Modèles neuronaux pour la simplification de parole, application au sous-titrage

François Buet

université
PARIS-SACLAY

Modèles neuronaux pour la simplification de parole, application au sous-titrage

Neural models for speech simplification, application to closed captioning

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la Communication (STIC)

Spécialité de doctorat : Informatique

Graduate School : Informatique et sciences du numérique, Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche Laboratoire interdisciplinaire des sciences du numérique (Université Paris-Saclay, CNRS), sous la direction de François YVON, Directeur de recherche

Thèse soutenue à Paris-Saclay, le 21 octobre 2022, par

François BUET

Composition du jury

Annelies BRAFFORT	Présidente
Directrice de recherche, LISN / CNRS, Université Paris-Saclay	
Christophe CERISARA	Rapporteur & Examinateur
Chargé de recherche, LORIA / CNRS, Université de Lorraine	
Benoit FAVRE	Rapporteur & Examinateur
Professeur, LIS / CNRS, Aix-Marseille Université	
Yannick ESTÈVE	Examinateur
Professeur, LIA, Avignon Université	
Thierry ETCHEGOYHEN	Examinateur
Principal Researcher, Vicomtech	
François YVON	Directeur de thèse
Directeur de recherche, LISN / CNRS, Université Paris-Saclay	

THESE DE DOCTORAT

NNT : 2022UPASG074

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Modèles neuronaux pour la simplification de parole, application au sous-titrage

Mots clés : Traitement automatique des langues, Sous-titrage, Simplification de parole

Résumé :

Dans le contexte linguistique, la simplification est généralement définie comme le processus consistant à réduire la complexité d'un texte (ou de paroles), tout en préservant au mieux son sens. Son application principale est de rendre plus aisée la compréhension et la lecture pour un utilisateur. Elle est entre autres une solution envisagée pour renforcer la lisibilité des textes auprès des sourds et malentendants (la surdité étant souvent à l'origine d'un retard dans l'apprentissage de la lecture), en particulier dans le cas du sous-titrage. Alors que les sous-titres interlinguistiques servent à diffuser les films et programmes dans d'autres langues, les sous-titres intralinguistiques sont le seul moyen, avec l'interprétation en langue des signes, par lequel sourds et malentendants peuvent accéder aux contenus audio-visuels. Or la vidéo a pris une place importante dans la société, que ce soit dans le contexte professionnel, récréatif, ou de l'éducation. Afin de garantir l'égalité des individus dans la participation à la vie publique et sociale, un certain nombre de pays dans le monde (dont la France) ont mis en œuvre des obligations légales concernant le sous-titrage des émissions télévisées. ROSETTA (RObot de Sous-titrage Et Toute Traduction Adaptés) est un projet de recherche collaboratif privé-public, qui se propose de développer des solutions technologiques d'accessibilité pour les contenus audiovisuels en français. La présente thèse, réalisée dans le cadre de ce projet, vise à étudier la simplification automatique de la parole par des modèles neuronaux, et à l'adapter au contexte du sous-titrage intralinguistique d'émissions télévisées en français. Nos travaux portent principalement sur l'analyse de méthodes de contrôle de longueur, l'adaptation de modèles de sous-titrage aux genres télévisuels, et l'évaluation de la segmentation des sous-titres. Nous présentons notamment un nouveau corpus pour le sous-titrage créé à partir de données recueillies au cours du projet ROSETTA, ainsi qu'une nouvelle métrique pour l'évaluation des sous-titres, *Sigma*.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title : Neural models for speech simplification, application to closed captioning

Keywords : Natural language processing, Closed captioning, Speech simplification

Abstract :

In the context of linguistics, simplification is generally defined as the process consisting in reducing the complexity of a text (or speech), while preserving its meaning as much as possible. Its primary application is to make understanding and reading easier for a user. It is regarded, inter alia, as a way to enhance the legibility of texts toward deaf and hard-of-hearing people (deafness often causes a delay in reading development), in particular in the case of subtitling. While interlingual subtitles are used to disseminate movies and programs in other languages, intralingual subtitles (or captions) are the only means, with sign language interpretation, by which the deaf and hard-of-hearing can access audio-visual contents. Yet videos have taken a prominent place in society, wether for work, recreation, or education. In order to ensure the equality of people through participation in public and social life, many countries in the world (including France) have implemented legal obligations concerning television programs subtitling. ROSETTA (Subtitling RObot and Adapted Translation) is a public-private collaborative research program, seeking to develop technological accessibility solutions for audio-visual content in French. This thesis, conducted within the ROSETTA project, aims to study automatic speech simplification with neural models, and to apply it into the context of intralinguistic subtitling for French television programs. Our work mainly focuses on analysing length control methods, adapting subtitling models to television genres, and evaluating subtitles segmentation. We notably present a new subtitling corpus created from data collected as part of project ROSETTA, as well as a new metric for subtitles evaluation, *Sigma*.

Remerciements

Je souhaite en premier lieu remercier mon directeur de thèse François Yvon, qui m'a guidé avec patience et bienveillance tout au long de ce fleuve pas toujours tranquille qu'est le doctorat. Merci pour sa confiance, sa grande disponibilité (malgré ses multiples obligations au laboratoire), et la qualité de son encadrement scientifique.

Je remercie ensuite Annelies Braffort, Christophe Cerisara, Benoit Favre, Yannick Estève et Thierry Etchegoyhen, qui m'ont fait l'honneur d'être les membres mon jury. Merci pour l'attention qu'ils ont porté à mon travail, ainsi que pour les commentaires et les analyses éclairantes qu'ils ont apportés dans les rapports et lors de la soutenance.

J'adresse également mes remerciements aux autres personnes avec lesquelles j'ai eu le plaisir de travailler au cours des quatre dernières années, en tant que co-auteur, collaborateur au sein du projet ROSETTA, ou dans le cadre d'enseignements. Merci notamment à : Alina Karakanta, Jitao Xu, Élise Bertin-Lemée, Josep Crego, Éric Florence, Jean-Luc Gauvain, Mauro Cettolo, Marco Turchi, Joël Falcou, Philippe Chatalic, Joël Gay.

Cette thèse a été effectuée au sein de l'équipe TLP, dans le département STL au LISN : je remercie l'ensemble des membres avec lesquels j'ai pu interagir (que ce soit pour discuter de TAL ou d'autre chose), de même que le personnel des services de soutien et support à la recherche, toujours réactif et efficace.

Un merci particulier à mes collègues doctorants ou stagiaires : Aina, Alban, Aman, Anh Khoa, Benjamin, David, Hicham, Hugues, Hussein, Jitao, Juan, Léa-Marie, Léo, Lufei, Marc, Margot, Mathilde, Maxime, Minh Quang, Natasha, Paul, Rémi, Sam, Sofiya, Ruiqing, Shu, Soyoung, Syrielle, Théo, Yajing, Yuming et `r". +"` [1]. De restaurants éthiopiens en fondues chinoises, sans oublier les nombreuses pauses café, leur bonne humeur et leur camaraderie ont été essentielles dans mon expérience de thésard.

Je pense maintenant aux amis d'avant ou d'ailleurs : de Nancy (le groupe de *L'Amicale du Local*, Aurélien, Clovis, Corentin, David, Florian, Grégoire, Guillaume, Olivier, Renaud, Stanislas...), de Marseille (le groupe de la MPSI3 du Lycée Thiers, Franck, Hugo, Zoé) et du pays d'Aigues (Corentin, Kévin). Nos réunions sont désormais espacées, mais je les attends toujours avec impatience.

Enfin merci à ma famille, pour ses conseils et pour la constance de son soutien et de son réconfort. À mon père, source d'un savoir absolument véridique sur l'histoire de nos

1. Que je n'oublie pas, en dépit des apparences, qui sont parfois trompeuses, quand même.

ancêtres des montagnes. À ma mère, nonobstant son habitude de toujours avoir raison. À mes sœurs Blanche et Sandra, qui n'ont jamais cessé de veiller sur leur petit frère.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

Dans le cadre linguistique, la *simplification* est généralement définie comme le processus consistant à réduire la complexité d'un texte (ou de paroles), tout en préservant au mieux son sens. Son application principale est de rendre plus aisée la compréhension et la lecture pour un utilisateur. Cette première définition est pour le moins imprécise, dans la mesure où différentes opérations peuvent être considérées pour arriver à ce résultat : remplacement des mots compliqués ou techniques (*simplification lexicale*), restructuration grammaticale des phrases (*simplification syntaxique*), schématisation des idées (*simplification conceptuelle*), redondance et explicitation des points clefs (*modification élaborée*), suppression des informations secondaires pour réduire la longueur et faire ressortir l'essentiel (*résumé* ou *compression*) (Siddharthan, 2014). Le type et l'intensité de la simplification à exécuter dépendent de l'utilisation visée et du public destinataire. La littérature comporte notamment des exemples d'application pour :

— les enfants (De Belder & Moens, 2010),
— les personnes apprenant une langue étrangère (Petersen & Ostendorf, 2007),
— les personnes atteintes d'un trouble affectant la lecture, comme certains cas d'autisme (Evans et al., 2014), d'aphasie (Carroll et al., 1999) ou de dyslexie (Rello et al., 2013),
— les personnes non familières avec un domaine technique (médical par exemple, Elhadad & Sutaria (2007); Siddharthan & Katsos (2010); Grabar & Hamon (2014)),
— les personnes sourdes ou malentendantes.

La surdité, lorsqu'elle intervient avant l'acquisition de la langue, est souvent à l'origine d'un isolement communicatif entraînant un retard dans l'apprentissage de la lecture (Hamm, 2008). Torres Monreal & Santana Hernández (2005) ont observé que pour un test standard sur le niveau de lecture, des élèves sourds espagnols en fin de collège obtenaient des scores comparables ou inférieurs à la valeur moyenne correspondant aux enfants de 7 ans ; des résultats qui sont cohérents avec ceux de Traxler (2000) aux États-Unis. Ainsi, quoiqu'il existe une variabilité importante de la maîtrise de la langue écrite au sein de cette population (en fonction de l'âge auquel est survenue la perte d'audition, de l'exposition à une culture tournée vers l'oral ou la langue des signes), la simplification est une

solution envisagée pour renforcer l'accessibilité des textes auprès des sourds et malentendants (Alonzo et al., 2020), en particulier dans le cas du sous-titrage (Daelemans et al., 2004).

Diaz Cintas & Remael (2007) définissent le sous-titrage de la manière suivante : « une pratique de traduction qui consiste à présenter un texte écrit, généralement sur la partie basse de l'écran, qui s'efforce de rapporter le dialogue original des locuteurs, ainsi que les éléments discursifs qui apparaissent à l'image (lettres, encarts, graffiti, inscriptions, pancartes, etc.), et les informations qui sont contenues sur la bande-son (chants, voix hors champ) ». Comme le montre la figure 1.1, l'affichage suppose une segmentation du texte en lignes et sous-titres (ou blocs), qui est soumise à des contraintes spatiales (le plus souvent fixées par des conventions officielles). Les sous-titres interagissent avec les paroles et l'image : les sous-titres doivent apparaître à une vitesse permettant la lecture par le spectateur, tout en restant synchrones avec l'information sonore et graphique.

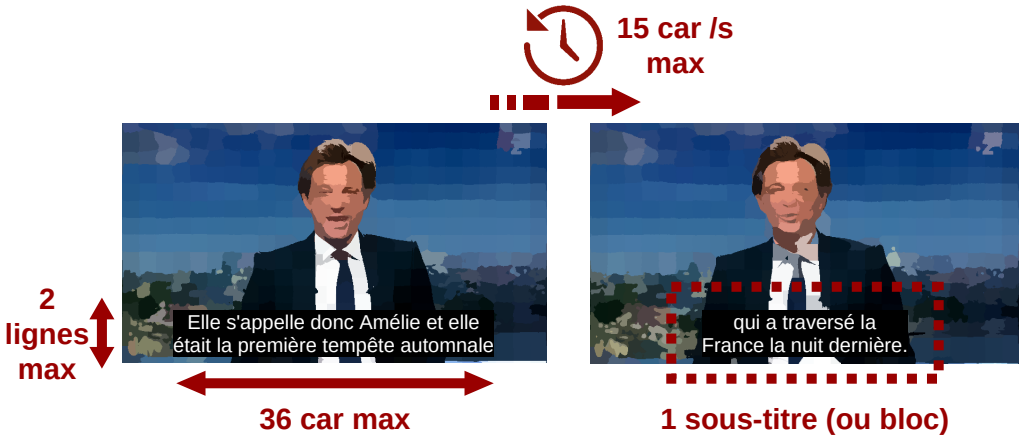


FIGURE 1.1 – Illustration des contraintes spatiales et temporelles pour les sous-titrage. La phrase « Elle s'appelle donc Amélie et elle était la première tempête automnale qui a traversé la France la nuit dernière. » est segmentée en deux sous-titres (parfois appelés *blocs* par commodité) de deux lignes chacun.

Alors que les sous-titres *interlinguistiques* servent à diffuser les films et programmes dans d'autres langues, les sous-titres *intralinguistiques* sont le seul moyen, avec l'interprétation en langue des signes, par lequel sourds et malentendants peuvent avoir accès aux contenus audio-visuels. Or, en conséquence des avancées dans les télécommunications, et de la généralisation des appareils d'enregistrement et des dispositifs d'affichage, la vidéo a pris une place importante dans la société, que ce soit dans le contexte professionnel, récréatif, ou de l'éducation. Afin de garantir l'égalité des individus dans la participation à la vie publique et sociale, un certain nombre de pays dans le monde ont mis en œuvre des obligations légales concernant le sous-titrage des émissions télévisées : section 508 du *Re-*

habilitation Act [1] et *21st Century Communications and Video Accessibility Act (CVAA)* [2] aux États-Unis, *Communications Act 2003* [3] et *Equality Act 2010* [4] au Royaume-Uni etc. En France la loi n° 2005-102 du 11 février 2005 pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées (2005-102) rend obligatoire l'accompagnement de sous-titres pour l'ensemble des programmes des chaîne dont l'audience moyenne annuelle dépasse 2,5 % de l'audience totale des services de télévision.

Le projet *ROSETTA* (RObot de Sous-titrage Et Toute Traduction Adaptés), dans le cadre duquel à été réalisée cette thèse, se propose de développer des solutions technologiques d'accessibilité pour les contenus audiovisuels en français. Plus précisément, ROSETTA cherche à automatiser la chaîne de production de sous-titres multilingues (français, anglais, espagnol et chinois) et à fournir une représentation en langue des signes française (LSF) de ces contenus par l'animation d'avatars virtuels, en utilisant les dernières avancées dans le domaine de l'intelligence artificielle. Ces objectifs se placent dans une optique de réduction des coûts de production pour les sociétés et organismes à l'origine de programmes vidéos partagés en ligne ou télédiffusés, et d'augmentation du volume de vidéos rendues accessibles. Un consortium de cinq partenaires collabore dans ce projet :

— **SYSTRAN** : Entreprise spécialisée dans le développement de logiciels de traduction ; chef de file du projet.
— **france.tv access** : Filiale du groupe France Télévisions, qui travaille sur la production des sous-titres pour sourds et malentendants.
— **MOCAPLAB** : Studio de services complets de capture et d'animation de mouvement.
— **LISN** : Laboratoire Interdisciplinaire des Sciences du Numérique. Les équipes de recherche du LISN couvrent des thématiques en relation avec les sciences du numérique, les sciences de l'ingénieur, l'intelligence artificielle et la science des données, l'interaction humain-machine, le traitement automatique des langues, et la bio-informatique. Nos travaux ont été menés dans l'équipe Traitement du Langage Parlé (TLP).
— **LUTIN/EPHE** : Laboratoire des Usages en Technologies d'Information Numériques. Le Lutin a comme objet d'études les systèmes cognitifs naturels et artificiels et leurs interactions pragmatiques et sémantiques.

Pendant la durée du projet, france.tv access a mis à disposition les vidéos et les sous-titres professionnels des émissions diffusées par France Télévisions, à travers une API

1. `https://www.fcc.gov/general/section-508-rehabilitation-act`
2. `https://www.fcc.gov/consumers/guides/21st-century-communications-and-video-accessibility-act-cvaa`
3. `https://www.legislation.gov.uk/ukpga/2003/21/section/303`
4. `https://www.legislation.gov.uk/ukpga/2010/15/section/29`

pour laquelle nous avons développé un service de requêtes. Ces programmes divergent par leurs formats (journaux, magazines, jeux, fictions etc.), leurs thèmes (politique, culture, santé etc.), et les situations d'énonciation qui produisent différentes formes de langue parlée, en fonction par exemple du niveau de préparation des prises de parole.

Cette thèse vise à étudier la simplification automatique de la parole par des modèles neuronaux, et à l'adapter au contexte du sous-titrage intralinguistique d'émissions télévisées en français. La suite de ce manuscrit est organisée de la façon suivante. Au **chapitre 2**, nous donnons un aperçu des méthodes et ressources pour la simplification automatique. Au **chapitre 3**, nous revenons sur le sous-titrage, ses enjeux, ses caractéristiques, et les méthodes et moyens disponibles pour l'effectuer automatiquement.

Au **chapitre 4**, nous examinons les méthodes de compression contrôlée, qui peuvent être utilisées pour adapter les sous-titres à la vitesse d'élocution ou à la capacité de lecture des usagers. Les principales questions traitées sont : Est-il possible de comprimer un texte en contrôlant précisément sa longueur ? La précision du contrôle a-t-elle une incidence sur la qualité du texte ? Nous y répondons en comparant les performances de plusieurs modèles de contrôle de longueur pour une tâche artificielle de compression et de décompression de phrase que nous avons conçue, et en analysant les représentations internes au cours du décodage d'un de ces modèles.

Au **chapitre 5**, nous présentons le corpus que nous avons assemblé avec la participation de france.tv access, et qui est destiné au sous-titrage d'émissions en français. Nous abordons notamment les questions suivantes : Quelles sont les caractéristiques des données de sous-titrage ? Quelles distinctions observe-t-on ? Peut-on mesurer la simplification opérée par les sous-titres ? Nous examinons les caractéristiques de cet ensemble de données, en étudiant en particulier l'influence du mode de sous-titrage et celle des genres télévisuels.

