est de dimension latente l. On a donc l'équation :

$$\mathbf{e} = encodeur(\mathrm{x}_1, \ldots, \mathrm{x}_n) \tag{5.1}$$

Ensuite, la partie décodeur se sert de la représentation $\mathbf{e}$ de la séquence pour initialiser ses états internes ds_0. Il reçoit aussi un symbole de début de séquence <START> en entrée et tente alors de prédire le premier élément y_1 de la séquence de sortie. Puis, à partir de l'élément prédit, il va prédire l'élément suivant, et ainsi de suite, jusqu'à générer un symbole de fin de séquence <END>. À chaque étape, il met à jour ses états internes ds_j. Le décodeur peut être représenter par le système suivant :

$$\begin{cases} \mathrm{y}_0, ds_0 = <\text{START}>, \mathbf{e} \\ \mathrm{y}_j, ds_j = decodeur(\mathrm{y}_{j-1}, ds_{j-1}), \forall j \in [\![1, m]\!] \end{cases} \tag{5.2}$$

Les modèles que nous présentons ensuite sont des variantes de ce modèle : ils ont cette structure en commun et sont différenciés par la manière de fournir les entrées x_i et y_j à l'encodeur-décodeur.

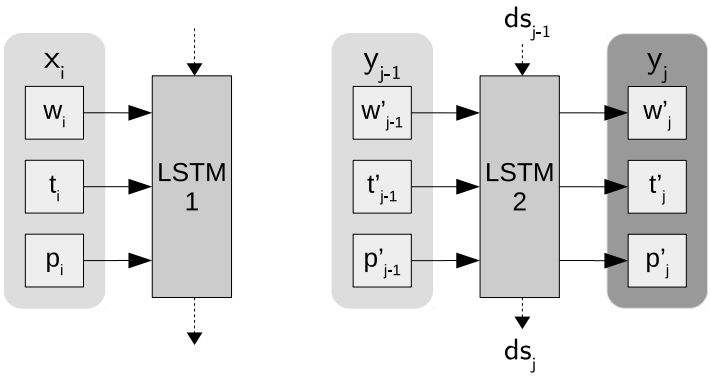

FIGURE 5.2 – Entrées et sorties des LSTM du modèle de base.

5.1.1 Modèle de base

Ce modèle est relativement basique. En effet, chaque élément x_i de l'encodeur et y_j du décodeur comportent trois composantes correspondant aux trois caractéristiques mot w, type de disfluence t et étiquette morphosyntaxique p. Ces composantes sont toutes encodées en *one-hot* et sont directement connectées aux couches LSTM du modèle (voir figure 5.2). Les équations 5.1 et 5.2 peuvent être complétées avec le système d'équations suivant :

$$\begin{cases} encodeur(\mathrm{x}_1, \ldots, \mathrm{x}_n) = LSTM1((w_1, t_1, p_1), \ldots, (w_n, t_n, p_n)) \\ \quad decodeur(\mathrm{y}_j, ds_j) = LSTM2((w'_j, t'_j, p'_j), ds_{j-1}), \forall j \in [\![1, m]\!] \end{cases} \tag{5.3}$$

Comme précisé dans la section 3.1, nos modèles sont multi-tâches. En effet, bien que nous ne nous intéressons pas à la prédiction du POS, pour chaque élément de la séquence produite, le modèle prédit celui-ci en plus du mot et du type de disfluence. Lors de la phase d'apprentissage, ce modèle va donc adapter ces poids afin de prédire au mieux ces trois informations.

Néanmoins, nous ne souhaitons pas accorder la même importance à chaque composante. En effet, nous voulons qu'en priorité le bon mot soit prédit, puis que le bon type de disfluence y soit associé, la prédiction du POS étant anecdotique. C'est pourquoi, nous proposons une variante de ce modèle pour laquelle nous pondérons les sorties. Ainsi, nous affectons arbitrairement un poids de 8 à la prédiction du mot w', de 4 à celle du type de disfluence t' et de 1 à celle du POS p' : il est donc 8 fois plus important de prédire le bon mot que de prédire le bon POS.

5.1.2 Modèle avec plongements

Dans le modèle de base que nous venons de présenter, les LSTM de l'encodeur et du décodeur reçoivent trois entrées en *one-hot*. Les entrées correspondant au type de disfluence et au POS sont de dimensions assez petites (respectivement 7 et 24). Cependant, alors que pour les corpus Arti+ et Arti l'entrée w est de taille 22, c'est-à-dire la taille du vocabulaire fluide, elle est d'une taille proche de 7500 pour le corpus RATP-DECODA. Le fait d'avoir une entrée de cette taille directement connectée aux LSTM augmente considérablement la durée d'apprentissage car, comme nous l'avons vu, les LSTM possèdent deux informations récurrentes (les entrées bouclent deux fois). Le fait d'avoir des entrées de dimension plus élevée augmente donc considérablement le nombre de paramètres.

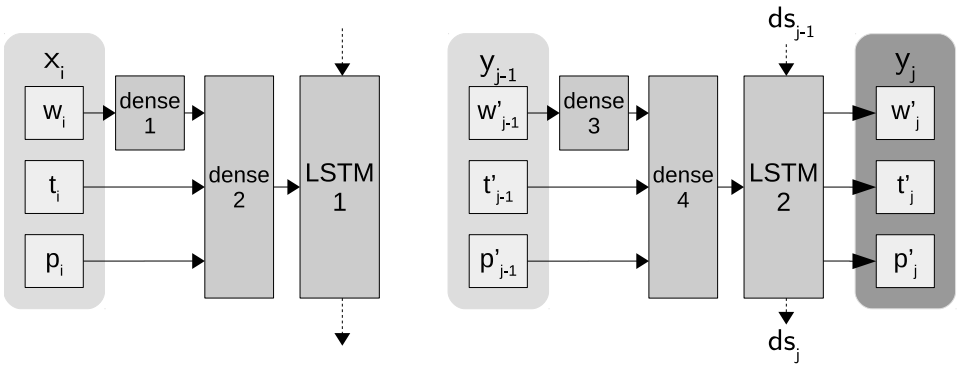

FIGURE 5.3 – Entrées et sorties des LSTM du modèle avec plongements.