Au **chapitre 6**, nous présentons nos expériences sur le sous-titrage adapté aux genres télévisuels, avec des systèmes appris sur le corpus du chapitre 5. Les interrogations que nous traitons sont : Quelles techniques peuvent être mises en place pour adapter les sous-titres produits aux genres télévisuels ? Comment ces techniques se comparent-elles ? Nous montrons que les méthodes classiques d'adaptation au genre peuvent être transposées efficacement à la tâche de sous-titrage, et nous les comparons à une approche d'adaptation originale fondée sur le contrôle de longueur au cours du décodage. De plus, nous analysons de façon détaillée l'influence de certains facteurs sur les performances de nos systèmes.

Au **chapitre 7**, nous nous penchons sur la question de l'évaluation automatique des sous-titres, et plus particulièrement sur l'aspect de la segmentation. Nous considérons les questions suivantes : Quelles sont les différences entre les métriques automatiques pour

l'évaluation de la segmentation ? Comment comparer la segmentation de deux séquences qui n'ont pas le même contenu ? Afin d'y répondre, nous menons une étude parallèle des métriques de segmentation existantes, et nous introduisons un nouveau score *Sigma*, qui isole l'information liée à la segmentation indépendamment de la qualité du texte. Nous comparons *Sigma* aux métriques classiques en implémentant une méthode d'alignement et de projection de la segmentation.

Productions

Liste des productions scientifiques réalisées pendant la durée de la thèse :

Publications dans des conférences :

- Buet (2020) : Analyse de la régulation de la longueur dans un système neuronal de compression de phrase : une étude du modèle LenInit (RÉCITAL 2020).
- Buet & Yvon (2021b) : Vers la production automatique de sous-titres adaptés à l'affichage (TALN 2021).
- Buet & Yvon (2021a) : Toward Genre Adapted Closed Captioning (Interspeech 2021).
- Xu et al. (2022) : Joint Generation of Captions and Subtitles with Dual Decoding (IWSLT 2022).
- Karakanta et al. (2022) : Evaluating Subtitle Segmentation for End-to-end Generation Systems (LREC 2022).

Publication dans une revue :

- François Buet, François Yvon. Sous-titrage automatique : étude de stratégies d'adaptation aux genres télévisuels (TAL 63-1, *à paraître*).

Développements :

- Corpus ROSETTA pour le sous-titrage automatique.
- Outil EvalSubtitle [5] pour l'évaluation de la segmentation de sous-titres.

5. `https://github.com/fyvo/EvalSubtitle/`

Chapitre 2

La simplification pour les systèmes séquence-à-séquence

2.1 Introduction

Comme rappelé au chapitre 1, la *simplification automatique* peut être vue au sens large comme l'exercice qui vise à transformer des phrases écrites ou parlées afin de réduire leur complexité, tout en préservant autant que possible leur sens initial. Son application première est de faciliter la lecture et la compréhension. Les catégories de personnes pouvant en bénéficier sont multiples, et les besoins potentiels sont importants ; à titre indicatif, le *3e Rapport mondial sur l'apprentissage et l'éducation des adultes* (GRALE III) (UNESCO-UIL, 2017) indique que 758 millions d'adultes (dont 115 millions âgés de 15 à 24 ans) présentent un faible degré d'alphabétisation.

La complexité peut être considérée de plusieurs manières : au niveau du vocabulaire, de la structure syntaxique, des concepts mis en jeu, des connaissances extérieures présupposées, des choix orthographiques (dans le cas où les abréviations sont autorisées par exemple), de la quantité d'information dispensée, ou même simplement de la longueur en nombre de mots ou de caractères. Dans une partie de la littérature sur la complexité, les termes de *lisibilité* (c.-à-d. facilité de la lecture) et de *compréhensibilité* (c.-à-d. facilité de la compréhension) apparaissent souvent de façon concomitante, et les métriques dites « de lisibilité » (voir Section 2.4) servent parfois d'indicateurs pour la simplicité. Toutefois, même si les difficultés de lecture affectent naturellement la compréhension, certains auteurs insistent sur la distinction entre les deux notions, Telles (2018, p. 23) résumant : « alors que la lisibilité est passive et centrée sur le texte, la compréhensibilité est un concept plus large qui mesure l'interaction texte-lecteur ». Notons que la lisibilité dépend aussi d'éléments concernant l'apparence et la mise en forme du texte (taille et type de caractère, contraste, organisation sur la page etc.).

Avant d'examiner les méthodes et les ressources pour la simplification automatique, il convient d'aborder et de préciser notre position par rapport à plusieurs questions : celle

de l'importance accordée à la réduction de longueur (simplification/compression), celle des différences de modalités (source audio/textuelle), et celle de l'échelle de traitement (document/phrase).

2.1.1 Simplification p. opp. compression

Comme évoqué précédemment, la longueur d'un texte peut être source de complexité, en particulier si une partie du corps du texte est dédiée à des informations non-essentielles voire inopportunes, qui parasitent le sens principal. Dans cette situation, la suppression ou l'abrègement d'une partie des éléments peut apporter de la clarté à la signification, et constitue une forme de simplification. Toutefois, la réduction de la longueur d'un texte [1], ou compression [2], peut constituer un objectif en soi : pour le résumé par exemple (qui doit permettre à un utilisateur de juger le contenu d'un document rapidement), ou pour le sous-titrage (qui doit pouvoir être lu en synchronisation avec l'image ; voir Section 3.2.3). Désormais nous utiliserons le terme *compression* pour le traitement phrase par phrase, et *résumé* pour l'application au niveau d'un document.

Il faut noter que dans certains cas de figure, la simplification va à l'opposé de la compression ou du résumé. L'explicitation peut passer par l'ajout d'informations contextuelles, ou par l'usage de périphrases pour lever l'ambiguïté sur certains mots de vocabulaire (p. ex. « interprète en langue des signes » au lieu de « signeur »). Les règles *facile à lire et à comprendre* [3] (FALC) encouragent la redondance pour les informations importantes. Comme rappelé par Woodsend & Lapata (2011), une opération de simplification courante est la sous-segmentation en phrases, qui résulte en l'introduction de plus de mots.

Les plus anciens travaux sur la compression, qui reposaient le plus souvent sur l'apprentissage de grammaires d'arbres syntaxiques, se concentraient sur la suppression d'éléments (dans les parties basses des arbres), fonctionnant ainsi de manière *extractive* (Knight & Marcu, 2000; Turner & Charniak, 2005; McDonald, 2006). Ce genre d'approche avait l'avantage d'être efficace d'un point de vue calculatoire, autorisant une résolution par programmation dynamique, mais n'était évidemment pas optimal pour la conservation du sens et de la grammaticalité. Les travaux ultérieurs ont progressivement intégré les opérations de paraphrase (Cohn & Lapata (2008), utilisent par exemple une grammaire permettant les insertions, les substitutions et les réorganisations), se rapprochant ainsi da-

1. Toujours en conservant au mieux son sens.
2. Nous utilisons le terme « compression » pour désigner la réduction de longueur d'un texte, et non dans le sens de la diminution de la taille de stockage (un texte plus court nécessite effectivement moins d'espace mémoire, mais l'information est altérée).
3. `https://www.unapei.org/article/de-nouvelles-fiches-en-facile-a-lire-et-a-comprendre-falc-realisees-par-la-cnsa/`

vantage de la simplification. Étant donné cette proximité dans la littérature récente, nous inclurons par la suite des méthodes de compression de phrase parmi celles de simplification.

2.1.2 Simplification de parole p. opp. simplification de texte

La simplification de parole peut être appliquée au sous-titrage pour sourds ou malentendants, ou comme pré-traitement pour la génération de pictogrammes à partir de parole spontanée, dans le cadre de la communication alternative et augmentée (CAA) (Macaire et al., 2022). Du point de vue de l'implémentation, deux stratégies principales sont envisageables : l'architecture *en cascade*, qui combine un module de reconnaissance de parole et un module de simplification de texte, et l'architecture *directe* ou *bout en bout*, pour laquelle un seul modèle effectue la transformation complète. À notre connaissance, seuls des travaux en lien avec le sous-titrage intralinguistique (p. ex. Liu et al. (2020)) ont mis en œuvre des systèmes de simplification de parole bout en bout : nous en discuterons plus en détail au chapitre suivant, à la section 3.3.0.2. Notons également que le résumé automatique de parole a été utilisé pour faciliter l'accès à l'information dans des bases de données audio (Favre, 2007). Dans la suite de ce chapitre nous nous concentrerons sur la **simplification de texte**.

2.1.3 Simplification de document p. opp. simplification de phrase

Comme le fait remarquer Martin (2021, p. 17), la plupart des cas d'utilisation pour la simplification ont naturellement pour entrée des documents ; qu'il s'agisse d'instructions pour un médicament ou un appareil, d'articles de presse, de sous-titres de films ou d'émissions etc. Cependant, la majeure partie de la littérature s'est consacrée à la *simplification de phrase*. Celle-ci a l'avantage de pouvoir être évaluée plus facilement (la diversité des opérations pouvant être appliquées à l'échelle du document rend l'évaluation avec références encore plus difficile, tant il y a de façons de simplifier qui sont acceptables), et peut en outre être appliquée récursivement aux documents. Woodsend & Lapata (2011) mettent en place une simplification pour articles de Wikipédia fondée sur deux modules : un premier chargé d'attribuer un score aux phrases pour faire ressortir les plus importantes (un procédé courant pour la production de résumé), et un second réalisant une simplification des phrases sélectionnées avec une grammaire quasi-synchrone. Ce genre d'adaptation reste néanmoins assez loin des transformations que peuvent effectuer des humains sur un document. L'étude de Alva-Manchego et al. (2019) sur Newsela (corpus de simplification créé par des rédacteurs professionnels, voir Section 2.3) met en lumière les types de modification suivants : réorganisation des phrases, ajout d'information, jonction de phrases, résolution d'anaphore, et sélection de contenu. Dans ce chapitre et dans nos travaux, nous

nous focaliserons sur la **simplification au niveau des phrases**. Remarquons que dans le cas du sous-titrage, la réorganisation est proscrite, et l'ajout est limité à des informations non-verbales (p. ex. « [rires] » ou « [applaudissements] »).

2.2 Méthodes pour la simplification automatique de phrase

Outre son utilisation pour l'amélioration de la lisibilité et de la compréhensibilité, la *simplification de phrase* intervient dans certaines méthodes de production de résumé de document (Siddharthan et al., 2004), est un possible pré-traitement pour certaines tâches de *Traitement Automatique des Langues* (TAL) telles que l'analyse syntaxique (Chandrasekar et al., 1996) ou l'étiquetage de rôle sémantique (*semantic role labeling*) (Vickrey & Koller, 2008), et est aussi utilisable pour l'affichage adapté de texte sur un écran (Corston-Oliver, 2001). En dépit de la potentielle perte d'information, la régularisation de la syntaxe et la réduction à l'essentiel peuvent être bénéfiques dans des tâches de compréhension de la langue.

Les études posant les fondements de la simplification de phrase datent des années 1990 (Chandrasekar et al., 1996; Chandrasekar & Srinivas, 1997; Carroll et al., 1999). Les opérations communément admises sont la suppression, la paraphrase (qui équivaut à des opérations de substitution, d'insertion, ou de réorganisation), et la segmentation en sous-phrases. De façon générale, la simplification et la compression automatiques ont bénéficié des ressources et modèles développés dans le cadre de la paraphrase et de la traduction automatique.

2.2.1 Simplification lexicale

La simplification lexicale se focalise sur le remplacement de mots ou d'expressions courtes par des équivalents plus simples, en tenant compte du contexte de la phrase (Specia et al., 2012). Il s'agit d'une composante importante de la simplification de façon globale ; le vocabulaire peu fréquent ou trop technique peut être une entrave pour des lecteurs qui n'auraient pas autrement de difficulté avec le raisonnement ou les concepts sous-jacents. Un cas particulier qui a fait l'objet d'attention est celui de la substitution de termes médicaux par des synonymes du langage commun (p. ex. « mal de tête » pour « céphalée ») (Elhadad & Sutaria, 2007; Grabar & Hamon, 2014).

Deux étapes constituent habituellement la simplification lexicale : l'identification des mots complexes, et la proposition d'un remplaçant. Horn et al. (2014) ont par exemple extrait des règles de substitution (avec plusieurs candidats) à partir de phrases alignées entre *English Wikipedia* et *Simple English Wikipedia* (versions classique et simplifiée des

articles Wikipédia en anglais ; voir Section 2.3), en filtrant les mots suffisamment simples et en imposant des conditions sur les étiquettes parties du discours des paires potentielles. Un modèle hiérarchique à base de traits est chargé de classer les substituts potentiels de chaque mot : l'option de conservation est représentée par le mot lui-même.

2.2.2 Analogie avec la traduction

L'approche la plus courante dans les travaux récents a été d'inscrire la simplification de phrase dans le modèle de la traduction automatique, en considérant que la phrase source est formulée dans une « langue complexe », et que la phrase cible est écrite dans une « langue simple » (le même genre d'analogie pourrait être faite pour d'autres types de modification monolinguistique, comme le changement de style). Dès lors les techniques développées pour la traduction peuvent être appliquées, à condition de disposer pour l'apprentissage de données parallèles représentant la transformation attendue.

2.2.2.1 Traduction à base de segment

La *traduction automatique à base de segment* (*phrase based machine translation*, PBMT) a été utilisée pour mettre en œuvre la simplification, principalement à travers une série de systèmes fondés sur Moses (Koehn et al., 2007), et entraînés sur un corpus parallèle *English Wikipédia-Simple English Wikipedia*. Specia (2010) utilise assez simplement Moses pour effectuer une traduction de phrase complexe en phrase simple. Coster & Kauchak (2011a) poursuivent la démarche en ajoutant des opérations de suppression de mots. Wubben et al. (2012) se servent également de Moses pour réaliser une transduction monolinguistique ; ils réalisent ensuite une hiérarchisation des sorties selon leur dissemblance avec l'entrée. Hybrid, modèle créé par Narayan & Gardent (2014), effectue d'abord une segmentation et des suppressions en agissant sur une représentation sémantique structurée de la phrase, puis poursuit la simplification (pour les substitutions et la réorganisation notamment) avec la méthode de Wubben et al. (2012).

2.2.2.2 Traduction neuronale

Les travaux de Kalchbrenner & Blunsom (2013); Cho et al. (2014b); Sutskever et al. (2014); Bahdanau et al. (2015) ouvrent une nouvelle page dans le domaine de la traduction automatique, établissant le cadre des systèmes encodeur-décodeur fondés sur les *réseaux de neurones récurrents* (RNN), avec mécanisme d'*attention*. La première étude sur la compression neuronale, réalisée par Rush et al. (2015), n'utilise pourtant pas de RNN à proprement parler : le modèle de langue et de traduction est implémenté par un perceptron multicouche qui reçoit le plongement des derniers mots prédits, ainsi qu'un encodage de

la phrase initiale. Cet encodage prévoit bien toutefois un mécanisme d'attention, par la combinaison pondérée de représentations des mots d'entrée (pour être plus précis ces représentations sont des sac-de-mots sur le voisinage de chaque position).

Zhang et al. (2017) constatent que la simplification lexicale et la simplification syntaxique présentent une certaine complémentarité. La première détériore parfois la grammaticalité, et la deuxième ne parvient pas toujours à opérer de réelle simplification (reproduisant souvent à l'identique la phrase d'entrée). Ils introduisent donc un système en deux étapes qui combine les deux types de simplification : dans un premier temps, les mots complexes sont identifiés et remplacés (à l'aide des ressources en paraphrase de PPDB, voir Section 2.3) ; puis dans un second temps, un modèle séquence-à-séquence contraint encode la phrase dans un vecteur, et décode celui-ci en s'appliquant à conserver les mots récemment substitués. Zhang & Lapata (2017) soulignent également la tendance des modèles encodeur-décodeur de simplification à répéter la phrase source, et proposent une méthode d'apprentissage par renforcement pour favoriser une réécriture moins conservatrice, tout en maintenant la justesse et la fidélité par rapport au sens initial. La récompense à maximiser tient compte de la simplicité (via l'utilisation de la métrique SARI, voir Section 2.4), de la pertinence vis-à-vis de la phrase d'origine, et de la grammaticalité. L'optimisation des paramètres est réalisée par descente de gradient sur l'espérance négative de la récompense (et est sujette aux problèmes de variance fréquemment rencontrés avec l'apprentissage par renforcement).

Comme il peut être attendu que la simplification ou la compression prennent différentes formes, et soient appliquées avec différents degrés d'intensité, plusieurs travaux ont pris la direction de la génération contrôlée. Kikuchi et al. (2016) comparent quatre méthodes visant à contrôler explicitement la longueur de la sortie d'un système encodeur-décodeur. Deux de ces solutions, fixLen et fixRng, agissent pendant la phase de décodage, en régulant la longueur des phrases candidates du faisceau de recherche. Les deux autres, LenEmb et LenInit, ajoutent des paramètres entraînables au modèle, de manière à ce que celui-ci ait accès à une représentation extérieure de la longueur cible souhaitée. Dans le même esprit, Takase & Okazaki (2019) proposent un modèle *Transformer* pour le contrôle de longueur. Selon une démarche similaire à celle de Kikuchi et al. (2016), leurs systèmes LDPE et LRPE reçoivent un encodage de la longueur visée, qui n'est toutefois pas calculé grâce à un paramètre appris, mais grâce à une version modifiée de l'encodage positionnel conçu par Vaswani et al. (2017). Nous étudierons de manière plus précise ces méthodes au chapitre 4. Martin et al. (2020) expérimentent également avec le décodage contrôlé : ils fournissent à leur modèle *ACCESS* (à base de *Transformer*) des paramètres régulant le taux de compression, la quantité de ré-écriture, la complexité lexicale, et la complexité syntaxique. Dans l'ensemble de ces méthodes, le système est habitué à recevoir à l'entraî-

nement des valeurs de paramétrage correspondant à la réalité de la source et de la cible, alors qu'à l'inférence l'utilisateur est libre de fixer les consignes.