Afin de limiter la durée d'apprentissage, nous proposons un second modèle utilisant des plongements. Dans ce modèle, les entrées ne sont plus directement connectées aux couches LSTM. Comme le montre la figure 5.3, nous connectons l'entrée w à une première couche dense afin de réduire la dimension de cette entrée. En sortie de cette couche, nous obtenons une représentation du mot w de taille 200. De plus, nous souhaitons fournir au LSTM1 une représentation globale de toutes les entrées : la sortie de la première couche dense et les entrées t et p sont connectées à une autre couche dense qui fournira un vecteur de taille 200 au LSTM1. Le même traitement est réalisé pour les entrées du décodeur. De cette façon, les informations récurrentes des LSTM seront beaucoup plus petites et l'apprentissage en sera plus rapide.

Cette fois, on précise les équations 5.1 et 5.2 avec le système d'équations suivant :

$$\begin{cases} d1_i = dense1(w_i) \\ d2_i = dense2(d1_i, t_i, p_i) \\ encodeur(\mathrm{x}_1, \ldots, \mathrm{x}_n) = LSTM1(d2_1, \ldots, d2_n) \\ d3_j = dense3(w'_j) \\ d4_j = dense4(d1_j, t'_j, p'_j) \\ decodeur(\mathrm{y}_j, ds_j) = LSTM2(d4_j, ds_{j-1}), \forall j \in [\![1, m]\!] \end{cases} \tag{5.4}$$

Comme pour le modèle de base, nous proposons une variante pondérée de ce modèle.

Ces modèles possèdent un grand nombre de paramètres. Tout d'abord, il y a les poids du réseau et nous avons vu que les LSTM en possèdent beaucoup. Ensuite, il y a plusieurs hyper-paramètres. Il y a la taille de *batch*, c'est le nombre de données parcourues avant chaque modification des poids. Il y a également le nombre d'époques : une époque correspond à un parcours de toutes les données par le réseau de neurones et il y a la dimension latente de la représentation en sortie de l'encodeur.

5.2 Modèle aléatoire

Nous proposons un modèle qui génère des disfluences aléatoirement. Pour chaque type de disfluences, ce modèle choisit aléatoirement la position de ces dernières (PI) en respectant les taux de disfluences par mot du corpus de référence. Ensuite, il y insère un contenu aléatoire lui-aussi.

Toutefois, ce contenu est restreint au type de disfluences (l'insertion d'une pause sera tirée aléatoirement parmi toutes les pauses possibles par exemple). De cette façon, nous vérifierons si nos modèles insèrent les disfluences à de meilleurs positions que l'aléatoire et si le contenu est également meilleur.

Pour résumer, nous étudions cinq modèles : un modèle dit « de base » que nous appelons **mod-B** et sa variante pondérée **mod-BP**, un modèle avec plongements que nous appelons **mod-P** et sa variante pondérée **mod-PP**, et un modèle aléatoire que nous appelons **mod-Alea**.

6 Expérimentations et résultats

Dans cette section, nous présentons, dans un premier temps, nos démarches expérimentales. Celles-ci définissent le contexte de nos expérimentations. Nous présentons ensuite les expérimentations réalisées et discutons de leurs résultats.

6.1 Démarches expérimentales

Afin de réaliser les différents entraînements des modèles neuronaux, nous avons divisé chaque corpus en trois ensembles :

— un ensemble d'entraînement (70% des données) : c'est avec ces données que nos modèles sont appris ;
— un ensemble de développement (15% des données) : ces données sont utiles pour régler les paramètres ;
— un ensemble de test (15% des données) : nous utilisons ces données pour évaluer les modèles.

Les apprentissages ont été réalisés sur un serveur GPU avec 2 cartes GTX 1080 et 32 Go de RAM. Les modèles neuronaux ont été réalisés avec l'API fonctionnelle de Keras. La fonction de perte utilisée est l'entropie croisée catégorielle, telle qu'utilisée dans des travaux de traduction. Cette fonction va servir au modèle à adapter ses poids, le but étant de minimiser la perte mesurée par cette fonction entre les prédictions du modèle et les références attendues. La sortie d'un modèle neuronal n'étant pas directement l'encodage *one-hot* correspondant aux sorties souhaitées mais une distribution, l'entropie croisée catégorielle évalue la différence entre la distribution prédite et celle de référence. La fonction d'optimisation que nous utilisons est RMSprop.

Nous avons vu que les modèles neuronaux possèdent plusieurs hyper-paramètres. Nous avons choisi une taille de *batch* égale à 32, un nombre d'époques de 100, sauf indication contraire (chaque donnée sera parcourue 100 fois par le modèle) et une dimension latente égale à 256, sauf indication contraire.

Les probabilités nécessaires à l'évaluation des mesures de crédibilités sont calculées grâce à des modèles de langage. Ces derniers ont été appris sur les données d'entraînement avec l'outil SRILM. Différentes versions de chaque corpus ont été réalisées pour obtenir des corpus contenant un seul type de disfluences. Des versions supplémentaires ont été produites en remplaçant chaque disfluence par des symboles les représentant. De cette façon, nous avons pu apprendre les modèles de langage pour les mesures de crédibilité positionnelle.

Corpus	Modèle	Paramètres	Préserv.(%)	Distor.(%)
Arti	mod-B	/	**98.78**	**1.40**
	mod-BP	/	98.29	2.05
RATP-DECODA	mod-P	/	42.12	86.71
	mod-PP	/	45.77	83.24
	mod-B	/	46.04	65.96
	mod-BP	/	48.52	68.05
		200 époques	50.59	73.46
		dim. latente = 512	50.8	64.29
		longueur max. = 30	**56.16**	**64.04**

TABLE 6.1 – Résultats des différentes expériences pour l'auto-encodage sur l'ensemble de test.