Globalement, les progrès importants réalisés dans le domaine de la traduction automatique (méthodes statistiques, réseaux récurrents, *Transformer*) ont successivement été adaptés à la question de la simplification automatique. Une difficulté fréquemment rencontrée par les systèmes issus de cette démarche est la tendance à recopier la phrase d'entrée ; et un certain nombre de travaux ont dû ajouter des mécanismes contraignants pour forcer leurs modèles de transduction à réaliser des changements. Un problème sous-jacent est que la simplification (comme la compression) requiert une compréhension sémantique du texte pour effectuer les bonnes opérations (p. ex. déterminer les éléments non indispensables pour la conservation du sens), quand la traduction peut a priori reposer davantage sur des éléments de surface comme la syntaxe ou le vocabulaire.

2.2.3 Modèles génératifs et représentations latentes

La simplification de phrase peut être aussi décrite du point de vue du transfert de style, dans la mesure où elle vise à reformuler la même signification avec un tour moins complexe (et plus concis parfois). La littérature sur le transfert de style propose généralement de travailler avec des représentations dans un espace continu, et d'y séparer le contenu sémantique des dimensions stylistiques. Ainsi Hu et al. (2017) utilisent un *auto-encodeur variationnel* avec des représentations *dissociées* (*disentangled*) pour engendrer des phrases dont les attributs (parmi lesquels la longueur) sont contrôlés. La séparation entre la représentation style et celle du fond sémantique est réalisée dans ce cas par apprentissage adverse.

Liu et al. (2019) s'éloignent des méthodes de type *dissociation* (*disentanglement*). Leur approche consiste à obtenir des modèles prédisant les attributs stylistiques et le contenu d'une phrase qui pourrait être générée à partir d'une certaine représentation. Le transfert de style est effectué en déplaçant itérativement la représentation z dans la direction qui rapproche (d'après le calcul du gradient des prédicteurs par rapport à z) la phrase potentielle des valeurs d'attributs et du contenu voulus.

L'idée de pouvoir séparer ce qui relève du fond et de la forme dans une représentation est intéressante du point de vue des possibilités de génération contrôlée ; elle a été appliquée avec un certain succès pour le traitement d'image (par exemple, la génération de portraits par Lample et al. (2017)). Cependant dans le domaine du traitement des langues, une dissociation complète paraît plus difficile à admettre (par rapport au cas de la compression, est-il concevable de dire toujours la même chose en utilisant un nombre arbitraire de caractères ou de mots ?). En outre, la nature discrète de l'espace des phrases rend difficile l'adaptation des techniques d'apprentissage usuellement associées à la *dissociation*, dont

des travaux récents remettent en cause l'utilité (voire l'effectivité) (Lample et al., 2019).

2.3 Ressources

English Wikipedia-Simple English Wikipedia L'alignement des articles *English Wikipedia*[4] (EW) et *Simple English Wikipedia*[5] (SEW) a été une des sources de données les plus populaires pour les études sur la simplification de texte (présentant notamment l'avantage de pouvoir être largement diffusée, en vertu de la licence CC-BY-SA). Le corpus *Parallel Wikipedia Simplification* (PWKP, ou parfois *WikiSmall*) a été créé par Zhu et al. (2010) en alignant automatiquement (en utilisant TF-IDF) les phrases provenant d'articles EW et SEW équivalents. Il contient approximativement 108K paires de phrases complexe-simple. Parallèlement et indépendamment, Woodsend & Lapata (2011) et Coster & Kauchak (2011b) ont réalisé un travail similaire. Par la suite, Zhang & Lapata (2017) ont formé *WikiLarge* en agrégeant à PWKP les ensembles de données que les autres auteurs avaient également constitués grâce à Simple Wikipédia. WikiLarge contient près de 300K paires de phrases. Plus récemment, de nouvelles versions de corpus EW-SEW ont été mises au point : *Wiki-Auto*, par Jiang et al. (2020), qui bénéficie d'un meilleur alignement que WikiLarge et qui contient environ 488K paires, et *WikiSplit*, par Botha et al. (2018), qui est axé sur les segmentations en sous-phrases (extraites à partir des révisions apportées aux articles).

Newsela Xu et al. (2015) expliquent qu'après inspection manuelle de PWKP, ils estiment qu'environ 50 % des données ne conviennent pas pour l'apprentissage de la simplification (les paires de phrases étant mal alignées, ou ne présentant pas de vraie simplification). Ils font aussi remarquer que le style encyclopédique ne se généralise pas très bien aux autres types de texte pouvant être rencontrés par la tâche. Cela les conduit à proposer le corpus *Newsela*[6], créé à partir de 1 130 articles qui ont été ré-écrits par des rédacteurs professionnels selon cinq niveaux de difficultés, destinés initialement à correspondre aux niveaux de lecture d'élèves de différentes classes (de 8 à 12 ans). Un ensemble de 394K paires de phrase alignées est notamment proposé par Jiang et al. (2020).

Gigaword *Gigaword* (Graff et al., 2003; Napoles et al., 2012) contient environ 9,5 millions d'articles de presse, issus de sources et de domaines divers. Dans leur étude, Rush et al. (2015) associent le titre et la première phrase de chaque article pour avoir un corpus d'entraînement (néanmoins, ils précisent n'utiliser que 4 millions de paires, dans la

4. https://en.wikipedia.org/
5. https://simple.wikipedia.org/
6. https://newsela.com

mesure où ils appliquent une série de filtres devant écarter les données non-adaptées).

Bases de paraphrases *Simple PPDB* a été conçue par Pavlick & Callison-Burch (2016) comme un sous-ensemble de la base de paraphrases PPDB (Ganitkevitch et al., 2013; Pavlick et al., 2015), ne regroupant que celles correspondant à une simplification. Plus précisément, Pavlick & Callison-Burch (2016) ont entraîné un modèle de régression logistique pour qualifier les paraphrases (mots ou expressions courtes) selon trois classes : simplification, complexification, ou inexacte. En testant leur système sur PPDB ils ont pu sélectionner un sous-ensemble de 4,5 millions de paires correspondant à des simplifications lexicales. D'autres ressources provenant du domaine de la paraphrase pourraient également être employées pour la simplification, comme la base *ParaBank* (Hu et al., 2019) (ensemble de paraphrases à l'échelle de la phrase, construit par traduction autour d'une langue pivot, avec un décodage contraint pour former des phrases différentes des entrées).

Corpus en français Notons que tous les corpus mentionnés jusqu'ici sont en anglais. En ce qui concerne le français, les ressources de simplification sont assez éparses :

— de Loupy et al. (2010) ont créé un ensemble de données pour la compression (extractive) de phrase contenant 8 432 paires (les phrases initiales proviennent de journaux français, et les compressions ont été rédigées manuellement).

— Mallinson et al. (2018) ont réalisé un corpus multilingue pour la compression, MOSS [7], qui inclut le français (les phrases initiales proviennent d'enregistrements d'institutions européennes, de discussions TED, et de journaux ; les compressions ont été faites par des volontaires). Les compressions, réalisées par des volontaires, ne sont toutefois pas toujours de bonne qualité.

— Une version synthétique de Newsela en français a été obtenue par traduction avec l'outil Google translate (qui réalisait un score BLEU de 35,6 sur le test anglais-français de WMT14) (Abdul Rauf et al., 2020).

— Le corpus *Alector* [8] (Gala et al., 2020) est un recueil de 79 textes littéraires (histoires, contes) ou scientifiques, appariés avec leur équivalent simplifié, qui ont été sélectionnés parmi des documents à disposition d'élèves d'école primaire (CE1, CE2, CM1).

7. https://github.com/Jmallins/MOSS/
8. https://alectorsite.wordpress.com/

2.4 Évaluation

Comme pour d'autres tâches de TAL, le jugement humain est le mode d'évaluation théoriquement préférable, quoique plus difficile à mettre en place. Pour la simplification il consiste généralement en deux notes sur une échelle de 5, qui rendent compte de la grammaticalité de la phrase produite, et sa fidélité par rapport au sens de la phrase initiale. Cette méthodologie peut éventuellement être complétée par le calcul du taux de compression (longueur de la phrase produite divisée par la longueur de la phrase originale).

Napoles et al. (2011) ont analysé les biais pouvant survenir lors de l'évaluation de la compression de phrase. Ils préconisent en particulier de respecter les points suivants :

— ne pas comparer des phrases engendrées avec des taux de compression visés différents,
— tester néanmoins les systèmes en visant différents taux de compression (s'ils peuvent intégrer ce genre de contrainte),
— préférer les corpus associant plusieurs références de compression pour chaque phrase.

D'une manière comparable, Martin (2021, p. 175) conclut qu'il existe bien un compromis entre la réduction de la complexité et la préservation du sens, et que l'évaluation de la simplification devrait tenir compte de ces deux dimensions.

2.4.1 Métriques de lisibilité

Les métriques de lisibilité évaluent la simplicité d'un texte en se fondant principalement sur deux critères : la complexité syntaxique, et la complexité lexicale. En général, ces deux aspects sont mesurés par rapport à des caractéristiques de surface, typiquement la longueur des phrases (pour rendre compte de la difficulté de la structure syntaxique) et la longueur des mots (les mots longs tendant à être plus rares, et aussi plus compliqués). Plusieurs formules combinant ces deux traits ont été créées au fil des années, avec plus ou moins de variations. Les plus utilisées sont *Flesch-Kincaid Grade Level* (FKGL) et *Flesch Reading Ease* (FRE) (voir ci-dessous) ; peuvent être cités également les indices classiques *Dale-Chall* (Dale & Chall, 1948), *Fog* (Gunning et al., 1952), *Lix* (Björnsson, 1968), *SMOG* (Mc Laughlin, 1969), et *Kwolek* (Kwolek, 1973). Un certain nombre de limitations ont été mises en avant concernant ces métriques : Shardlow (2014) note que ce genre de calcul ne serait pas indicatif pour une simplification remplaçant des anaphores courtes ou ajoutant des définitions, tandis que Telles (2018, p. 57) fait remarquer qu'elles favorisent la rédaction de phrases courtes, pouvant amener le texte a perdre sa cohésion.

Flesch Reading Ease et Flesch-Kincaid Grade Level FRE et FKGL mesurent la lisibilité en s'appuyant sur le nombre moyen de mots par phrase et sur le nombre moyen de syllabes par mot :

$$\text{FRE} = 206,835 - 1,015\left(\frac{\frac{total\ mots}{total\ phrases}}{}\right) - 84,6\left(\frac{\frac{total\ syllabes}{total\ mots}}{}\right) \tag{2.1}$$

$$\text{FKGL} = 0,39\left(\frac{\frac{total\ mots}{total\ phrases}}{}\right) + 11,8\left(\frac{\frac{total\ syllabes}{total\ mots}}{}\right) - 15,59 \tag{2.2}$$

L'échelle de FRE est typiquement 0 – 100 (une valeur élevée indique un texte simple), tandis que celle FKGL correspond en principe aux années scolaires (*grade*) du primaire au lycée, de 0 à 12 (une valeur élevée indique un texte complexe) [9]. FRE et FKGL doivent normalement être calculés à l'échelle d'un document (Graesser et al. (2004) spécifient un minimum de 200 mots). Kandel & Moles (1958) ont adapté les coefficients de FRE pour le français ; pour nos travaux des chapitres 5 et 6, nous utiliserons leur formule :

$$\text{FRE(Fr)} = 207 - 1,015\left(\frac{\frac{total\ mots}{total\ phrases}}{}\right) - 73,6\left(\frac{\frac{total\ syllabes}{total\ mots}}{}\right) \tag{2.3}$$

2.4.2 Métriques provenant d'autres tâches

Certaines métriques initialement développées pour la traduction ou le résumé automatique ont été adoptées pour la simplification et la compression. De façon globale, ces métriques peuvent être vues comme des distances dans un espace monolinguistique.

BLEU (Papineni et al., 2002) est une métrique standard pour la traduction automatique, qui calcule la précision *n*-gramme de la sortie du système par rapport à une ou plusieurs références et intègre une correction qui pénalise les sorties qui sont significativement plus courtes que la référence. Formellement, la métrique BLEU est définie comme suit (Hyp est l'hypothèse de sous-titrage, Ref est le sous-titre de référence produit par un expert, et

9. En théorie, à cause du terme dépendant de la longueur des phrases, FRE et FKGL n'ont respectivement pas de minorant et de majorant.

$|A|$ dénote le cardinal de l'ensemble A) :

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \text{ avec} \tag{2.4}$$

$$p_n = \frac{|\text{Ref} \cap \text{Hyp}|}{|\text{Hyp}|} \text{ et} \tag{2.5}$$

$$\text{BP} = \begin{cases} 1 & \text{si } |\text{Hyp}| > |\text{Ref}| \\ e^{1-\frac{|Ref|}{|Hyp|}} & \text{sinon} \end{cases} \tag{2.6}$$

La métrique BLEU ne peut être utilisée qu'au niveau de larges segments de textes et n'est pas indiquée pour évaluer des phrases isolées, pour lesquels les termes correspondants à la précision peuvent être nuls. Xu et al. (2016) ont montré que dans le cas de la simplification, BLEU corrèle les jugements humains pour le sens et la grammaticalité, mais pas pour la simplicité. Sulem et al. (2018a) abondent dans le même sens.

ROUGE (Lin, 2004) est le nom générique d'un ensemble de mesures développées pour l'évaluation de la production de résumé, qui calculent le rappel [10] entre la sortie du système et une ou plusieurs références pour des unités telles que les n-grammes (*ROUGE-n*), les plus longues séquences de mots communes (*ROUGE-L*, *ROUGE-W*), ou les « bi-grammes à trous » (*ROUGE-S*, *ROUGE-SU*). Certains auteurs ont employé ROUGE (R-1, R-2, R-L) pour évaluer la compression (Kikuchi et al., 2016; Takase & Okazaki, 2019). Cependant Ng & Abrecht (2015) mettent en avant deux problèmes avec ROUGE : (i) les paraphrases par rapport à la référence sont pénalisées, (ii) il n'y a pas de garde-fou en ce qui concerne la grammaticalité de l'hypothèse (en particulier pour R-1, R-2 et R-SU). La première critique est généralisable à l'ensemble des métriques fondées sur le recoupement d'unités, notamment dans le cadre de la simplification où la variabilité de la cible est importante ; à moins de disposer de plusieurs références de bonne qualité (ce qui est rare en pratique).

2.4.3 Métriques conçues pour la simplification

SARI (Xu et al., 2016) est une métrique pour la simplification de texte, qui profite de la qualité monolinguistique de la tâche pour mesurer la différence entre la source et la cible. SARI compare les opérations d'édition (insertion, copie, suppression de n-gramme) observées entre l'entrée et la sortie, avec celles observées entre l'entrée et les références [11]. La mesure SARI récompense le système lorsque ses insertions/copies/suppressions sont

10. Ou alternativement la F-mesure.

11. Il est important de noter que cette comparaison repose sur un recoupement des unités, et non sur un réel alignement entre hypothèse et références.

conformes à celles des références. Plus exactement, SARI équivaut à la moyenne des F-mesures calculées pour chacune des trois opérations d'édition (la précision et le rappel étant eux-même moyennés sur les ordres de n-grammes) :

$$\text{SARI} = \frac{1}{3}(F_{ins} + F_{cop} + P_{sup}), \tag{2.7}$$

$$F_{ope} = \frac{2 \times P_{ope} \times R_{ope}}{P_{ope} + R_{ope}}, \tag{2.8}$$

$$P_{ope} = \frac{1}{k} \sum_{n=[1,\ldots,k]} p_{ope}(n), \tag{2.9}$$

$$R_{ope} = \frac{1}{k} \sum_{n=[1,\ldots,k]} r_{ope}(n). \tag{2.10}$$

SAMSA (Sulem et al., 2018b) a été conçue pour mesurer la simplicité structurelle d'une séquence, sans avoir besoin de référence. SAMSA réalise une analyse sémantique des séquences source et cible selon le schéma *Universal Cognitive Conceptual Annotation* (UCCA) (Abend & Rappoport, 2013), afin d'identifier les « scènes » (une unité sémantique contenant une relation, pouvant décrire une action ou un état). Le système est récompensé si : (i) toutes les phrases dans la séquence cible ne contiennent qu'une scène, (ii) toutes les scènes dans la source sont conservées dans la cible. SAMSA est particulièrement adaptée pour estimer une transformation réalisant la sous-segmentation en phrases de la séquence d'entrée ; en cela elle est complémentaire à SARI qui convient pour évaluer les opérations de paraphrase. SAMSA a assez peu été utilisée dans la littérature jusqu'ici ; un frein potentiel est son implémentation complexe, qui pour l'heure et à notre connaissance n'a été réalisée que pour l'anglais (bien qu'elle soit en théorie adaptable à d'autres langues).

2.5 Conclusion

La simplification automatique de texte a majoritairement été explorée à l'échelle de la phrase ; une branche de la littérature s'étant en particulier concentrée sur la réduction de longueur (compression), notamment de façon contrôlée : nos expériences du chapitre 4 porteront sur l'étude de ce genre d'approche.

Actuellement, la plupart des travaux sur la simplification automatique envisagent une analogie avec la traduction, et reprennent les méthodes développées dans ce domaine, qui requièrent le plus souvent de grandes quantités de données parallèles pour l'apprentissage des modèles.

Or les données de bonne qualité sont relativement peu nombreuses pour la simplification : alors que les documents traduits sont courants sur internet, il est rare que coexistent

des versions d'un même texte différant par le degré de simplicité. Les corpus les plus fréquemment exploités dans la recherche sont l'alignement *English Wikipedia-Simple English Wikipedia* et Newsela.

Pour l'évaluation, la simplification de phrase suit la tendance générale en TAL de l'utilisation de métriques automatiques. Si certaines proviennent des domaines de traduction et de résumé, un processus de recherche de nouvelles métriques adaptées aux spécificités de la tâche (entre autres la variabilité des transformations possibles) est en cours.