6.2 Auto-encodage

Avant de nous intéresser à la production de disfluences, il nous a semblé pertinent d'évaluer les différents modèles sur une tâche d'auto-encodage. Un auto-encodage signifie que le modèle apprend à prédire son entrée. Dans notre cas, chaque modèle reçoit en entrée des séquences de mots fluides et doit prédire ces mêmes séquences fluides.

Nous avons entraîné les différents modèles sur chaque corpus, puis évalué uniquement la préservation et la distorsion des séquences produites. Les autres métriques sont effectivement inutiles pour cette tâche d'auto-encodage puisqu'aucune disfluence n'est produite. Nous n'avons pas entraîné les modèles mod-P et mod-PP sur les données artificielles car ces modèles ont pour but d'accélérer la durée d'apprentissage lorsque les dimensions des entrées sont grandes, ce qui n'est pas le cas des données artificielles. En effet, les modèles mod-B et mod-BP prennent 100 secondes pour réaliser une époque sur le corpus Arti, alors qu'ils prennent environ 30 minutes sur le corpus RATP-DECODA. Les modèles mod-P et mod-PP prennent quant à eux une dizaine de minutes par époque.

Les résultats pour l'auto-encodage sont donnés dans la table 6.1. Nous notons tout d'abord que, dans le cas des données artificielles, les modèles mod-B et mod-BP parviennent très bien à restituer la séquence initiale : la préservation est aux alentours de 98.5%. Ces deux variantes ont des résultats similaires. Il semble que la version pondérée soit légèrement moins performante (préservation environ 0.5% plus faible et distorsion 0.6% plus élevée) mais les différences ne sont pas significatives. Nous remarquons également que les modèles ajoutent très peu de mots. En effet, le calcul $\textit{distorsion} - (100 - \textit{préservation})$ fournit le nombre d'insertion par mot fluide et nous observons que cette valeur varie de 0.18 à 0.34%. Les résultats sur l'auto-encodage des données artificielles sont donc prometteurs pour la suite.

Les résultats chutent dès que l'on s'intéresse aux données réelles du corpus RATP-DECODA. Le modèle mod-B ne préserve que 46.04% de la séquence initiale et la distorsion est élevée. Le modèle mod-P a des résultats encore moins bons : 4 points de préservation en moins et une différence plus importante au niveau de la distorsion (20% en plus). Nous pensons que cette différence est due aux plongements en sortie des couches denses du modèle mod-P qui sont de mauvaises qualités. Ces couches ne semblent pas parvenir à créer des représentations pertinentes des entrées, nous pourrions envisager d'utiliser des plongements déjà calculés pour d'autres applications. Contrairement aux données factices, les variantes pondérées de ces modèles montrent une amélioration notable (2 à 3%

ID	Corpus	Modèle	Remarques
R-D	RATP-DECODA	mod-BP	/
A+	Arti+		/
A	Arti		/
A-AE			spéc. de l'auto-encodeur
A-apAlea			appris sur l'aléatoire
A-modAlea		mod-Alea	/

TABLE 6.2 – Description des différentes expériences de production de disfluences.

de préservation en plus). Dans l'ensemble, les résultats sur le corpus réel sont décevants. Cela peut être expliqué par la grande taille du vocabulaire qui rend l'apprentissage beaucoup plus compliqué. En effet, il y a à peine 3 fois plus de données d'entraînement que de mots différents dans le vocabulaire. Un vocabulaire plus grand signifie qu'un certain nombre de mots sont peu fréquents : les modèles ne parcourent pas assez ces mots et ne les rencontrent pas dans des situations assez variées. L'ensemble de test a donc de grandes chances de contenir ces mots rares dans des configurations absentes de l'ensemble d'entraînement et les modèles neuronaux ne savent pas traiter correctement les données qui ressemblent peu aux données d'entraînement.

Nous avons cherché les hyper-paramètres qui pourraient améliorer ces résultats en utilisant le modèle le plus encourageant : mod-BP. En doublant le nombre d'époques, la préservation est légèrement meilleure au détriment de la distorsion. Les résultats ne sont donc pas mauvais à cause d'un manque d'apprentissage. Nous nous sommes alors intéressés à la dimension latente. En effet, comme la taille du vocabulaire, et donc des entrées de l'encodeur, est plus grande, il est possible qu'une dimension de 256 ne suffisent pas pour encoder toute la séquence initiale. Avec une dimension latente de 512, le modèle est plus performant (meilleur préservation et distorsion). Enfin, nous essayons de voir si la longueur des séquences ne serait pas trop contraignante pour le modèle car plus les séquences sont longues, plus il y a d'informations à encoder et décoder. Nous effectuons l'apprentissage uniquement sur les séquences de longueur inférieure à 30 mots, au lieu de 40. Bien que les séquences de plus de 30 mots ne représentent que 2.5% de toutes les séquences, l'impact est élevé : la préservation passe à 56% (+8%) et la distorsion à 64% (-4%). Cependant, les données artificielles nous ont prouvées que le modèle mod-BP est capable de traiter les longues séquences, les mauvais résultats dépendent donc surtout de la taille du vocabulaire. Le but étant de rendre disfluentes des phrases réelles, peu importe leur longueur, nous utiliserons les séquences de longueur inférieure à 40 mots pour nos futures expériences. Nous utiliserons également une dimension latente de 256 car plus celle-ci est grande, plus l'apprentissage est long. Toutefois, nous garderons en mémoire que l'augmenter peut améliorer légèrement les résultats.