Chapitre 3

Sous-titrage automatique

3.1 Introduction

3.1.1 Historique du sous-titrage

L'origine du sous-titrage est étroitement liée aux débuts du cinéma. Au temps du cinéma muet [1], des intertitres ou « cartons » étaient utilisés pour transcrire des morceaux de dialogue, donner des précisions contextuelles, ou introduire des scènes (les « titres », à proprement parler, étant les cartons donnant le nom des scènes). Les éléments nécessitant traduction étaient ainsi peu nombreux, si bien que le cinéma paraissait alors aux yeux de certains comme une nouvelle langue universelle [2]. Cette conception est sérieusement remise en cause par l'arrivée du cinéma parlant, technique inaugurée en 1927 par *Le Chanteur de jazz*. La question de l'internationalité, relativement étrangère aux films muets dont la communication reposait essentiellement sur l'image, se pose désormais de façon prégnante à l'industrie cinématographique, qui développe trois solutions : les versions multiples (nécessitant de rejouer les scènes dans les autres langues), le *doublage* (qui consiste en un ré-enregistrement des dialogues par post-synchronisation), et le *sous-titrage* [3].

Représentant la méthode de traduction audio-visuelle la plus économique, l'utilisation des sous-titres *interlinguistiques* (écrits dans une langue différente) va prendre de l'essor avec la diffusion des films parlants. Les sous-titres *intralinguistiques* (écrits dans la langue originale), destinés à rendre le média accessible au public sourd ou malentendant, ne connaissent en revanche dans un premier temps qu'un développement limité (les premières tentatives sont réalisées dans les années 1940 par l'acteur sourd Emerson Romero),

1. Le premier usage d'intertitres remonte à 1903, pour le film *La Case de l'oncle Tom* d'Edwin S. Porter (Marleau, 1982).

2. « Renouveau ! Renouveau ! Éternelle Révolution. Les derniers aboutissements des sciences précises, la conception de la relativité, les convulsions politiques, tout fait prévoir que nous nous acheminons vers une nouvelle synthèse de l'esprit humain, vers une nouvelle humanité et qu'une race d'hommes nouveaux va paraître. Leur langage sera le cinéma. » (Cendrars, 1926).

3. Le terme « sous-titre » est employé dans son sens moderne dès 1912 ; « sous-titrage » et « sous-titrer » se rencontrent à partir de 1923 (Marleau, 1982).

avant l'arrivée à la télévision du télétexte et des sous-titres optionnels (aussi appelés *sous-titres codés – closed captioning*, p. opp. aux *sous-titres ouverts – open captioning*, fixés dans l'image). En France, le sous-titrage télétexte est assuré par le système Antiope (Acquisition Numérique et Télévisualisation d'Images Organisées en Pages d'Écriture) de 1976 à 1995, date à laquelle il est remplacé par le système britannique Ceefax (dont dérive le *World System Teletext*, encore en usage en Europe actuellement).

3.1.2 Enjeux des sous-titres intralinguistiques

Les contenus audio-visuels ne peuvent être accessibles aux spectateurs sourds ou malentendants qu'à travers la traduction en langue des signes (Bigand et al., 2019) ou le sous-titrage intralinguistique. D'après l'enquête Handicap-Santé de 2008, 0,3 % de la population française (soit 182 000 personnes) serait atteinte de surdité complète, tandis que 11,2 % (soit 7 056 000 personnes) serait touchés par une forme de déficience auditive (Haeusler et al., 2014). Mais de façon plus générale, ce type de sous-titres peut également servir dans le cadre de l'apprentissage de la langue par des locuteurs étrangers, ou plus simplement dans un contexte qui n'autoriserait pas la diffusion du son.

D'un point de vue législatif, les chaînes françaises dont la part d'audience est supérieure à 2,5 % de l'audience totale des services de télévision doivent sous-titrer la totalité de leurs programmes, en application de la loi n° 2005-102 du 11 février 2005 pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées (2005-102), qui modifie la loi n° 86-1067 du 30 septembre 1986 relative à la liberté de communication (loi Léotard) (86-1067). Une convention conclue avec l'Arcom (anciennement CSA) fixe les proportions des programmes accessibles pour les chaînes représentant moins de 2,5 % des parts d'audience. En 2017, le volume cumulé de programmes accessibles pour l'ensemble des chaînes concernées s'élevait à 135 526 heures [4]. Au niveau européen, la directive « Services de médias audiovisuels » (2010/13/UE) incite les fournisseurs à intégrer le sous-titrage (ainsi que notamment la langue des signes et la description audio) parmi les moyens à mettre en œuvre pour l'intégration des personnes handicapées et des personnes âgées.

À côté de ces besoins réglementés, la demande de sous-titrage explose également sur Internet [5] pour d'autres types de contenus : cours en lignes (en particulier depuis la pandémie de Covid-19), vidéos de tutoriels, films promotionnels, etc. Sans compter le contenu

4. Le volume et la proportion de programmes sous-titrés en 2017 par les chaînes sont détaillés sur la page `https://www.csa.fr/Proteger/Garantie-des-droits-et-libertes/Les-droits-des-personnes-handicapees/Le-sous-titrage` (consultée le 03/07/22).

5. La directive (UE) 2016/2102 (UE 2016/2102) encourage les pays membres de l'Union européenne à rendre plus accessibles les sites internet et les applications mobiles des organismes du secteur public, sans toutefois mentionner explicitement le sous-titrage.

généré par les utilisateurs, dont la qualité intrinsèque est inférieure, mais dont l'importance en terme de quantité atteint un autre ordre de grandeur (à titre indicatif, 500 heures de vidéos sont mises en ligne sur Youtube chaque minute [6]).

Dans ce contexte, le besoin de disposer d'outils performants pour assister à la production de sous-titres, voire de les réaliser de manière entièrement automatique, est de plus en plus pressant.

3.2 Caractéristiques du sous-titrage

3.2.1 Contraintes sur la forme

Le sous-titrage implique de traiter un flux audio-visuel pour :

1. y associer automatiquement un contenu textuel qui rend fidèlement compte des informations présentes dans le flux audio : principalement une retranscription des contenus parlés, mais également une description des changements de locuteurs et des segments non-parlés (musique, évènements sonores) ;
2. afficher ce contenu textuel de manière synchrone avec l'image et le son.

Il s'agit de par nature d'un exercice très contraint. À l'affichage, les sous-titres doivent satisfaire des astreintes spatiales (les tronçons de phrases doivent rentrer dans la largeur de l'écran, sans trop obstruer la scène dont ils partagent le cadre) et temporelles complexes (le texte doit être approximativement synchronisé avec les paroles ou l'image, et doit rester affiché suffisamment longtemps pour la lecture). Diverses conventions et recommandations régissent également l'affichage et la position à l'écran des événements sonores, le signalement des changements de locuteurs, etc. Ainsi, il est couramment entendu que chaque sous-titre doit contenir au plus deux lignes, si possible équilibrées en taille, et que les césures entre lignes doivent préserver la cohérence des groupes syntaxiques, comme illustré dans les exemples suivants, fournis par Laks (1957) :

— **Bon :**
Je n'ai pas d'argent sur moi, // revenez demain.
— **Moins bon :**
Je n'ai pas d'argent // sur moi, revenez demain.
— **Mauvais :**
Je n'ai pas d'argent sur // moi, revenez demain.
— **Très mauvais :**
Je n'ai // pas d'argent sur moi, revenez demain.

Les diffuseurs utilisent en général des listes de normes ou de lignes directrices of-

6. Source : `https://blog.youtube/press/` (consultée le 04/07/2022).

POUR TOUS LES PROGRAMMES

— *Respect du sens du discours.*

— *Respect de l'image.* Le sous-titre, limité à deux lignes pour les programmes en différé et à trois lignes pour le direct, ne doit pas cacher, dans la mesure du possible, les informations textuelles incrustées ni les éléments importants de l'image.

— *Parfaite lisibilité.* Il est recommandé que les sous-titres se présentent sur un bandeau noir translucide et si possible avec des lettres ayant un contour noir, quel que soit le réseau et notamment en TNT.

POUR LES PROGRAMMES DE « STOCK » DIFFUSÉS EN DIFFÉRÉ

— *Temps de lecture approprié :* 12 caractères pour une seconde, 20 caractères pour deux secondes, 36 caractères pour trois secondes, 60 caractères pour quatre secondes. Les laboratoires seront incités à respecter ces critères avec une tolérance de 20 %.

— *Utilisation systématique du tiret pour indiquer le changement de locuteur.*

— *Placement du sous-titre au plus proche de la source sonore.*

— *Découpage phrastique sensé.* Lorsqu'une phrase est retranscrite sur plusieurs sous-titres, son découpage doit respecter les unités de sens afin d'en faciliter sa compréhension globale.

— *Respect des changements de plans.* Le sous-titrage doit se faire discret et respecter au mieux le rythme de montage du programme.

POUR LES PROGRAMMES DIFFUSÉS EN DIRECT OU SOUS-TITRÉS DANS LES CONDITIONS DU DIRECT

— *Distinction des intervenants* par l'indication de leur nom en début de prise de parole et l'usage de couleurs appropriées, notamment lorsque le programme fait intervenir plusieurs personnes dans un échange qui peut être confus.

— *Réduction du temps de décalage entre le discours et le sous-titrage* visant à ramener ce décalage en dessous de 10 secondes. Ne pas omettre une partie significative du discours sous prétexte de supprimer le décalage pris par rapport au direct, mais l'adapter éventuellement. Tous les propos porteurs de sens doivent être rapportés.

FIGURE 3.1 – Directives extraites de la *Charte relative à la qualité du sous-titrage à destination des personnes sourdes ou malentendantes* du CSA.

ficiellement définies. La figure 3.1 reproduit une partie de la charte de qualité établie par le CSA en 2011 (CSA, 2011). Ces directives concernent le sous-titrage destiné aux personnes sourdes et malentendantes, pour les émissions de télévisions françaises ; elles ne correspondent pas toutes nécessairement à celles appliquées dans d'autres contextes. Ainsi les instructions de TED (TED, 2021), qui visent un sous-titrage interlinguistique, préconisent une longueur de ligne maximale de 42 caractères (au lieu de 36 pour la télévision française), et une vitesse de lecture inférieure à 21 car/s (quand le CSA conseille entre 12 et 15 car/s).

On remarquera aussi dans la charte du CSA qu'une distinction est faite en fonction du contexte ou de la stratégie de production des sous-titres. Pour ce qui est des émissions de télévision, le sous-titrage intervient souvent en bout de chaine du processus de production

— Quand un sous-titre est plus long que 42 caractères, coupez-le en deux lignes.
— N'utilisez jamais plus de deux lignes par sous-titre.
— Gardez les lignes d'un sous-titre aussi proches que possible en longueur.
— Ne découpez pas les lignes au milieu de « groupes linguistiques ».
— Maintenez la vitesse de lecture à un maximum de 21 caractères / seconde.
— Comprimez les sous-titres au-delà de 21 caractères / seconde. Essayez de préserver le sens autant que possible.

FIGURE 3.2 – Directives extraites des *Subtitling Tips* de TED.

et est principalement réalisé selon deux modalités très différentes : *en direct*, pour les journaux d'information, les émissions de plateau ou les évènements retransmis en simultané ; *en différé* pour les émissions de jeux, les documentaires et les fictions. Dans le premier cas, des contraintes de temps réels sont critiques, et le sous-titreur [7] doit s'adapter à la spontanéité des prises de parole et plus généralement aux aléas du direct ; dans le second cas, il faut potentiellement faire face à une plus grande variété de la parole et des événements sonores à prendre en compte : chansons, rires, bruits d'ambiance, interventions en langue étrangère.

3.2.2 Passer de l'oral à l'écrit

Quoique le sous-titrage soit plutôt décrit en tant que *traduction audiovisuelle* (Diaz Cintas & Remael, 2007), du fait du besoin de synchronisation et d'adéquation entre l'information textuelle, l'image, et le son, il peut être vu de façon simplifiée comme une transcription de dialogue (les contraintes pratiques ont poussé la plupart des travaux sur l'automatisation à l'aborder de cette manière, voir Section 3.3.0.3). La génération de sous-titre se confronte donc notamment aux difficultés qui découlent des *différences entre oral et écrit*, auxquelles certains auteurs réfèrent par le terme de *diamésie* (qui plus généralement désigne les « différences de moyens utilisés pour communiquer » (Wüest, 2009)).

Unités

En premier lieu, il convient de souligner que les unités de l'écrit ne s'appliquent pas naturellement à la parole. La segmentation en mots n'est nullement apparente dans la continuité du phénomène, de même que la segmentation en phrases. Blanche-Benveniste (2010, p. 35) note : « L'usage conventionnel des signes de ponctuation n'est pas en rapport direct avec les phénomènes de la langue parlée qu'il est censé représenter. Par exemple, les points – démarcatifs les plus forts – notent plus une limite syntaxique de fin de phrase

7. Les méthodes les plus adaptées pour le sous-titrage en direct sont la sténotypie (saisie phonétique), la vélotypie (saisie syllabique), et la reconnaissance vocale, généralement mise en place avec un « perroquet » (opérateur qui répète les paroles dans un environnement sans bruit), et suivie d'une correction manuelle.

qu'une pause réelle. ». Ainsi dans une transcription, la ponctuation correspond davantage à une analyse syntaxique a posteriori qu'à une réalité dans le rythme du discours.

Hétérogénéité et intonation

La langue parlée présente une grande variabilité [8], offrant différentes façons de dire la même chose. D'un point de vue linguistique, des variations se trouvent au niveau de la phonologie (p. ex. prononciation ou non de la liaison), de la morphologie (pronom *on* ou *nous*), de la syntaxe (« *Quand venez-vous ?* » p. opp. « *Vous venez quand ?* ») ou du lexique (synonymes). Ces variations peuvent aussi être étudiées sous un angle extra-linguistique ; quatre dimensions sont alors communément reconnues (Gadet, 2021) :

— la variation dans le temps ou *diachronie* (français classique p. opp. français moderne),
— la variation géographique ou *diatopie* (Paris p. opp. Marseille),
— la variation sociodémographique ou *diastratie* (selon l'âge, la classe sociale etc.),
— la variation situationnelle/stylistique ou *diaphasie* [9] (discussion entre amis p. opp. échange professionnel).

Ajoutons que l'intonation peut avoir un rôle important dans la signification des paroles (pour marquer l'ironie ou l'interrogation par exemple), et est révélatrice de l'humeur du locuteur, de ses émotions. Ainsi les variations et la prosodie sont importantes pour la caractérisation de l'individu et de la situation de communication. Or il est difficile de rendre compte à l'écrit d'une partie de ces informations [10] : les prononciations atypiques, les accentuations, l'emphase, les pauses ne peuvent être retranscrites que par le moyen de conventions typographiques ou orthographiques ad hoc (Campione & Véronis, 2001), ou en tirant parti de la mise en forme (Schlippe et al. (2020) utilisent la largeur et la graisse du caractère pour représenter la vitesse et l'intensité sonore). De plus, il est habituellement attendu de l'écrit une certaine normalisation, un respect plus strict des codes syntaxiques standard [11], qui nécessite souvent une restructuration des énoncés. Le sous-titreur, ou adaptateur, a donc la tâche délicate de devoir composer avec les limitations de son canal de transmission pour rester le plus fidèle possible à la source orale (Marleau (1982) fait remarquer que « la parole est *sens* mais aussi *tonalité humaine* »).

8. Il n'est pas anodin que le sociolinguiste William Labov ait nommé variationnisme son cadre d'étude des propriétés des langues.
9. Dans laquelle peut rentrer la notion de *niveaux de langue*.
10. Blanche-Benveniste (2010, p. 35) : « L'écriture orthographique du français n'est pas adaptée à la notation des variations. »
11. Gadet (1997, p. 14) : « Quand à la syntaxe, on sait à quel point l'écrit est codifié. »

Scories

Enfin la langue parlée se démarque par la présence plus ou moins abondante de scories : répétitions, pauses remplies (p. ex. « *euh* »), faux départs, bégaiement etc. Ces éléments sont directement liés à la spontanéité de la parole et ne transposent pas (ou peu) à l'écrit, langue travaillée, pour laquelle ils seraient considérés comme du bruit.

3.2.3 Comprimer et simplifier

En plus de la transformation diamésique, il est fréquemment admis que le sous-titrage nécessite d'effectuer une compression, voire une simplification du contenu : en effet le temps de lecture visuelle d'un texte est en général sensiblement plus long que celui de sa perception auditive (Williams & Thorne, 2000) ; cela peut évidemment dépendre de la vitesse d'élocution, mais il s'agit d'une tendance renforcée dans le cas de sous-titres, puisque le spectateur doit aussi porter son attention sur le reste de l'image. De plus certains utilisateurs sont susceptibles de ne pas maîtriser parfaitement la langue écrite ; il peut s'agir par exemple de personnes ayant une autre langue maternelle, ou bien de personnes sourdes ou malentendantes locutrices de la langue des signes et pour qui l'écrit est assimilable à une langue étrangère (Torres Monreal & Santana Hernández, 2005). Dans le contexte de la lecture de texte sur internet, des études ont observé des bénéfices à la simplification lexicale ou syntaxique pour sourds et malentendants (Kushalnagar et al., 2018; Alonzo et al., 2020).

Cependant, contrairement aux contraintes évoquées à la section 3.2.1, le niveau de simplification ne fait pas l'objet de recommandations officielles très précises, et est même souvent un point de contention. Ainsi dans ses directives pour le sous-titrage (BBC, 2021), la BBC recommande spécifiquement de rester au plus proche de la parole (dans la mesure du possible), et d'éviter la simplification, afin de préserver l'information audio et de ne pas perturber les usagers lisant sur les lèvres [12]. Romero-Fresco (2009) analyse que trois groupes se distinguent autour de la question :

— les télédiffuseurs, qui préfèrent un sous-titrage proche du mot à mot (qui est aussi plus économique, car moins travaillé) ;
— les spectateurs sourds (ou plutôt les associations de sourds), dont l'avis est partagé, mais dont une partie est fermement opposée à la simplification qui pourrait être la cause d'une inégalité dans l'accès à l'information ;
— la communauté académique, dont les travaux tendent à être en faveur des sous-titres simplifiés.