6.3 Production de disfluences

Nous nous intéressons maintenant à la production de disfluences telle que décrite précédemment dans la section 3.1. Nous souhaitons étudier le comportement des modèles pour chaque type de disfluences. Nous étudierons donc la production de chaque type de disfluences individuellement tout d'abord, puis la production de toutes les disfluences en même temps. L'ensemble des expériences réalisées est détaillé dans la table 6.2. Ces expériences sont réalisées pour chaque travail de production. Peu importe le corpus, le modèle neuronal que nous utilisons est le modèle mod-BP puisque

c'est le modèle qui donnait les meilleurs résultats pour l'auto-encodage dans le cas des données réelles. Nous nous concentrerons sur le corpus Arti car le taux de disfluences est similaire à celui des données réelles. L'expérience A consiste à entraîner le modèle mod-BP sur le corpus Arti. Nous essaierons également des variantes du modèle mod-BP. Tout d'abord, l'expérience A-AE pour laquelle nous réalisons une spécialisation de l'auto-encodeur : le modèle apprend tout d'abord la tâche d'auto-encodage $\mathcal{F} \to \mathcal{F}$, puis ce même modèle (avec les poids appris) se ré-entraîne (se spécialise) sur la tâche $\mathcal{F} \to \mathcal{D}_X$. Ainsi, l'apprentissage de la production de disfluences se fera avec un modèle qui sait déjà garder en mémoire la séquence initiale, nous espérons que le modèle aura une meilleure préservation et distorsion. Ensuite, l'expérience A-apAlea consiste à entraîner le modèle à produire des disfluences aléatoires et de l'évaluer sur les disfluences de référence (ces dernières ne sont pas aléatoires) afin de mesurer la capacité du modèle à reproduire les disfluences du corpus. Cela sert de comparaison avec un modèle neuronal aléatoire. Ces disfluences aléatoires sont celles produites par l'expérience A-modAlea qui utilise tout simplement le modèle mod-Alea de la section 5.2. Les résultats de l'auto-encodage avec les données réelles étant relativement mauvais, nous présenterons malgré tout les résultats de la production de disfluences pour celles-ci pour en donner un aperçu. De même, nous étudierons la production de disfluences sur les données Arti+ pour comprendre comment le taux de disfluences influe sur leur production.

Les prochaines sections présentent les résultats de ces expériences sur les données de test. Nous commencerons par la production de pauses, puis celle de répétitions et enfin la production de révisions. Pour finir, nous présenterons les résultats sur la production de toutes les disfluences en même temps.

6.3.1 Pauses seules

Nous commençons par étudier la seule production de pauses. Cette tâche possède des caractéristiques spécifiques. Tout d'abord, le vocabulaire employé lors des pauses n'est pas inclus dans le vocabulaire fluide. En effet, de nombreux mots (tous pour les données artificielles) du vocabulaire des pauses ne sont utilisés que pour les pauses. Par exemple, les mots « euh » ou « hmmm » sont forcément des pauses. Ensuite, la longueur des sections de pauses est en moyenne plus courte que celles des autres disfluences. Dans la plupart des cas, une pause n'est composée que d'un mot (il y a quelques exceptions comme « je veux dire »). Enfin, le contenu des pauses possède moins de liens que pour les répétitions ou les révisions. En effet, ces dernières doivent reproduire ou corriger une partie des séquences alors que d'après la littérature, les pauses servent à ralentir le discours. Avec ces caractéristiques, nous avons pour intuition que c'est la tâche de production de disfluences la plus simple.

Les résultats des expériences sont visibles dans la table 6.3. Comme en témoignent les mesures de préservation de de distorsion, les modèles parviennent correctement à conserver la séquence initiale pour les données artificielles. Si l'on ne prend pas en compte l'expérience A-modAlea (le modèle aléatoire a par construction une préservation parfaite), les résultats sont évidemment moins bons que pour l'auto-encodage mais restent satisfaisants. La distorsion est relativement élevée par rapport à 100−préservation, c'est-à-dire que le modèle a tendance à ajouter des mots fluides. Nous avons regardé en détail les séquences produites par les modèles neuronaux et nous avons remarqué que, parmi les mots ajoutés, de nombreux sont des pauses qui ne sont pas étiquetées en tant que telles. Par exemple, « euh » est parfois considéré comme étant fluide. Cela est d'autant plus étrange que les pauses n'existent pas dans le corpus Arti en tant que mot fluide. Ces mots mal étiquetés augmentent la distorsion du modèle. Étonnamment, les meilleurs résultats pour ces mesures sont

ID	Préserv.(%)	Distor.(%)	F1(%)	IRR(%)	Conf.(%)
R-D	51.67	101.07	20.62	25.54	83.87
A+	83.45	45.71	21.04	45.53	90.03
A	**89.53**	**23.48**	1.51	234.85	**99.18**
A-AE	86.53	23.75	1.06	776.90	98.47
A-apAlea	85.48	28.05	**4.87**	**107.47**	83.33
A-modAlea	100	0.0	3.95	110.43	100

TABLE 6.3 – Résultats pour la production de pauses sur l'ensemble de test.

ceux de l'expérience A (avec le modèle mod-BP d'origine). La spécialisation de l'auto-encodeur (A-AE) n'a pas eu l'effet escompté car l'information initiale est moins bien conservée. Les mesures d'IRR indiquent que, pour le corpus Arti, le modèle mod-BP tend à produire plus de disfluences que dans la référence. Toutefois, la forme des disfluences produites est globalement correcte. L'expérience A-apAlea produit un taux de disfluences proche de la référence, mais il semblerait que réaliser l'apprentissage sur des disfluences aléatoires perturbe le modèle. En effet, les pauses produites sont moins conformes que pour les autres expériences. La mesure F_1 met en évidence que le modèle mod-BP prédit mal la position des pauses du corpus Arti.