12. « It is not necessary to simplify or translate for deaf or hard-of-hearing viewers. This is not only condescending, it is also frustrating for lip-readers. »

Puisque la vitesse de lecture est le facteur décidant si les sous-titres doivent être compressés ou non, Jensema (1998) a étudié l'appréciation de différentes fréquences d'affichage par un échantillon d'individus comprenant 205 sourds, 110 malentendants, et 262 entendants. Ses conclusions sont que la vitesse préférée en moyenne par l'ensemble des participants est d'environ 145 mots/min (8 700 mots/h, ou 12,1 car/s en considérant que la longueur moyenne des mots est de 5 lettres en anglais), et que le seuil à partir duquel l'affichage commence à être considéré comme trop rapide est aux alentours de 170 mots/-min (soit 10 200 mots/h, ou 14,2 car/s ; au Chapitre 5 nous observons dans nos données des valeurs supérieures pour la vitesse d'élocution et la fréquence d'affichage des sous-titres). De façon remarquable, il est observé que les sourds sont à l'aise avec des vitesses plus grandes que les personnes sans handicap auditif ; ce que Jensema (1998) met en relation avec le fait que les premiers sont également ceux ayant le plus l'habitude de lire des sous-titres.

Wieczorek et al. (2011) ont de leur côté réalisé une expérience comparant l'utilisation par plusieurs groupes – sourds, malentendants, sans handicap – de sous-titres transformés à différents degrés – transcription verbatim, texte standard exempt des marqueurs oraux (p. ex. hésitations, répétitions), texte simplifié. Leur résultats, qui reposent sur un examen du mouvement des yeux (durée de l'attention portée à la zone de texte, nombre de fixations, nombre d'allers et retours avec l'image principale) suggèrent un effort cognitif plus important pour la lecture de la transcription verbatim et du texte standard, en particulier pour les personnes sourdes. Toutefois un test de compréhension réalisé auprès des participants ne révèle pas d'incidence claire du type de sous-titres sur l'intelligibilité (à noter que les meilleurs scores ont été obtenus dans le cas verbatim).

L'expression « sourds et malentendants » masque en réalité une hétérogénéité dans le rapport des personnes avec l'oral et l'écrit, notamment selon le contexte social et l'éducation qui peuvent privilégier la langue nationale environnante ou la langue des signes comme moyen de communication et d'apprentissage (Braffort, 2016, p. 23). Et en conséquence l'aptitude à la lecture dans cette population est variable. En somme, il est vraisemblable que les attentes des utilisateurs en matière de compression/simplification sont très dépendantes de leur niveau de langue, ainsi que du type d'émission regardé (les vitesses d'élocution pouvant varier, par exemple entre un documentaire et un journal d'informations).

Le tableau 3.1 donne des exemples représentatifs du type de simplification mis en place pour les sous-titres de télévision en France : l'accent est davantage porté sur la contraction, par suppression (comme pour la fin de la deuxième transcription) ou reformulation (« je vous emmène visiter » remplacé par « nous visitons »), que sur l'explicitation (le retrait de « département » aurait même plutôt l'effet inverse). Notons aussi qu'il existe des pratiques

TR	Cette semaine, **je vous emmène visiter** la Normandie, **et** notamment **le département du** Calvados.
ST	Cette semaine, <eob> **nous visitons** la Normandie, <eol> notamment **le** Calvados. <eob>
TR	**Avec nous pour en parler** ce soir, **Jérôme** Fourquet, **vous êtes** directeur du département opinion de l'institut de sondages Ifop**, citons ce soir votre livre « Le nouveau clivage » publié aux éditions du Cerf**.
ST	**Nous recevons** ce soir **J.**Fourquet, <eob> directeur du département opinion <eol> de l'institut de sondages Ifop. <eob>

TABLE 3.1 – Exemples de simplification entre une transcription manuelle TR et des sous-titres associés ST, le premier issu d'un extrait de l'émission *Échappées belles*, et le second issu d'un extrait de l'émission *C dans l'air*. Les balises représentent la segmentation à l'affichage : saut de ligne au sein d'un sous-titre `<eol>` (*end of line*) et fin de sous-titre `<eob>` (*end of block*) ; nous prenons ici les symboles proposés par Karakanta et al. (2020b). Les différences entre TR et ST sont en caractères gras.

systématiques, telles que la réduction des prénoms à leur initiale.

3.3 Architectures pour l'automatisation

Dans cette partie nous nous intéressons aux modèles et aux stratégies pour la production automatique de sous-titres. À cet effet, nous allons également examiner des travaux portant sur la *traduction de parole* : comme souvent dans le traitement automatique des langues, les innovations initialement développées pour la traduction peuvent être adaptées aux tâches ayant des modalités comparables (dans le cas présent, le fait d'avoir une entrée audio et une sortie textuelle). Nous réalisons l'analogie suivante : là où la traduction doit transposer d'une langue à une autre, le sous-titrage (intralinguistique) doit simplifier et mettre en forme pour l'affichage (en particulier segmenter en lignes), potentiellement en plusieurs étapes. Sperber & Paulik (2020) recensent les méthodes et les problèmes rencontrés pour la traduction de parole. La figure 3.3 (reprise de leur article) représente graphiquement les différentes stratégies qu'ils identifient, et le tableau 3.2 exprime les équations de recherche associées. Le lecteur pourra aussi se référer à Sulubacak et al. (2020) pour un panorama sur le sujet de la traduction de parole (et sur la traduction multimodale en général).

3.3.0.1 Architecture en cascade

L'approche classique est celle de la *cascade*, qui consiste à juxtaposer plusieurs modules distincts pour effectuer la transformation attendue : dans un premier temps une représentation intermédiaire z (p. ex. une transcription, ou des états cachés) est engendrée

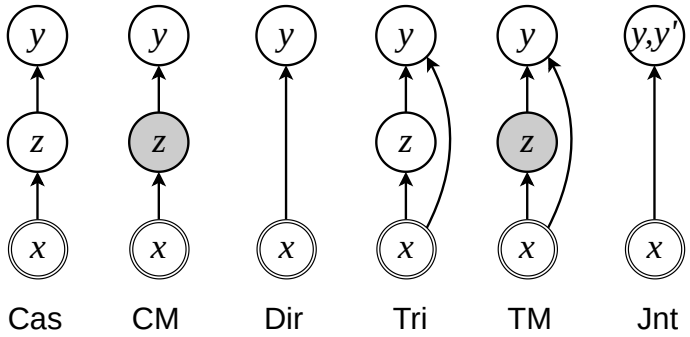


FIGURE 3.3 – Représentation graphique des stratégies pour la traduction de parole (ou le sous-titrage) : en cascade simple (Cas), en cascade avec marginalisation (CM), directe (Dir), en triangle simple (Tri), en triangle avec marginalisation (TM), jointe (Jnt). x est la variable observée (entourée par une ligne double), c.-à-d. la parole initiale ; z est une représentation intermédiaire, et dans certains cas une variable latente (disque grisé) marginalisée ; y est la traduction (ou les sous-titres) ; y' peut être la transcription ou une traduction dans une langue tierce dans le cas de la génération jointe.

grâce à un système de reconnaissance de parole, puis une sortie y est produite à partir de z en appliquant un modèle de traduction. Un certain nombre de travaux sur le sous-titrage automatique, dont les plus anciens menés dans le cadre des projets ATraNoS [13] (*Automatic Transcription and Normalisation of Speech*) et MUSA [14] (*Multilingual Subtitling of Multimedia Content*) (Daelemans et al., 2004; Piperidis et al., 2004; Vandeghinste & Pan, 2004), mettent en place ou supposent une stratégie en cascade qui comporte notamment un module de compression/simplification (Glickman et al., 2006; Prokopidis et al., 2008; Aziz et al., 2012; Angerbauer et al., 2019; Milde et al., 2021). La segmentation pour l'affichage, quand elle est abordée, est généralement aussi traitée par un module distinct (Piperidis et al., 2004; Matusov et al., 2019). Par exemple, Milde et al. (2021) recherchent la segmentation optimale par une recherche en faisceau maximisant une heuristique fondée sur la ponctuation, la longueur de segment, et la profondeur de rattachement dans l'arbre d'analyse syntaxique (pour éviter des coupures entre des mots étroitement liés syntaxiquement).

La stratégie en cascade présente l'avantage de pouvoir bénéficier des ressources considérables (données ou modèles) constituées indépendamment pour la reconnaissance et la traduction (pour le sous-titrage, il s'agirait des ressources pour la simplification, voir la Section 2.3). Sperber & Paulik (2020) mettent cependant en lumières quatre inconvénients :

— les *décisions précoces* (*early decisions* : dans l'incertitude entourant la génération de z, le système de reconnaissance fait un choix mal informé par rapport à la se-

13. `https://homes.esat.kuleuven.be/~spch/atranos/` (consulté le 04/07/22).
14. `http://sifnos.ilsp.gr/musa/` (consulté le 04/07/22).

Recherche directe (Dir)	$\arg\max_y P(y\|x)$
Recherche en cascade (Cas/CM)	$\arg\max_y \sum_{z\in Z(x)} P(y\|z)P(z\|x)$
Recherche en triangle (Tri/TM)	$\arg\max_y \sum_{z\in Z(x)} P(y\|x,z)P(z\|x)$
Couplage lâche (Cas/Tri)	$Z(x) = \{\arg\max_z P(z\|x)\}$
Couplage ferme (CM/TM)	$\|Z(x)\| > 1$
Recherche jointe (Jnt)	$\arg\max_y P(y,y'\|x)$

TABLE 3.2 – Équations associées aux stratégies de recherche pour la traduction de parole (ou le sous-titrage). x est la variable observée, c.-à-d. la parole initiale ; z est une représentation intermédiaire ; y est la traduction (ou les sous-titres) ; y' peut être la transcription ou une traduction dans une langue tierce dans le cas de la génération jointe ; $Z(x)$ est l'espace de marginalisation (inclus dans l'espace des représentation intermédiaires).

conde phase de traitement),

— la *propagation d'erreur* (*error propagation* : les erreurs de la reconnaissance se répercutent avec une incidence multipliée),

— la *perte d'information* (*information loss* : le système de traduction ne peut pas tirer parti de la prosodie par exemple),

— l'*inadéquation des domaines* (*mismatched domains* : les données d'apprentissage cibles pour la reconnaissance de parole de correspondent pas a priori aux données d'apprentissage sources pour la traduction ou simplification ; en particulier les premières contiennent des scories, comme expliqué à la Section 3.2.2).

Diverses solutions ont été proposées pour atténuer certains de ces problèmes. Les décisions précoces peuvent être contrées en marginalisant la représentation intermédiaire (CM dans la Figure 3.3) ; comme il serait impossible d'un point de vue calculatoire de traiter toutes les représentations intermédiaires, une approximation est de couvrir une partie de l'espace de z, avec par exemple les n meilleurs candidats, ou un treillis.

Une façon de limiter la propagation d'erreur est de rendre le système de traduction plus robuste, en dégradant artificiellement ses données d'apprentissage. Sperber et al. (2017) appliquent par exemple des opérations de substitution, d'insertion, et de suppression, avec des probabilités sur le vocabulaire et dans des proportions relatives qui reflètent les phénomènes observés en reconnaissance de parole.

Enfin des méthodes d'adaptation au domaine ont été explorées, telles que la re-segmentation

et la re-ponctuation des énoncés afin de les rapprocher de la forme habituelle du texte écrit (Matusov et al., 2006).

3.3.0.2 Architecture directe

Plus récemment, des modèles capables de suivre l'approche directe (ou bout en bout) ont vu le jour, le premier prototype fonctionnel ayant été implémenté par Bérard et al. (2016). Par conception, l'architecture directe échappe aux défauts majeurs associés à la stratégie en cascade (évoqués à la section précédente). La principale faiblesse liée au mode bout en bout est la sensibilité au manque de données. Pour pallier ce problème, les données pour la reconnaissance de parole et la traduction peuvent être intégrées par le biais de méthodes de *pré-entraînement* (les paramètres du modèle bout en bout sont initialisés avec ceux des modèles appris sur les sous-tâches) et d'apprentissage multitâche (les paramètres sont appris parallèlement sur la tâche bout en bout et sur les sous-tâches), avec des résultats équivalents (Bérard et al., 2018). Sperber & Paulik (2020) suggèrent cependant que l'utilisation de corpus séparés de reconnaissance et de traduction pourraient réintroduire les déficiences constatées pour les modèles en cascade, ce qui impliquerait un compromis sur le volume de données indépendantes à admettre.

Dans le cadre du sous-titrage, Liu et al. (2020) et Karakanta et al. (2020a) ont réalisé des systèmes fondés sur *Transformer* (de façon alternative, d'autres auteurs ont utilisé des LSTM (Jia et al., 2019)) et ont opté pour un apprentissage avec pré-entraînement. Pour les systèmes de Karakanta et al. (2020a), la segmentation d'affichage est effectuée durant la génération de la séquence de sortie : les frontières entre lignes étant matérialisées par des symboles spécifiques, insérés au préalable dans la partie cible du corpus d'apprentissage (de la même façon que dans les exemples du Tableau 3.1).

Du point de vue des performances, les études récentes laissent penser que si les données d'entraînement ne sont pas trop limitées, les systèmes bout en bout peuvent être compétitifs avec l'architecture en cascade (Sulubacak et al., 2020; Bentivogli et al., 2021). Toutefois, pour la campagne IWSLT 2021 sur la traduction de parole, l'architecture cascade obtient encore des résultats égaux ou supérieurs à l'architecture directe, même quand les corpus d'apprentissage excluent les données indépendantes de transcription et de traduction (Anastasopoulos et al., 2021). Si ces données indépendantes sont pleinement utilisées, alors l'architecture en cascade a l'avantage (Etchegoyhen et al., 2022).

3.3.0.3 Autres architectures

L'architecture triangulaire (Tri/TM dans la Figure 3.3 et dans le Tableau 3.2) est une option intermédiaire entre l'approche en cascade et l'approche directe : par rapport à

la première, une dépendance résiduelle avec la source ($P(y|x,z)$ au lieu de $P(y|z)$) est ajoutée, permettant de façon naturelle de pré-entraîner les modules de reconnaissance et de traduction sur des données séparées, puis de pratiquer l'apprentissage bout en bout sur la totalité (ou une partie) des paramètres. Anastasopoulos & Chiang (2018) mettent par exemple en œuvre la dépendance résiduelle au moyen d'une attention portée sur les états cachés de l'encodeur de parole.

L'architecture jointe est intéressante pour des cas d'application nécessitant d'engendrer des séquences cohérentes entre elles : par exemple, dans des pays ayant plusieurs langues officielles, il peut arriver que des sous-titres en langues différentes soient affichés simultanément. Une méthode d'implémentation possible est celle du décodage double (Xu & Yvon, 2021).

Il convient de noter que si nous avons vu jusqu'ici le sous-titrage au travers du paradigme de la traduction de parole, la nature de l'exercice devrait en théorie le rapprocher de la *traduction guidée par vidéo*, dans le cadre plus général de la *traduction automatique multimodale* (Specia et al., 2016). Cette tâche, longtemps freinée par l'absence de données libre d'accès, a reçu plus d'attention récemment grâce à la création de nouveaux corpus (voir Section 3.4). Une difficulté rencontrée dans ce mode de traduction est que la vidéo semble souvent fournir peu d'informations supplémentaires, n'apportant que des gains très faibles par rapport à des modèles ne prenant en entrée qu'une source textuelle (Wu et al., 2019; Yang et al., 2022). Par ailleurs la vidéo peut être utilisée pour le sous-titrage en post-traitement, pour réaliser placement des sous-titres à proximité du locuteur, comme proposé par Tapu et al. (2019).

Remarquons également que les sous-titres contiennent fréquemment des informations non verbales, par exemple : « [bruit de porte] », « [musique douce] » ou « [rires] ». Ces éléments pourraient être intégrés dans le sous-titrage automatique en utilisant les méthodes de description audio (Gontier et al., 2021).

3.4 Ressources

OpenSubtitles Du fait des droits associés aux émissions, films et séries, qui en général sont des freins à leur large diffusion, les corpus librement accessibles associant sous-titres et contenu audio-visuel sont relativement rares. Le corpus *OpenSubtitles* [15] (Lison & Tiedemann, 2016) fournit ainsi une grande quantité de sous-titres parallèles (1689 paires de textes, pour un cumul de 2,6 milliards de phrases) répartis sur 60 langues (la paire anglais-français contient par exemple 32,6 millions de phrases parallèles), mais n'y joint aucune vidéo.

15. Issu de : `https://www.opensubtitles.com/`

QED Les *formations en ligne ouvertes à tous* ou MOOCs (*massive open online courses*) sont une source de vidéos publiquement accessibles. Le corpus *QCRI Educational Domain* [16] (QED, anciennement QCRI AMARA) (Guzman et al., 2013; Abdelali et al., 2014) a été créé à partir de MOOCs sous-titrés par des volontaires sur la plateforme collaborative *Amara* [17] (Jansen et al., 2014). Sa version 1.4 contient 23,1K vidéos éducatives sur des sujets divers, dans une vingtaine de langues. Des sous-titres multilingues sont alignés pour une part substantielle des vidéos (la paire anglais-français correspond à 125K sous-titres).

How2 L'ensemble de données *How2* [18] (Sanabria et al., 2018) est destiné aux tâches multimodales de façon générale. Il rassemble près de 80K vidéos pédagogiques en anglais (2000 heures au total, pour une longueur moyenne de 90 secondes), portant sur des sujets variés (yoga, couture, cuisine etc.), récupérées sur Youtube avec certaines de leurs métadonnées : en particulier des sous-titres et un résumé en anglais écrits par le créateur de la vidéo. De plus, les sous-titres ont été alignés avec l'audio au niveau des mots, et également traduits en portugais par des volontaires grâce à une plateforme collaborative.

Corpus TED Les conférences TED [19] sont une autre source publique d'enregistrements audiovisuels, principalement en anglais, pour lesquels ont été produits manuellement des sous-titres intralinguistiques (en réalité assez proches de transcriptions mot à mot), et interlinguistiques dans 116 langues.