Lorsque les données contiennent davantage de disfluences (corpus Arti+), les résultats du modèle mod-BP diminuent sensiblement. L'information initiale est moins bien conservée et le modèle produit moins de disfluences. De plus, ces dernières sont moins conformes. La mesure F_1 est comparable aux résultats de l'état de l'art, mais cela peut être expliqué par le fait que, le corpus étant davantage disfluent, il y a plus de positions correctes pour insérer les pauses.

Pour le corpus RATP-DECODA, la préservation est équivalente avec celle de l'auto-encodage. Tout comme pour les données artificielles, la distorsion augmente (+30% environ). Le modèle mod-BP produit très peu de disfluences et celles-ci sont globalement conformes, si on prend en compte qu'elles sont beaucoup plus variées et complexes que pour le corpus Arti. La différence avec les données artificielles provient sûrement du fait que les pauses peuvent être composées de plusieurs mots et que le vocabulaire des pauses n'est pas disjoint avec le vocabulaire fluide. La mesure F_1 est également comparable aux résultats de l'état de l'art. Une explication possible à ce phénomène est la distorsion élevée. À cause de celle-ci, l'alignement de la séquence produite avec la référence n'est pas évident et il est difficile d'évaluer correctement la mesure F_1.

L'ensemble des mesures de crédibilité présenté dans la table 6.4 est biaisé. En effet, si la préservation et la distorsion était parfaite (comme pour l'expérience A-modAlea), ces mesures devraient évaluer la crédibilité des disfluences. Or, dans notre cas, la séquence prédite est peu crédible même sans prendre en compte les disfluences. Cela est mis en évidence par les versions fluides des mesures de crédibilité qui sont beaucoup plus élevées que leur équivalent non fluides. Malgré tout, nous remarquons que l'expérience A semble la plus réussie et que, bien que la séquence initiale soit entièrement conservée, le modèle aléatoire (A-modAlea) n'a pas une crédibilité similaire à celle de la référence.

ID	CP	CPF	$\widehat{\text{CP}}$	$\widehat{\text{CPF}}$	CS	CSF	$\widehat{\text{CS}}$	$\widehat{\text{CSF}}$
R-D	1.42	$\sim 10^{116}$	1.41	$\sim 10^{116}$	2.33	$\sim 10^{115}$	2.39	$\sim 10^{115}$
A+	235.05	528.5	243.47	513.11	90.44	103.53	119.95	132.78
A	**2.20**	147.80	**2.67**	157.83	49.75	681.5	34.86	652.14
A-AE	6.60	415.60	6.57	423.71	**44.0**	**54.25**	**29.43**	**40.86**
A-apAlea	86.60	**108.40**	103.0	**129.0**	728.75	775.0	575.57	607.86
A-modAlea	0.60	0.60	0.67	0.67	0.50	0.50	0.43	0.43

TABLE 6.4 – Mesures de crédibilité pour la production de pauses sur l'ensemble de test. Les valeurs indiquées correspondent au résultat du modèle sur le résultat de la référence.

ID	Préserv.(%)	Distor.(%)	F1(%)	IRR(%)	Conf.(%)
R-D	52.04	127.95	3.48	65.91	4.73
A+	87.11	190.08	26.08	34.08	7.63
A	87.85	164.16	**10.23**	**52.66**	5.61
A-AE	**88.92**	191.97	6.60	22.55	**16.14**
A-apAlea	57.29	**94.96**	4.28	35.32	9.65
A-modAlea	100	0.0	3.82	95.90	0.70

TABLE 6.5 – Résultats pour la production de répétitions sur l'ensemble de test.

6.3.2 Répétitions seules

Nous étudions maintenant la production de répétitions seules. Cette tâche est plus compliquée que la production de pauses. En effet, le vocabulaire des répétitions et le vocabulaire fluide sont identiques. De plus, la longueur des répétitions est variable. Enfin, le lien avec le contexte est extrêmement fort : il s'agit de reproduire le contenu d'une partie de la séquence avant que cette partie ne soit produite.

Les résultats de cette tâche sont présentés dans la table 6.5. Dans un premier temps, nous remarquons que la préservation est similaire à celle pour la production de pauses, à l'exception du modèle neuronal aléatoire. Pour ce dernier, le contenu des répétitions des données d'entraînement est aléatoire. Par conséquent, il est plus difficile de dégager une logique dans l'enchaînement des mots et le modèle a du mal à distinguer la séquence fluide. La distorsion est nettement plus élevée que pour la production de pauses. Cela signifie que de nombreux mots, considérés comme fluides, sont ajoutés. Ce phénomène a un impact sur les autres mesures (hormis la conformité). Par exemple, l'IRR mesure le rapport entre le taux de disfluences par mot fluide des séquences prédites et celui de la référence. Cependant, le modèle prédisant trop de mots fluides, le taux sera biaisé. Cela peut expliquer les valeurs d'IRR très faibles.

L'observation de la mesure *conformité* confirme la difficulté de la tâche. Le meilleur résultat est obtenu par le modèle mod-BP spécialisé à partir de l'auto-encodeur (expérience A-AE), et il n'est que de 16.14%. Nous pourrions remettre en doute la mesure car elle est très stricte : il ne faut aucune

ID	CP	CPF	$\widehat{\text{CP}}$	$\widehat{\text{CPF}}$	CS	CSF	$\widehat{\text{CS}}$	$\widehat{\text{CSF}}$
R-D	$\sim 10^{116}$	$\sim 10^{116}$	$\sim 10^{116}$	$\sim 10^{116}$	$\sim 10^{116}$	$\sim 10^{116}$	$\sim 10^{116}$	$\sim 10^{116}$
A+	$\sim 10^{51}$	$\sim 10^{51}$	$\sim 10^{51}$	$\sim 10^{51}$	$\sim 10^{59}$	$\sim 10^{59}$	$\sim 10^{61}$	$\sim 10^{61}$
A	$\sim 10^{33}$	$\sim 10^{33}$	$\sim 10^{33}$	$\sim 10^{33}$	$\sim 10^{31}$	$\sim 10^{31}$	$\sim 10^{33}$	$\sim 10^{33}$
A-AE	$\sim 10^{33}$	$\sim 10^{33}$	$\sim 10^{33}$	$\sim 10^{33}$	$\sim 10^{22}$	$\sim 10^{22}$	$\sim 10^{22}$	$\sim 10^{22}$
A-apAlea	6374.40	7977.80	6565.14	8216.43	$\sim 10^{4}$	$\sim 10^{5}$	$\sim 10^{5}$	$\sim 10^{5}$
A-modAlea	0.20	0.20	0.29	0.29	0.60	0.01	0.50	0.03

TABLE 6.6 – Mesures de crédibilité pour la production de répétitions sur l'ensemble de test. Les valeurs indiquées correspondent au résultat du modèle sur le résultat de la référence.

ID	Préserv.(%)	Distor.(%)	F1(%)	IRR(%)	Conf.(%)
R-D	55.92	102.42	2.12	5.50	98.93
A+	85.21	67.90	4.77	5.82	96.97
A	**80.79**	65.67	2.16	22.47	**97.97**
A-AE	76.24	70.08	**5.76**	**126.40**	96.35
A-apAlea	73.34	**51.22**	3.18	19.59	90.99
A-modAlea	100	0	5.26	88.29	95.17

TABLE 6.7 – Résultats pour la production de révisions sur l'ensemble de test.

erreur entre le *reparandum* et la réparation pour qu'une répétition soit bien formée. Toutefois, en étudiant les séquences produites, nous remarquons que les mots des répétitions ne correspondent que rarement à ceux de la réparation.

Les mesures de crédibilité (table 6.6) ne sont pas du tout pertinentes car nous obtenons des résultats absurdes pour celles-ci. Cela peut être expliqué par la distorsion élevée qui ajoute un biais conséquent à ces mesures. De plus, le peu de répétitions correctement formées impacte ces mesures. En effet, celles-ci évaluent la crédibilité des séquences produites. Il est donc logique que des répétitions mal formées soient peu crédibles.

Les résultats sur les données réelles confirment nos remarques.

6.3.3 Révisions seules

Nous nous intéressons cette fois à la seule production de révisions. Cette tâche est similaire à la production de répétitions, sans être aussi complexe. Comme pour les répétitions, le vocabulaire des révisions est le même que le vocabulaire fluide. Cependant, les révisions artificielles comportent un seul mot (ce n'est pas le cas pour les données réelles) et le lien avec le contexte est moins fort.

Les résultats de cette tâche sont présentés dans la table 6.7. Cette fois, nous remarquons une différence significative de préservation par rapport aux autres tâches. En effet, elle est en générale plus faible (-8 à 10%). Toutefois, la distorsion diminue également. Les expériences réalisées produisent donc moins de mots. Cette impression est vérifiée par l'IRR qui est plutôt faible.

ID	CP	CPF	$\widehat{\text{CP}}$	$\widehat{\text{CPF}}$	CS	CSF	$\widehat{\text{CS}}$	$\widehat{\text{CSF}}$
R-D	$\sim 10^5$	$\sim 10^5$	$\sim 10^5$	$\sim 10^5$	20,50	20.50	17.60	17.60
A+	42.67	60.20	41.43	58.43	$\sim 10^4$	$\sim 10^4$	$\sim 10^4$	$\sim 10^4$
A	789.75	827.25	719.17	753.17	6757.43	6630.0	4859.27	4763.82
A-AE	1456.75	$\sim 10^{116}$	1326.50	$\sim 10^{116}$	5897.0	$\sim 10^{116}$	5093.18	$\sim 10^{116}$
A-apAlea	2874.75	2899.0	2617.83	2639.83	559.43	550.43	553.18	546.18
A-modAlea	0.75	0.75	0.67	0.67	82.43	0.14	71.45	0.18

TABLE 6.8 – Mesures de crédibilité pour la production de révisions sur l'ensemble de test. Les valeurs indiquées correspondent au résultat du modèle sur le résultat de la référence.

ID	Préserv.(%)	Distor.(%)	F1(%)	IRR(%)	Conf.(%)
R-D	53.71	201.06	23.52	38.68	72.94
A+	77.70	501.05	20.19	34.59	66.18
A	**81.42**	372.49	**7.25**	**110.05**	88.19
A-AE	49.03	**109.17**	2.51	2522.25	**98.89**
A-apAlea	56.94	148.89	5.31	9.08	45.40
A-modAlea	100	0.0	5.27	97.37	65.31

TABLE 6.9 – Résultats pour la production de toutes les disfluences sur l'ensemble de test.

Les taux de révisions bien formées sont élevés, mais, comme nous l'avons déjà évoqué, la mesure de conformité est très permissive pour les révisions. En étudiant les séquences produites, on remarque que très peu de révisions produites peuvent être considérées comme naturelles.

Les mesures de crédibilité, présentées dans la table 6.8, montrent encore une fois que la crédibilité des séquences produites est très différente de celles servant de référence. Ces mesures sont tout de même moins absurdes que celles pour la production de répétitions, ce qui confirme l'impact de la distorsion sur ces mesures.

6.3.4 Composition des disfluences

Nous étudions maintenant la production de toutes les disfluences d'un coup. En plus de cumuler les difficultés relatives à chacune des tâches de production différentes, les modèles doivent apprendre à différencier les disfluences.