TED-LIUM [20] (Rousseau et al., 2012) est un corpus pour la reconnaissance de parole (constitué par alignement de l'audio et des sous-titres intralinguistiques) ; la version 3 correspond à 452 heures de parole transcrite.

Le corpus *Web Inventory of Transcribed and Translated Talks* [21] (WIT^3) (Cettolo et al., 2012) est une collection de sous-titres (intralinguistiques et interlinguistiques) provenant de TED (2086 conférences), créée dans le but de faciliter l'accès à ces données ainsi que leur utilisation. Toutefois cette ressource, qui permet l'alignement de phrases entre sous-titres multilingues, est plutôt destinée à la tâche de traduction automatique (le contenu audio-visuel n'est pas fourni, et les informations relatives à l'affichage des sous-titres, comme la segmentation en lignes et les indices temporels, n'ont pas été conservées).

Plus récemment, MuST-C [22] (*Multilingual Speech Translation Corpus*) (Di Gangi et al.,

16. https://alt.qcri.org/resources/qedcorpus/
17. https://amara.org/
18. https://github.com/srvk/how2-dataset
19. https://www.ted.com/talks/
20. https://lium.univ-lemans.fr/ted-lium3/
21. https://wit3.fbk.eu/
22. https://ict.fbk.eu/must-c/

2019) a été formé pour l'apprentissage de modèles de traduction de parole. La version initiale (v1.0) du corpus compile contenus audio et sous-titres pour 8 directions : anglais vers néerlandais, français, allemand, italien, portugais, roumain, russe, espagnol, avec entre 385 et 504 heures de parole pour chacune, soit entre 211K et 270K phrases cibles. Comme WIT[3], MuST-C ne contient pas les indications pour l'affichage des sous-titres.

MuST-Cinema [23] (Karakanta et al., 2020b) adapte MuST-C pour le sous-titrage automatique en réintroduisant la segmentation pour l'affichage dans les fichiers de sous-titres, sous la forme de balises `<eol>` et `<eob>`, entrelacées dans le texte, qui indiquent respectivement la fin d'une ligne et la fin d'un sous-titre. Les positions de frontières entre lignes et sous-titres ont été extraites des fichiers Subrip (`.srt`) originellement utilisés pour les vidéos de TED ; les indices temporels indiquant les périodes d'affichage n'ont cependant pas été récupérés.

3.5 Évaluation automatique

Différents aspects des sous-titres sont susceptibles d'être jugés, étant donné les nombreuses conventions qui régissent leur production. À ce jour, l'évaluation automatique des sous-titres concerne principalement les propriétés suivantes : qualité du texte, respect des normes de sous-titrage, qualité de la segmentation, synchronisation de l'affichage. Dans le cas des sous-titres intralinguistiques, l'adéquation du contenu textuel peut être mesurée avec les métriques pour la simplification de phrases (Section 2.4). Nous décrivons ci-dessous les métriques conçues spécifiquement pour la tâche de sous-titrage.

Respect des normes superficielles de sous-titrage L'affichage de sous-titres nécessite des informations précisant certains aspects de la présentation à l'écran, tels que la segmentation du texte en blocs (c.-à-d. sous-titres) et en lignes, la durée d'apparition de chaque bloc, la couleur des caractères, ou encore le positionnement horizontal des lignes. Ce formatage doit se conformer à des codes et des normes qui assurent la lisibilité des sous-titres, comme expliqué à la section 3.2.1. En particulier le nombre de caractères par ligne (CPL) et le nombre de caractères par seconde (CPS, calculé à partir de la durée d'affichage des blocs) sont d'ordinaire soumis à des recommandations (voir les Figures 3.1 et 3.2). Karakanta et al. (2020a) ont ainsi défini des métriques pour rendre compte du respect de ces contraintes, en accord avec les seuils fixés dans les *subtitling tips* de TED [24] :

— *CPL<42* : la proportion de sous-titres respectant la consigne de 42 caractères par ligne ;

23. `https://ict.fbk.eu/must-cinema/`

24. Ayant entraîné leurs modèles sur le corpus MuST-Cinema, qui est composé de conférences TED, Karakanta et al. (2020a) ont décidé de se référer à leurs directives.

— *CPS<21* [25] : la proportion de phrases respectant la consigne de 21 caractères par seconde.

Pour les besoins de nos travaux (Chapitres 5 et 6), nous avons été amenés à utiliser des variantes de ces mesures, adaptées aux normes en vigueur pour la télévision française :

— *CPL>36* : la proportion de lignes de sous-titres dépassant la consigne de 36 caractères par ligne ;
— *CPS>15* : la proportion de sous-titres dépassant la consigne de 15 caractères par seconde.

BLEU$_{br}$ Matusov et al. (2019) appliquent pour leur évaluation trois types de BLEU, l'un d'eux – noté L-BLEU – étant défini de la manière suivante : « *[Le score BLEU appliqué] au niveau des sous-titres, en marquant la coupure de ligne au sein d'un sous-titre par un symbole spécial BR, dans la sortie du système comme dans la traduction de référence.* » Dans le même esprit, Karakanta et al. (2020a) ont par la suite proposé BLEU$_{br}$ [26] : BLEU calculé au niveau des phrases, en marquant la segmentation dans l'hypothèse et la référence avec les balises `<eol>` et `<eob>` (voir Tableau 3.1). Ce score permet d'évaluer la qualité globale des sous-titres engendrés, prenant en compte le contenu textuel et la segmentation (la précision modifiée unigramme pénalise le système s'il produit trop de balises, les précisions d'ordre supérieur sont sensibles à la position en contexte des balises). Wilken et al. (2022) montrent dans leur étude que BLEU$_{br}$ présente une bonne corrélation avec le jugement humain. Nous analyserons plus en détail BLEU$_{br}$ au chapitre 7.

TER$_{br}$ Également proposé par Karakanta et al. (2020a), TER$_{br}$ correspond au calcul du *taux d'erreur de traduction* (TER) (Snover et al., 2006) entre la séquence de sortie du système et la séquence de référence, en masquant tous les mots à l'exception des balises de segmentation `<eol>` et `<eob>`. L'intuition est que les opérations comptées (insertion, suppression, substitution, translation de groupe) correspondront principalement aux erreurs de positionnement des balises ; permettant ainsi une évaluation centrée sur la segmentation (en vérité une différence du nombre de mots entre l'hypothèse et la référence apportera aussi une pénalité). Une analyse du comportement de TER$_{br}$ dans un scénario de dégradation de la référence est présentée au chapitre 7.

T-BLEU Matusov et al. (2019) définissent S-BLEU comme un score BLEU calculé au niveau des sous-titres ; une condition à son utilisation est donc que l'hypothèse et la ré-

25. Dans leur article, Karakanta et al. (2020a) utilisent simplement les notations CPL et CPS pour ces deux scores ; nous nous permettons de les changer ici pour lever l'ambiguïté sur les valeurs limites.

26. Dans leur article, Karakanta et al. (2020a) notent simplement BLEU ; nous préférons utiliser un nom moins équivoque.

férence doivent être alignées au niveau des sous-titres (blocs), ce qui est envisageable dans le cas d'une traduction à partir d'un patron, mais ne peut pas toujours être supposé vrai. *Timed BLEU* (T-BLEU) (Cherry et al., 2021) généralise S-BLEU, en alignant temporellement les mots de l'hypothèse avec les segments de la référence : chaque mot de la prédiction du système est associé à un indice temporel, suite à une interpolation à partir des temps de début et de fin d'affichage des sous-titres. Wilken et al. (2022) font remarquer que cette approche est sensible aux erreurs ou à l'imprécision de l'étiquetage temporel des sous-titres.

SubER *Subtitle Edit Rate* (Wilken et al., 2022) (SubER) est une métrique globale pour l'évaluation des sous-titres : elle mesure simultanément la qualité du texte, de la segmentation pour l'affichage, et de l'alignement temporel. Il s'agit d'un calcul TER appliqué au texte dans lequel sont intégrées les balises de segmentation, avec la contrainte supplémentaire que les opérations sont cloisonnées entre les sous-titres hypothèse/référence qui se chevauchent dans le temps. Wilken et al. (2022) réalisent une évaluation humaine par des sous-titreurs professionnels qui révèle que SubER corrèle assez bien l'effort de post-édition ainsi que l'estimation directe de la qualité des sous-titres.

3.6 Conclusion

Les sous-titres intralinguistiques sont indispensables pour garantir l'accessibilité des contenus audiovisuels, notamment pour les personnes sourdes et malentendantes, et leur forme est régulée par des normes définies officiellement.

La tâche de sous-titrage se caractérise par une conversion de l'oral vers l'écrit, ainsi que par une compression/simplification des paroles : la compression est contrainte par la *vitesse d'élocution* (puisqu'il faut que le texte puisse être lu en synchronisation avec l'information audio-visuelle), tandis que la simplification doit être adaptée à la *maîtrise de la langue écrite* par l'utilisateur. La variabilité de ces deux critères nous poussent à étudier des méthodes de compression contrôlée au chapitre 4. De même, les variations au sein de la langue orale (selon les individus ou les situation d'énonciation) sont une des raisons pour lesquelles nous nous orientons vers des systèmes de sous-titrage adaptés au chapitre 6.

De façon classique, la production automatique de sous-titres procède d'une architecture en cascade dans laquelle se suivent une étape de transcription par reconnaissance de parole, une étape de simplification et d'élimination des scories, et une étape de mis en forme pour l'affichage. Cependant, la stratégie bout en bout a gagné en compétitivité ces dernières années, au point d'être désormais une alternative valable dans les cas où les données d'apprentissage sont disponibles en quantité suffisante.

Également de façon récente, plusieurs nouvelles métriques automatiques pour l'évaluation des sous-titres ont été proposées ; cela témoigne d'un besoin de prendre en compte les différents aspects sur lesquels peut être jugée la qualité du sous-titrage. Au chapitre 7 nous nous pencherons notamment sur la question de l'évaluation de la segmentation pour l'affichage.

Chapitre 4

Contrôler la complexité par la longueur

4.1 Introduction

La tâche de simplification vise à rendre plus aisée la lecture et la compréhension d'une texte, et de façon générale, elle est distincte de la tâche de compression dont le but est la réduction de la longueur de la phrase, à travers des suppressions qui induisent inévitablement une perte d'information. Dans certains cas, au contraire de la compression, la réduction de la complexité demande d'allonger la phrase initiale afin de la rendre plus intelligible (par la redondance, l'explicitation ou la sous-segmentation en phrases, comme expliqué à la Section 2.1.1). Néanmoins, il est souvent attendu que la phrase simplifiée soit plus courte que l'entrée, la longueur pouvant être considérée comme un indicateur superficiel de la complexité, comme cela apparaît dans le calcul des métriques SMOG (Mc Laughlin, 1969), FRE (Flesch, 1948) et FKGL (Kincaid et al., 1975). Dans bien des applications, il est intéressant qu'un système séquence-à-séquence soit capable de contrôler la réduction de la longueur. En effet, si le but final est d'améliorer la lisibilité d'un texte pour des usagers n'ayant pas tous la même maîtrise de la langue, un contrôle lâche de la longueur est un moyen simple d'adapter le niveau de lecture. Dans le cadre du sous-titrage automatique, un contrôle plus strict pourrait être utilisé pour respecter les contraintes de temps de lecture et de largeur du moniteur (Aziz et al., 2012).

Dans ce chapitre, nous nous intéressons aux méthodes de contrôle de longueur *fondées sur l'apprentissage*, qui au cours de la phase d'entraînement conditionnent le modèle selon la valeur d'un attribut qui peut être fixée à dessein par l'utilisateur pendant la phase de test (par opposition aux méthodes *fondées sur le décodage*, qui ne modifient pas la façon dont est appris le modèle). Dans le cadre des modèles encodeur-décodeur RNN, Kikuchi et al. (2016) ont proposé deux approches pour la compression *LenInit* et *LenEmb* [1] : LenInit contrôle la phrase engendrée en introduisant dans l'état initial du décodeur un vecteur qui encode la longueur visée, tandis que LenEmb ré-introduit à chaque étape du décodage un vecteur encodant la longueur restante. Avec l'architecture *Transformer*, Takase

1. Code disponible à l'adresse `https://github.com/kiyukuta/lencon`.

& Okazaki (2019) ont avancé les méthodes *LRPE* et *LDPE* qui modifient les formules de *l'encodage positionnel*, afin d'associer à chaque mot soit sa position par rapport à la longueur visée (LRPE), soit sa distance à la longueur visée (LDPE). Afin d'avoir une meilleure compréhension des mécanismes impliqués par la compression de phrase, notamment la modélisation de la longueur dans les systèmes séquence-à-séquence, et dans la continuité des travaux de *sondage* (*probing*) des modèles neuronaux (Shi et al., 2016; Adi et al., 2016; Conneau et al., 2018), nous avons décidé d'analyser la méthode LenInit, ainsi que dans une moindre mesure LenEmb, LRPE et LDPE, en tentant d'estimer ses limites, et en essayant de comprendre les changements qu'elle implique par rapport au fonctionnement d'un décodage classique.

Pour cette étude, nous avons créé un corpus artificiel de compression de phrase, et l'avons utilisé pour entraîner une ré-implémentation au niveau caractère de LenInit (Section 4.2.3). Puis nous avons mené trois groupes d'expériences. Premièrement, nous avons effectué des mesures sur la précision du contrôle de longueur par LenInit, LenEmb, LRPE et LDPE, et sur la qualité des phrases produites (Section 4.3.1). Deuxièmement, nous avons entraîné un classificateur pour prédire la longueur future (c.-à-d. le nombre de caractères à engendrer avant la fin de phrase) à partir d'un état caché du décodeur LenInit, de manière à suivre l'évolution de la représentation de la longueur au cours du décodage (Section 4.3.2). Troisièmement, nous avons tracé l'évolution de la probabilité associée par le modèle aux caractères associés à la fin de phrase (Section 4.3.3). Notre analyse montre que les méthodes examinées n'ont pas toutes la même précision en pratique, certaines comme LenInit exerçant plutôt un contrôle probabiliste sur la longueur ; elle suggère aussi la coexistence de deux influences distinctes au cours du décodage : celle de l'objectif explicite, et celle du modèle de langue du décodeur.

Nos contributions pour ce chapitre sont ainsi :

— Création d'un corpus artificiel pour la compression et la décompression de phrases.
— Comparaison de la performance des modèles de contrôle de longueur LenInit, LenEmb, LRPE et LDPE, ainsi que de deux variantes proposées LenInit2 et LenInit3.
— Analyse de la représentation interne de la longueur au cours du décodage pour LenInit.
— Observation de l'évolution de la probabilité associée aux symboles menant à la fin de phrase au cours du décodage pour LenInit.

4.2 Contexte

Nous décrivons ici les modèles et les données que nous avons utilisés pour nos expériences de compression de phrase au niveau des caractères.

4.2.1 Contrôle de la longueur dans un modèle RNN

Le cadre des systèmes *encodeur-décodeur* avec *réseaux de neurones récurrents* (*RNN*) présente un processus en deux phases pour la *transduction* de séquences. D'abord, le côté encodeur reçoit une *séquence d'entrée x* (de longueur l_x) et produit de façon récursive une séquence *d'états cachés* $(h_1, \ldots, h_{l_x})$. Puis, le côté décodeur reçoit les états cachés de l'encodeur et engendre, toujours récursivement, une *séquence de sortie* $\hat{y}$ (de longueur $l_{\hat{y}}$). À chaque étape j de cette seconde phase, le décodeur calcule un *vecteur de contexte* c_j comme une combinaison pondérée et normalisée des états cachés de l'encodeur (ce qui est désigné par *mécanisme d'attention* [2]) et l'utilise pour la mise à jour de son propre état caché courant s_j :

$$s_j = f(s_{j-1}, \hat{y}_{j-1}, c_j), \tag{4.1}$$

où $\hat{y}_{j-1}$ est l'unité engendrée à l'étape précédente, et f est une fonction non-linéaire. s_j est à son tour utilisé pour calculer la distribution de probabilité associée à la prochaine unité :

$$p_{decoder}(\hat{y}_j | \hat{y}_{[1;j-1]}, x) = \text{softmax}(g(s_j, \hat{y}_{j-1}, c_j)), \tag{4.2}$$

où g est une fonction non-linéaire.

Développées par Kikuchi et al. (2016), les méthodes *LenInit* et *LenEmb* sont fondées sur ce cadre. Nous les avons ré-implémentées, en les adaptant pour la transduction au niveau des caractères.

LenInit encode la longueur de sortie voulue l en la multipliant (de façon scalaire) avec un paramètre appris, le vecteur de longueur V. Cet encodage continu est ensuite introduit au sein du premier état caché du décodeur s_0, comme suit :

$$s_0 = \tanh\left(W_{init}\left[\frac{\sum_{i=1}^{l_x} h_i}{l_x}; L_{param}\right]\right), \quad L_{param} = l \times V, \tag{4.3}$$

où W_{init} est un paramètre appris. Pendant la période d'entraînement, l est égal à la longueur de la séquence cible de référence, l_y. Pendant la période de test, l est fixé par l'utilisateur : dans nos expériences, $l = r \times l_x$, avec r le *taux de compression* visé.

LenEmb introduit une dépendance à la longueur restante l_j à chaque étape du décodage, l'équation (4.1) devenant :

$$s_j = f(s_{j-1}, \hat{y}_{j-1}, l_j, c_j) \tag{4.4}$$

En pratique un plongement $L_{plong}(l_j)$ est réalisé à l'aide d'une matrice apprise $W_{init} \in$

2. Dans des versions antérieures de modèles séquence à séquence RNN (Kalchbrenner & Blunsom, 2013; Cho et al., 2014b; Sutskever et al., 2014), le vecteur contexte dont la taille est fixée agrège les états de l'ensemble de la séquence, menant à une limitation (ou goulot d'étranglement) sur la longueur de celle-ci.

$\mathbb{R}^{d \times l^{max}}$ (d étant la dimension de plongement, et l^{max} la longueur maximale envisagée), puis concaténé au plongement de $\hat{y}_{j-1}$. Initialement la longueur restante est égale à la longueur visée : $l_1 = l$; puis à mesure du décodage, les longueurs des unités produites lui sont retranchées (comme nous travaillons au niveau des caractères, dans notre cas il s'agit d'une simple décrémentation à chaque étape). Comme pour LenInit, $l = l_y$ pendant l'apprentissage, et l'utilisateur fixe la valeur l lors de l'inférence.

Nous proposons aussi des variantes de LenInit, intermédiaires avec LenEmb, désignées comme LenInit2 et LenInit3, qui se distinguent par la façon de définir L_{param} dans l'équation (4.3) :

$$\text{LenInit2 :} \quad L_{param} = [l \times V; L_{plong}(l)], \quad \text{LenInit3 :} \quad L_{param} = L_{plong}(l), \tag{4.5}$$

où $L_{plong}(l)$ est le *plongement* associé à l par une table apprise. L'encodage par la norme de LenInit a en théorie l'avantage de pouvoir s'appliquer pour toute valeur, alors qu'un plongement classique (qui apprend indépendamment un vecteur pour chaque valeur) est limité par la fréquence dans les données d'entraînement de la longueur considérée. Avec ces variantes nous avons voulu vérifier le bénéfice lié à un encodage continu, et voir s'il implique une perte pour la précision du contrôle de longueur.

4.2.2 Contrôle de la longueur dans un modèle Transformer

L'architecture *Transformer* (Vaswani et al., 2017) repose également sur la dualité encodeur-décodeur ; comme pour le RNN, l'encodeur prend en entrée une séquence x de longueur l_x et produit une séquence d'états cachés $(h_1, \ldots, h_{l_x})$, puis le décodeur engendre une séquence de sortie $\hat{y}$ de longueur $l_{\hat{y}}$ à partir des états cachés de l'encodeur. Cependant pour le *Transformer*, les états cachés sont calculés parallèlement par un nombre prédéfini de couches de transformation, entre lesquelles intervient le mécanisme d'attention, devenu prépondérant : entre deux couches, le décodeur porte attention sur les états finaux de l'encodeur, mais aussi encodeur et décodeur portent attention sur leurs *propres états*[3] (*self-attention*). La version de base du *Transformer* utilise de plus un *encodage positionnel* qui permet de différentier les positions dans la séquence. Cet encodage des positions est combiné avec le plongement de chaque unité de l'entrée (dans la partie encodeur) ou de l'amorce de séquence produite (dans la partie décodeur) selon les formules :

$$\text{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_m}}\right), \quad \text{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_m}}\right), \tag{4.6}$$

3. Pour le décodeur, qui doit toujours engendrer récursivement la séquence cible, l'attention propre ne peut suivre que les états associés aux unités déjà engendrées.

où *pos* est la position de l'unité dans la séquence, et $2i$ (resp. $2i+1$) correspond aux dimensions paires (resp. impaires) de l'encodage.

L'encodage positionnel est important dans *Transformer*. Dans un RNN, la notion de localisation des éléments se retrouve dans la relation de récurrence (Équation (4.1)), qui induit une proximité entre les états cachés de positions voisines. Le fonctionnement de mise à jour parallèle des états dans *Transformer* n'induit pas de tel phénomène : l'encodage positionnel est seul élément qui porte l'information des positions relatives entre les unités.

Pour contrôler de manière explicite la longueur des sous-titres produits par certains de nos modèles, nous avons aussi ré-implémenté les variantes LRPE et LDPE, proposées par Takase & Okazaki (2019) et utilisées dans un cadre de sous-titrage également par Lakew et al. (2019). Ces encodages intègrent une consigne sur la longueur l du texte à produire. Cette contrainte peut être exprimée comme un ratio de compression entre entrée et sortie (LRPE) ou bien encore comme une différence relative entre la position courante *pos* et la fin attendue de la sortie (LDPE). Formellement, ces contraintes prennent la forme suivante :

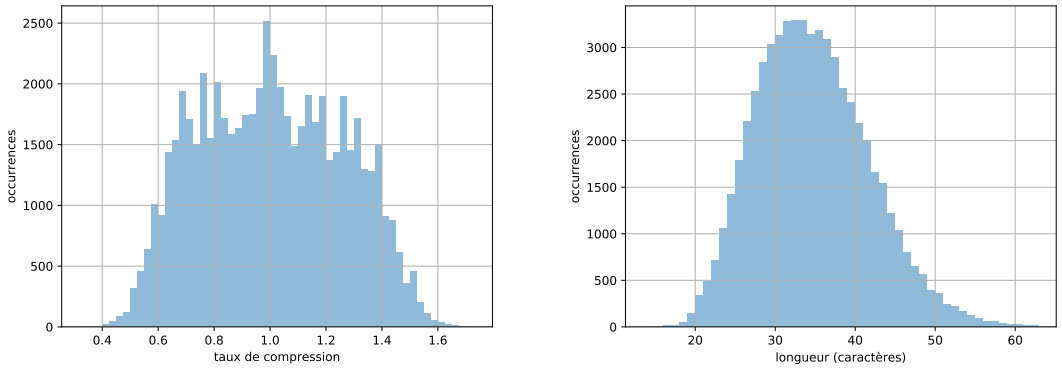
$$\mathrm{LRPE}_{(pos,l,2i)} = \sin\left(\frac{pos}{l^{2i/d_m}}\right), \qquad \mathrm{LRPE}_{(pos,l,2i+1)} = \cos\left(\frac{pos}{l^{2i/d_m}}\right), \tag{4.7}$$

$$\mathrm{LDPE}_{(pos,l,2i)} = \sin\left(\frac{l-pos}{10000^{2i/d_m}}\right), \quad \mathrm{LDPE}_{(pos,l,2i+1)} = \cos\left(\frac{l-pos}{10000^{2i/d_m}}\right). \tag{4.8}$$

l est égal à la longueur de la séquence cible de référence pendant la période d'entraînement, mais est fixé par l'utilisateur pendant la période de test (dans nos expériences, $l = r \times l_x$, avec r le *taux de compression* visé). LRPE caractérise à la fois la position courante *pos* et la longueur totale souhaitée l, tandis que LDPE exprime une distance à l'objectif de longueur.

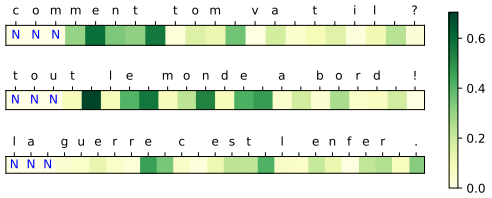
4.2.3 Un corpus artificiel pour la compression de séquence

Nous avons choisi de mener nos essais sur une tâche plus simple et plus contrôlée que la compression de phrase classique. Pour cela, nous avons créé des données artificielles à l'aide d'un système élémentaire de transduction, qui compresse ou décompresse une phrase source de façon extractive, respectivement en supprimant ou en répétant une partie des caractères. Afin de rendre l'apprentissage d'une telle transformation moins triviale, la décision de supprimer (resp. répéter) un caractère source x_i suit une probabilité p_{sup} (resp. p_{rep}) qui dépend de son « entropie » (autrement dit, qui dépend de sa propension à



(a) Taux de compression (b) Longueur source

FIGURE 4.1 – Histogrammes du taux de compression et de la longueur source dans l'ensemble d'entraînement du corpus artificiel.



$r = 0,8$	`commttomva t il ?`
$r = 1,2$	`toutt le mmondde a borrd !`
$r = 0,6$	`la guerrc est l enfer`

(a) phrases initiales (sources) (b) phrases modifiées (cibles)

FIGURE 4.2 – Exemples de paires dans le corpus de compression/décompression. Les phrases sources sont accompagnées d'une carte thermique indiquant pour chaque caractère (excepté les 3 premiers) la probabilité attribuée par un modèle de langue 4-gramme. Les phrases cibles sont compressées ou décompressées, selon r.

pouvoir être prédit étant donné le contexte antérieur) :

$$\begin{aligned} p_{sup}(x_i|x_{[i-n+1;i-1]};\theta) &= \max(1, \alpha \times p_\theta(x_i|x_{[i-n+1;i-1]})), \\ p_{rep}(x_i|x_{[i-n+1;i-1]};\theta) &= \max(1, \beta \times (1 - p_\theta(x_i|x_{[i-n+1;i-1]}))), \end{aligned} \tag{4.9}$$

où θ est un *modèle de langue* n-gramme au niveau caractère (en pratique nous prenons $n = 4$), $p_\theta(x_i|x_{[i-n+1;i-1]})$ est la probabilité d'après p_θ de x_i sachant le contexte des $n-1$ caractères précédents, et α et β sont des facteurs d'échelle (permettant de contrôler la valeur moyenne). Ainsi, une compression devrait supprimer les caractères les plus prévisibles, et une décompression devrait répéter les moins prévisibles [4].

Notre ensemble de données est constitué de 75 569 phrases de 5 à 10 mots en français,

4. Cette procédure ressemble sous cet aspect à la méthode de compression de texte par omission des caractères prévisibles (méthode 3 de Schmidhuber & Heil (1996)), ou plus généralement aux méthodes attribuant moins de bits pour la représentation des caractères les plus courants.

normalisées [5], provenant de Tatoeba [6] (7 694 d'entre elles sont allouées à un ensemble de développement, et 7 457 à un ensemble de test). Les phrases cibles sont un mélange de phrases compressées et décompressées. Nous avons défini les valeurs des coefficients α et β de manière à ce que le taux de compression l_y/l_x soit uniformément distribué dans $[0,5;1,5]$, quand échantillonné sur le corpus entier (Figure 4.1a) :

$$\alpha = \frac{1-r}{\tilde{p}_\theta}, \quad \beta = \frac{r-1}{1-\tilde{p}_\theta}, \tag{4.10}$$

où r est une variable échantillonnée uniformément dans $[0,5;1,5]$ pour chaque phrase (si $r > 1$ une décompression est opérée, sinon une compression), et $\tilde{p}_\theta$ est la probabilité moyenne attribuée par le modèle n-gramme θ aux caractères de référence sur ses données d'apprentissage. θ avait été appris au préalable sur le côté source de l'ensemble d'entraînement.

La figure 4.1b montre la répartition des longueurs sources de l'ensemble d'entraînement, et la figure 4.2 présente des exemples de phrases compressées ou décompressées de l'ensemble de test.

4.3 Expériences

4.3.1 Évaluation des modèles de contrôle de longueur

4.3.1.1 Précision du contrôle de longueur

Nous nous sommes attachés dans notre premier groupe d'expériences à mesurer la précision avec laquelle est contrôlée la longueur de la phrase engendrée. Pour cela nous avons choisi de calculer l'*erreur absolue moyenne* (EAM) et la *racine de l'erreur quadratique moyenne* (REQM) des taux de compression obtenus par rapport aux taux de compression visés, selon les formules suivantes :

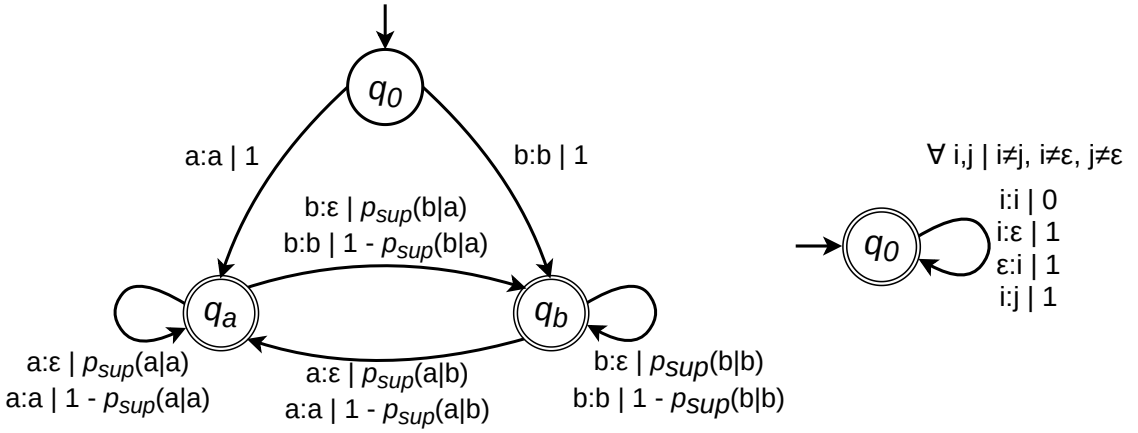
$$\text{EAM} = \frac{1}{n}\sum_{i=1}^{n}|\hat{r}_i - r_i|, \quad \text{REQM} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{r}_i - r_i)^2}, \tag{4.11}$$

où $n = 7457$ est le nombre d'instances dans l'ensemble de test, et $\hat{r}_i$ et r_i sont respectivement le taux de compression obtenu et le taux de compression visé pour la i-ème phrase transduite.

L'*erreur absolue* (EA) $|\hat{r} - r|$ peut aussi être vue comme la différence entre la longueur

5. Nous avons retiré les diacritiques, majuscules, apostrophes, tirets et virgules.

6. Tatoeba est une base de données de traduction multilingue, constituée de contributions volontaires. `https://tatoeba.org`, diffusé sous licence CC-BY 2.0 FR.



(a) Transducteur de compression ($n = 2, \Sigma = \{a, b\}$) (b) Transducteur d'édition

FIGURE 4.3 – Représentations de transducteurs finis pondérés. Dans (a) l'état q_a (resp. q_b) correspond au contexte $x_{[i-n+1;i-1]} = a$ (resp. $x_{[i-n+1;i-1]} = b$). Pour (a) les poids sont définis dans le *demi-anneau des probabilités*, pour (b) dans le *demi-anneau tropical*.

produite et la longueur visée $|l_{\hat{y}} - r \times l_x|$ rapportée à la longueur source l_x. Pour compléter nos métriques, nous avons évalué la proportion d'instances pour lesquelles l'erreur absolue est inférieure à 10%, 5%, ou bien est nulle [7] (Section 4.5.1).

4.3.1.2 Validité des phrases engendrées

Quoique davantage intéressés par l'aptitude des modèles à contrôler la longueur, nous avons mis en place une évaluation pour mesurer la validité (ou grammaticalité en un sens) des phrases engendrées par rapport aux données d'entraînement. La procédure suivie pour créer une phrase dans le côté cible du corpus peut être interprétée comme un *transducteur fini pondéré* : chaque lettre d'entrée est conservée ou supprimée/doublée (selon qu'une compression ou une décompression est réalisée) en suivant des probabilités (Équation (4.9)) conditionnées sur le contexte des $n-1$ caractères précédents, contexte qui peut être associé à un état. La figure 4.3a représente un tel transducteur de compression, pour $n = 2$ et un alphabet réduit à $\Sigma = \{a, b\}$ (ce qui est une simplification par rapport au vrai cadre expérimental, dans lequel $n = 4$ et $|\Sigma| = 30$).

Notons T_r^{comp} les transducteurs pour la compression ($r \in [0,5;1]$), et T_r^{decomp} les transducteurs pour la décompression ($r \in [1;1,5]$). Grâce à eux, nous sommes capables de vérifier si une phrase produite par un modèle encodeur-décodeur est « correcte » par rapport à la procédure suivie pour créer le côté cible de notre corpus artificiel pour la compression de phrase. Une phrase $\hat{y} = \text{SYS}_{r=0,6}(x)$, engendrée par un système (LenInit, LenEmb, LRPE ou LDPE) à partir d'une phrase source x avec un taux de compression visé $r = 0,6$,

7. Les cas où $|l_{\hat{y}} - \lfloor r \times l_x \rfloor| = 0$.

x	`la guerre c est l enfer .`
$\hat{y} = \text{LenInit}_{r=0,6}(x)$	`la gerre c est lenfre`
$\hat{y}'$	`la gerre c est lenfe`

TABLE 4.1 – Exemple de phrase source x, phrase prédite (produite par LenInit) $\hat{y}$, et plus proche phrase parmi celles que le transducteur de compression pourrait engendrer à partir de la source $\hat{y}'$. $\hat{y}$ est invalide à cause de l'interversion (opération non-autorisée) des deux dernières lettres.

est correcte si elle appartient à $\text{Im}(T^{comp}_{r=0,6} \circ x)$, l'ensemble des phrases cibles qui peuvent être obtenue depuis x par l'automate de compression correspondant (comme à chaque étape du décodage le système peut a priori engendrer n'importe quel caractère de l'alphabet, il est tout à fait possible que $\hat{y}$ ne soit pas dans le langage rationnel image du transducteur).

Nous pouvons aussi calculer la probabilité associée à une paire (x,y) formée d'une phrase source et d'une phrase cible correcte : $T^{comp}_r(x,y)$ (resp. $T^{decomp}_r(x,y)$), qui correspond à la probabilité cumulée de tous les chemins dans T^{comp}_r (resp. T^{decomp}_r) qui transduisent x en y. Dans notre évaluation, nous avons utilisé la probabilité logarithmique négative divisée par la longueur de la phrase source, définissant les scores :

$$S^{comp}_r(x,y) = \frac{-\log T^{comp}_r(x,y)}{l_x}, \quad S^{decomp}_r(x,y) = \frac{-\log T^{decomp}_r(x,y)}{l_x}. \tag{4.12}$$

Pour les cas dans lesquels une phrase prédite $\hat{y} = \text{SYS}_r(x)$ n'est pas correcte (au sens donné ci-dessus), nous recherchons dans $\text{Im}(T^{comp}_r \circ x)$ (resp. $\text{Im}(T^{decomp}_r \circ x)$) la phrase $\hat{y}'$ la plus proche en terme de distance d'édition, et calculons $S^{comp}_r(x,\hat{y}')$ (resp. $S^{decomp}_r(x,\hat{y}')$). Plus précisément, le calcul de la distance d'édition repose sur l'utilisation d'un transducteur T^{edit} (Figure 4.3b) implémentant les quatre opérations classiques (conservation, insertion, suppression, substitution) : $T^{edit} \circ T^{comp}_r \circ x$ représente alors les modifications de compressions de x, et $(T^{edit} \circ T^{comp}_r \circ x)^{-1} \circ \hat{y}$ les chemins transduisant x en $\hat{y}$ par une compression suivie d'une dégradation. $\hat{y}'$ est obtenue en appliquant un algorithme de plus courte distance [8] (Mohri, 2002) sur $(T^{edit} \circ T^{comp}_r \circ x)^{-1} \circ \hat{y}$. Le tableau 4.1 donne un exemple d'un tel triplet $(x,\hat{y},\hat{y}')$.