Les résultats sont visibles dans la table 6.9. Comme on pouvait s'y attendre, dans la plupart des cas, la préservation est moins bonne que pour chaque tâche individuelle. Toutefois, l'expérience A a une préservation similaire à celle pour la production de révisions seules. Ce modèle fournit même des résultats satisfaisants pour les autres mesures. L'IRR est proche de 100%, mais il faut prendre des précautions avec ce résultat. En effet, pour cette expérience, la distorsion est plus élevée que pour les autres expériences, ce qui signifie que plus de mots fluides sont produits. Par conséquent, à nombre égal de disfluences produites, l'IRR sera plus faible. On peut donc supposer que le modèle produit

ID	CP	CPF	$\widehat{\text{CP}}$	$\widehat{\text{CPF}}$	CS	CSF	$\widehat{\text{CS}}$	$\widehat{\text{CSF}}$
R-D	$\sim 10^{115}$	$\sim 10^{116}$	$\sim 10^{115}$	$\sim 10^{116}$	$\sim 10^{113}$	$\sim 10^{115}$	$\sim 10^{113}$	$\sim 10^{115}$
A+	8.35	2648.08	7.89	2545.46	$\sim 10^{86}$	$\sim 10^{89}$	$\sim 10^{88}$	$\sim 10^{90}$
A	355.0	2175.25	351.75	2157.67	$\sim 10^{14}$	$\sim 10^{14}$	$\sim 10^{15}$	$\sim 10^{16}$
A-AE	19.13	$\sim 10^{116}$	18.92	$\sim 10^{116}$	$\sim 10^{12}$	$\sim 10^{116}$	$\sim 10^{13}$	$\sim 10^{116}$
A-apAlea	1113.38	4138.75	1076.42	3982.17	3476.60	6647.0	$\sim 10^{4}$	$\sim 10^{4}$
A-modAlea	2.13	0.50	2.08	0.50	6.30	0.25	5.06	0.22

TABLE 6.10 – Mesures de crédibilité pour la production de toutes les disfluences sur l'ensemble de test. Les valeurs indiquées correspondent au résultat du modèle sur le résultat de la référence.

également trop de disfluences pour l'expérience A. De même, les disfluences produites ont l'air bien formées. Toutefois, en regardant le détail pour chaque type de disfluence, on remarque que les pauses sont bien formées ainsi que les révisions, mais pas les répétitions. Étant donnée la permissivité de la mesure de conformité pour les révisions, le seul résultat satisfaisant est la conformité des pauses produites.

L'expérience A-AE montre des résultats surprenants. Sa préservation est faible, sa distorsion plus faible que pour les autres modèles et, bien que l'IRR soit anormalement élevé (25 fois plus de disfluences par mot que dans la référence), le taux de disfluences bien formées est plus élevé que pour les autres expériences. Nous faisons la même remarque que pour l'expérience A : la conformité des pauses est élevée, celle des répétitions est faible et celle des révisions est peu significative. Toutefois, les disfluences sont globalement mieux formées. Cette expérience avait pour but d'améliorer la préservation et la distorsion en spécialisant un modèle appris sur l'auto-encodage, quitte à diminuer les performances sur la production de disfluences, mais cela a eu l'effet inverse. Nous pouvons supposer que, contrairement à ce que nous pensions, la tâche de production de disfluences n'est pas une variante de l'auto-encodage.

Pour le corpus Arti+, la distorsion est encore plus élevée que pour l'expérience A. L'IRR est très faible, ce qui semble cohérent avec notre hypothèse sur le biais provoqué par la distorsion. Pour le corpus RATP-DECODA, les résultats semblent être la moyenne des résultats pour la production de chaque type de disfluences séparemment Toutefois, la distorsion est plus élevée, ce qui confirme la difficulté à traiter les trois types de disfluences en même temps.

Les mesures de crédibilité (table 6.10) sont toujours trop biaisées par la distorsion élevée et la faible préservation pour être pertinentes.

Ces expériences nous révèlent que la difficulté majeure est la préservation de l'information initiale. En effet, lors de la production de disfluences, le modèle apprend à ajouter des mots aux séquences initiales mais cela perturbe l'information initiale d'une part, et d'autre part, les mots ajoutés sont mal catégorisés (des disfluences considérées comme étant fluides et des mots fluides comme étant disfluents). Cette difficulté de préservation de l'information initiale est accentuée lorsque les données sont plus complexes (corpus RATP-DECODA). Les taux de distorsion élevés de nos expériences ne nous permettent pas de juger de la pertinence de nos métriques sur les disfluences.

Il est à noter que les séquences obtenues sont actuellement trop chaotiques pour qu'une évaluation

subjective soit réalisée, bien que nous l'ayons envisagée.

7 Conclusion

Dans ce rapport, nous avons étudié comment insérer des disfluences dans des textes fluides afin de donner à ces textes l'apparence d'un style oral spontané. En particulier, nous avons visé à répondre à ce problème par l'emploi de réseaux de neurones séquences-à-séquence. Pour cela, notre état de l'art a mis en évidence l'importance des disfluences dans le caractère naturel de la parole spontanée, ainsi que des réseaux de neurones dans le traitement de diverses tâches séquence-à-séquence. À notre connaissance, il n'existe pas de travaux sur la production des disfluences utilisant des réseaux de neurones, nous voulions donc évaluer la possibilité et la complexité de cette tâche. Nous avons étudié la production de disfluences comme un problème de traduction automatique d'un texte fluide vers un texte disfluent. En s'inspirant des travaux en traduction automatique, nous avons proposer plusieurs variantes d'un modèle neuronal fondé sur la structure d'encodeur-décodeur. La production de disfluences étant relativement peu étudiée, il n'existe pas de métriques dédiées à cette tâche. Nous avons proposé certaines métriques pour évaluer les différentes caractéristiques des disfluences. Pour évaluer nos modèles, nous disposions du corpus RATP-DECODA, composé de parole spontanée naturelle issue d'un centre d'appel de la RATP. Nous avons également utilisé des données produites artificiellement. Nos travaux ont révélé plusieurs phénomènes. Tout d'abord, la production de disfluences est complexe, même pour des modèles neuronaux simples. La difficulté majeure pour ces modèles est de conserver la séquence de mots initiale dans la séquence produite. Malgré tout, nous avons remarqué que les pauses sont plus simples à produire que les autres disfluences. Nous avons mis en évidence que nos métriques sont sensibles à la conservation de la séquence initiale. Nous n'avons donc pas évalué la pertinence de celles-ci. Nous aurions voulu effectuer des évaluations subjectives, mais les séquences que nous avons produites étaient de trop mauvaises qualités pour les présenter à des testeurs. La production de disfluences par des modèles neuronaux ne semble pas encore possible, car la tâche d'auto-encodage est déjà un problème à résoudre.

De nombreuses idées, que nous n'avons pas pu essayer, restent à explorer. Afin d'améliorer la conservation des séquences initiales, il serait intéressant d'utiliser un encodeur-décodeur *« peeky »*, tel que celui proposé par [Cho et al., 2014]. Ce dernier a une structure similaire à celle que nous avons utilisée mais le décodeur reçoit la sortie de l'encodeur à chaque étape de production. Cela pourrait améliorer la conservation de la séquence initiale. Afin d'améliorer les résultats des modèles entraînés pour de l'auto-encodage, puis spécialisés pour la production de disfluences, il pourrait être envisagé de réaliser l'apprentissage de l'auto-encodage sur des données plus générales et plus nombreuses telles que le corpus WikipédiaFR2008. Nous pensons qu'avec des modèles plus complexes et d'éventuels post-traitements, la production de disfluences par des modèles neuronaux sera possible. Toutefois, pour une séquence source, ce modèle produira toujours la même sortie alors qu'en réalité, il y a plusieurs bonnes façons de produire des disfluences. C'est pourquoi, une fois les problèmes actuels résolus, l'utilisation d'une variante de GAN, telle que décrite dans la section 2.2.1.3 pourrait être judicieuse. Le vecteur de bruit en entrée devrait permettre de produire plusieurs séquences disfluentes différentes pour une même séquence en entrée.

Références

[Adell et al., 2007] Adell, J., Bonafonte, A., and Escudero, D. (2007). Filled pauses in speech synthesis : towards conversational speech. *Lecture Notes in Computer Science*, 4629.

[Adell et al., 2012] Adell, J., Bonafonte, A., and Escudero, D. (2012). Production of filled pauses in concatenative speech synthesis based on the underlying fluent sentence. *Speech Communication*, 54 :459–476.

[Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

[Betz et al., 2015] Betz, S., Wagner, P., and Schlangen, D. (2015). Micro-structure of disfluencies : Basics for conversational speech synthesis. *Proceedings of Interspeech*.

[Blankenship and Kay, 1964] Blankenship, J. and Kay, C. (1964). Hesitation phenomena in spontaneous english speech : A study in distribution. *Word*, 20 :360–372.

[Bruner and Deshpande, 2017] Bruner, J. and Deshpande, A. (2017). `https://www.oreilly.com/learning/generative-adversarial-networks-for-beginners`.

[Cho et al., 2014] Cho, K., Merriënboer, B. V., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

[Christenfeld et al., 1991] Christenfeld, N., Schachter, S., and Bilous, F. (1991). Filled pauses and gestures : it's not coincidence. *Journal of Psycholinguistic Research*, 20.

[Chung et al., 2014] Chung, J., Cho, K., Gülçehre, Ç., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

[Dall et al., 2014] Dall, R., Tomalin, M., Wester, M., Byrne, W., and King, S. (2014). Investigating automatic and human filled pause insertion for speech synthesis. *Proceedings of Interspeech*.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, pages 2672–2680.

[Greff et al., 2017] Greff, K., Srivastava, R., Koutnik, J., Steunebrink, B. R., and Schmidhuber, J. (2017). LSTM : A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28 :2222–2232.

[Hassan et al., 2014] Hassan, H., Schwartz, L., Hakkani-Tür, D., and Tur, G. (2014). Segmentation and disfluency removal for conversational speech translation. *Proceedings of Interspeech*.

[Hermann et al., 2015] Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. *Advances in Neural Information Processing Systems 28*.

[Le and Mikolov, 2014] Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. *Proceedings of the International Conference on Machine Learning 2014*.

[Maclay and Osgood, 1959] Maclay, H. and Osgood, C. E. (1959). Hesitation phenomena in spontaneous english speech. *Word*, 15 :19–44.

[Mikolov et al., 2013] Mikolov, T., Yih, W., and Zweig, G. (2013). Linguistic regularities in continuous space word representations. *Proceedings of the North American Chapter of the Association for Computionnal Linguistics : Human Language Technologies*, pages 746–751.

[Pascanu et al., 2013] Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *Proceedings of the 30-th International Conference on Machine Learning*, 28.

[Qader, 2017] Qader, R. (2017). *Pronunciation and disfluency modelling for spontaneous speech synthesis*. PhD thesis, University of Rennes 1.

[Qader et al., 2014] Qader, R., Lecorvé, G., Lolive, D., and Sébillot, P. (2014). Ajout automatique de disfluences pour la synthèse de la parole spontanée : formalisation et preuve de concept. *Proceedings of TALN*.

[Rao et al., 2015] Rao, K., Peng, F., Sak, H., and Beaufays, F. (2015). Grapheme-to-phoneme conversion using long-short-term memory recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*.

[Rose, 1998] Rose, R. L. (1998). *The communicative value of filled pauses in spontaneous speech*. PhD thesis, University of Birmingham.

[Shriberg, 1994] Shriberg, E. E. (1994). *Preliminaries to a theory of speech disfluencies*. PhD thesis, University of California.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associate, Inc.

[Tree, 1995] Tree, J. E. F. (1995). The effects of false starts and repetitions on the processing of subsequent words in spontaneous speech. *Journal of Memory and Language*, 34 :709–738.

[Tree, 2001] Tree, J. E. F. (2001). Listeners' uses of um and uh in speech comprehension. *Memory and cognition*, 29 :320–326.

[Young et al., 2017] Young, T., Hazarika, D., Poria, S., and Cambria, E. (2017). Recent trends in deep learning based natural language processing. *Computation and Language*.