4.3.2 Prédiction de longueur à partir des états cachés

Notre deuxième groupe d'expériences s'attache à comprendre comment la longueur est représentée dans les états cachés du décodeur, et comment la contrainte impliquée par

8. Lors de l'application de cet algorithme, un poids neutre est assigné aux arcs du transducteur T^{comp}_r, de sorte que seul le poids des opérations d'édition est pris en compte.

LenInit agit pendant le décodage. Shi et al. (2016); Adi et al. (2016) ont déjà montré qu'une information de longueur se trouve dans ce genre de représentations dans les cas de l'auto-encodage et de la traduction. Néanmoins nos essais se placent dans un cadre de compression de phrase avec un taux visé variable ; dans ce cas, la longueur cible ne peut être déterminée par une relation constante à partir de la longueur source (comme dans le cas de la traduction), et le décodeur doit intégrer des données extérieures.

En utilisant le modèle LenInit (Section 4.2.1) entraîné sur notre corpus de compression (Section 4.2.3), nous avons conçu une tâche de classification qui prédit – étant donné un état caché échantillonné à une certaine étape j du décodage – la longueur de la séquence restant à produire. Les classes de sortie correspondent à des valeurs de longueur, entre 0 et 149 (ce qui est supérieur à la plus grande longueur enregistrée dans le corpus).

Nous avons créé un ensemble de données pour cette tâche, en échantillonnant aléatoirement une fraction (1%) de tous les états cachés produits pendant le décodage (effectué pour divers objectifs de taux de compression) de phrases issues de l'ensemble d'entraînement du corpus de compression [9]. À chacun de ces états cachés a été associé le nombre de caractères de sortie engendrés après qu'il a été échantillonné (c'est-à-dire entre son étape j et la fin du décodage de sa phrase). Ainsi, nous avons obtenu des classes représentées selon leurs proportions naturelles.

4.3.3 Évolution de la probabilité de génération de la fin de phrase

Notre troisième groupe d'expériences suit l'évolution des probabilités respectivement associées au symbole *fin-de-phrase* et à certaines marques de ponctuation (à savoir « . », « ! » et « ? ») au cours du décodage. Des essais semblables ont été menés par Shi et al. (2016), sans trouver de progression régulière. Nous souhaitons vérifier si le cadre de la compression de phrase amène un changement à ce niveau.

Nous utilisons la même configuration que précédemment pour le modèle encodeur-décodeur (LenInit), que nous exécutons sur la partie test du corpus de compression (Section 4.2.3). Les probabilités sont extraites des distributions sur le vocabulaire [10] créées à chaque étape du décodage.

4.4 Implémentation

Modèle de langue n-gramme au niveau caractère

Pour pouvoir produire le corpus expérimental de compression (Section 4.2.3), nous avons implémenté un modèle de langue 4-gramme au niveau caractère sous la forme d'un

9. LenInit a été utilisé pour transduire les mêmes phrases sur lesquelles il avait été entraîné.
10. Pour notre modèle, le vocabulaire est l'ensemble des caractères utilisés

perceptron multicouche (Bengio et al., 2003) avec une seule couche cachée (de dimension 128). Il a été entraîné pendant 4 époques sur 60 418 phrases en français de Tatoeba, par descente de gradient stochastique, en utilisant *Adam* (Kingma & Ba, 2015) avec son paramétrage standard : $\beta_1 = 0,9$, $\beta_2 = 0,999$, $eps = 10^{-8}$.

Modèles RNN

Nous avons implémenté les modèles à base de RNN (Section 4.2.1) comme des bi-*GRU* (Cho et al., 2014b) (dimension de plongement = 20, dimension cachée = 300). Pour LenInit et ses variantes, la dimension du paramètre de longueur L_{param} est de 300 (dans le cas de LenInit2, elle est divisée à égalité entre $l \times V$ et $L_{plong}(l)$). Pour LenEmb, la dimension du plongement de la longueur restante $L_{plong}(l_j)$ est de 50 (nous prenons $l^{max} =$ 150). Ces systèmes ont été entraînés pendant une époque sur 60 418 phrases en français de Tatoeba, en utilisant Adam (paramétrage standard).

Modèles Transformer

Les modèles *Transformer* sur lesquels reposent LRPE et LDPE ont été implémentés avec les caractéristiques suivantes :

— dimension des représentations internes et des plongements lexicaux $d_{\text{m}} = 200$;
— dimension du perceptron multicouche $d_{\text{ff}} = 200$;
— nombre de têtes d'attention $h = 2$;
— nombre de couches pour l'encodeur et le décodeur $N = 2$.

Le nombre global de paramètres résultant est d'environ 1,3 millions, ce qui est comparable à la taille de nos systèmes RNN (approximativement 1,5 millions de paramètres).

Évaluation des phrases engendrées

Les transducteurs finis pondérés utilisés pour évaluer la validité des phrases engendrées ont été implémentés à l'aide de la librairie *Pynini* (Gorman, 2016), elle-même fondée sur *OpenFST* (Allauzen et al., 2007).

Classification de la longueur future

Le classificateur pour la longueur future est un perceptron multicouche avec une seule couche cachée (de dimension 300, égale à la dimension d'entrée) activée par la fonction *ReLU*. Il a été entraîné pendant 5 époques sur 17 112 vecteurs échantillonnés, en utilisant Adam (paramétrage standard).

Source	comment tom va t il ?
LenInit$_{r=0,6}$	commnt m a il ?
LenInit$_{r=1,0}$	comment tom va ti l ?
LenInit$_{r=1,4}$	commment tomm va ti l ??
Source	tout le monde a bord !
LenEmb$_{r=0,6}$	tou lmodabor!
LenEmb$_{r=1,0}$	tout le monde a bord !
LenEmb$_{r=1,4}$	tout lle oondde a boorrd !
Source	j ai a nouveau gagne .
LRPE$_{r=0,6}$	j ai nou gag
LRPE$_{r=1,0}$	j ai a nouveau gagne .
LRPE$_{r=1,4}$	j ai a nouveau gaaaggggggne .
Source	il s est mis en colere .
LDPE$_{r=0,6}$	il stmisecole
LDPE$_{r=1,0}$	il s est mis en coler .
LDPE$_{r=1,4}$	il s est mis en colereeeeeere ..

TABLE 4.2 – Exemples de phrases transduites par LenInit, LenEmb, LRPE et LDPE (taux 0,6, 1,0 et 1,4).

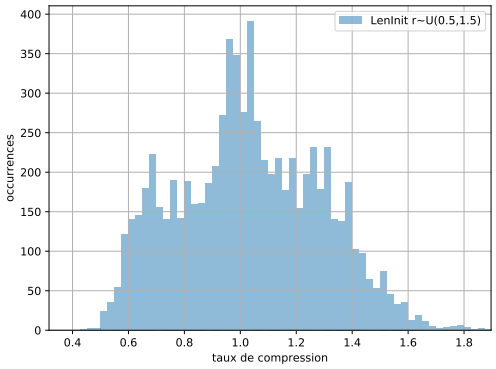
Les vecteurs des états cachés de l'ensemble de données pour la classification ont été produits en décodant des phrases de l'ensemble d'entraînement du corpus de compression : pour ces transductions nous avons utilisé LenInit, et pour chacune d'elles nous avons pris un taux de compression objectif uniformément échantillonné dans $[0,5;1,5]$. Cette répartition aléatoire a été choisie afin de limiter l'introduction de biais dans la tâche.

4.5 Résultats

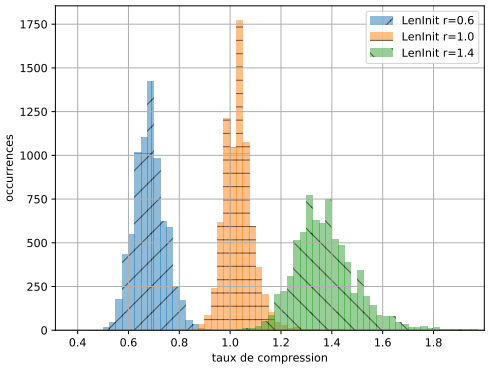
4.5.1 Compression/décompression de phrases

Avant de réaliser nos expériences autour des mécanismes de détermination de la longueur, nous avons testé LenInit et les autres modèles sur le corpus artificiel de compression de phrase. Le tableau 4.2 met en avant quelques exemples de phrases issues de l'ensemble de test, transduites avec différents taux de compression visés. Afin de vérifier la capacité des systèmes à contrôler la longueur, nous avons transduit l'ensemble de test selon plusieurs modalités :

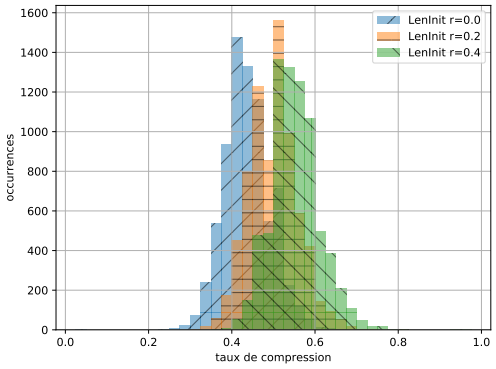
1. en échantillonnant le taux visé r uniformément dans $[0,5;1,5]$;
2. en fixant r à des valeurs présentes dans les données d'apprentissage (0,6, 1,0 et 1,4) ;
3. en fixant r à des valeurs hors domaine (en dessous – 0,0, 0,2, 0,4, et au dessus – 1,6, 1,8, 2,0)).



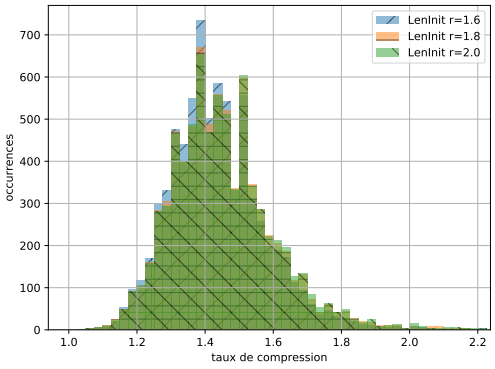
(a) LenInit $r \sim \mathscr{U}(0,5;1,5)$



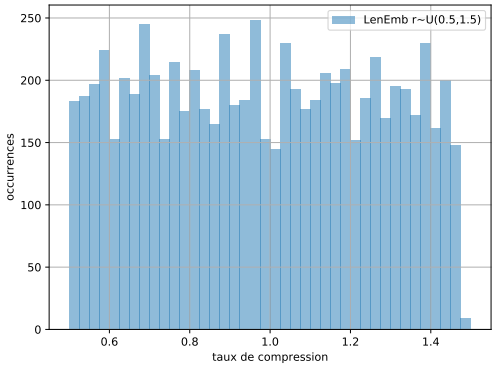
(b) LenInit $r = 0,6;1,0;1,4$



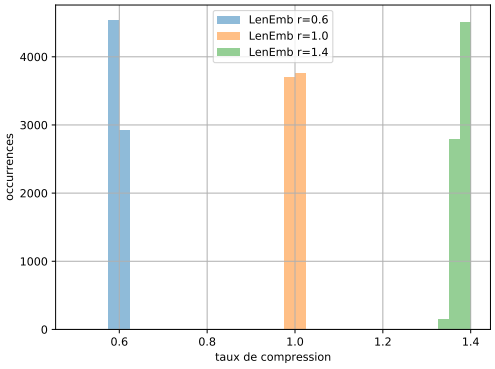
(c) LenInit $r = 0,0;0,2;0,4$



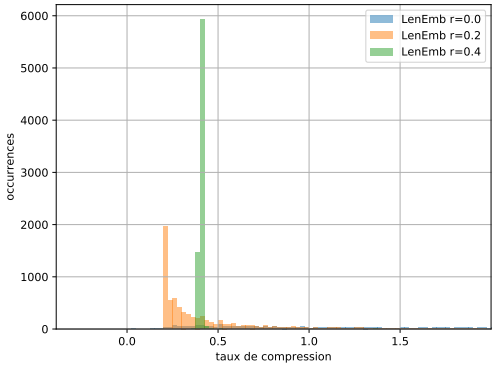
(d) LenInit $r = 1,6;1,8;2,0$



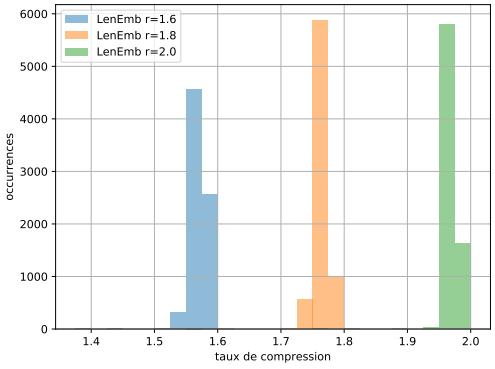
(e) LenEmb $r \sim \mathscr{U}(0,5;1,5)$



(f) LenEmb $r = 0,6;1,0;1,4$

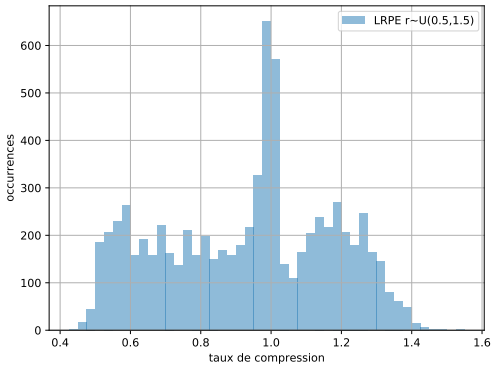


(g) LenEmb $r = 0,0;0,2;0,4$

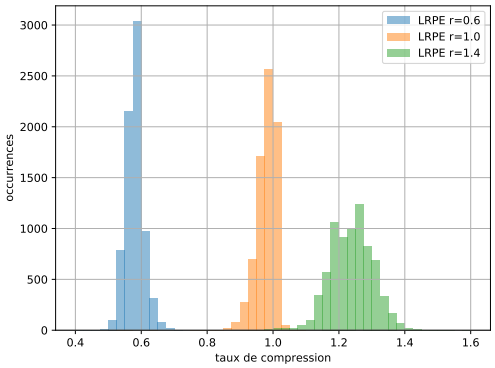


(h) LenEmb $r = 1,6;1,8;2,0$

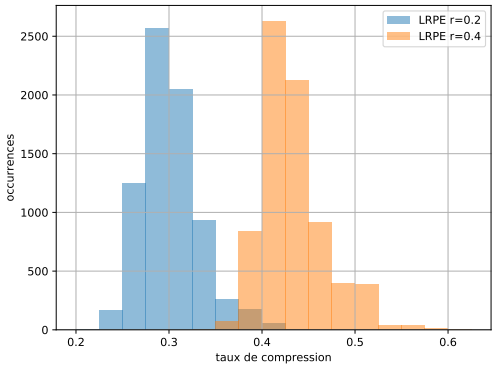
FIGURE 4.4 – Histogrammes des taux de compression dans l'ensemble de test, transduit par LenInit et LenEmb selon différentes modalités.



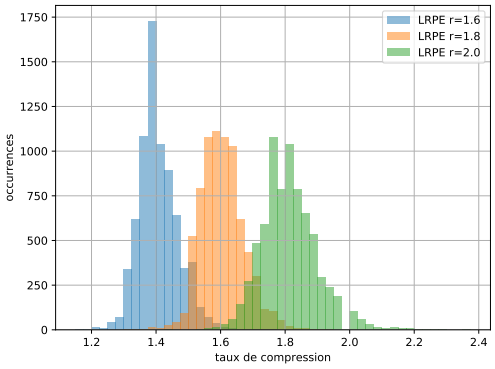
(a) LRPE $r \sim \mathscr{U}(0,5;1,5)$



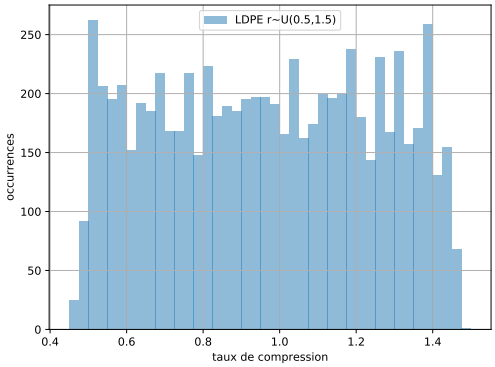
(b) LRPE $r = 0,6;1,0;1,4$



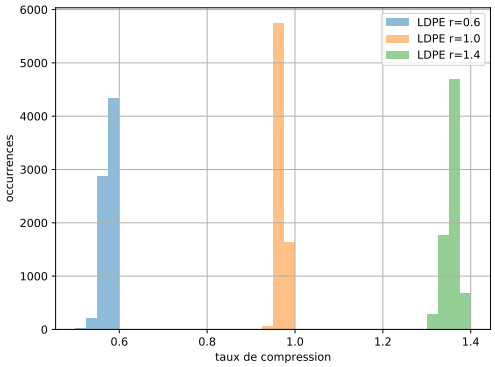
(c) LRPE $r = 0,2;0,4$



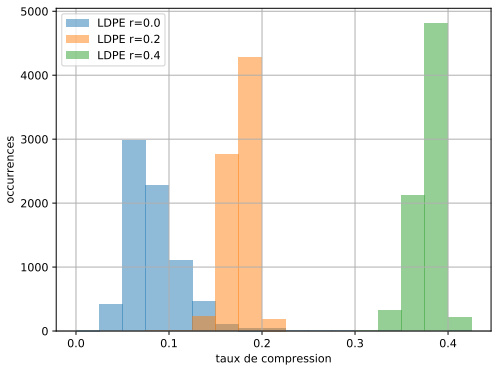
(d) LRPE $r = 1,6;1,8;2,0$



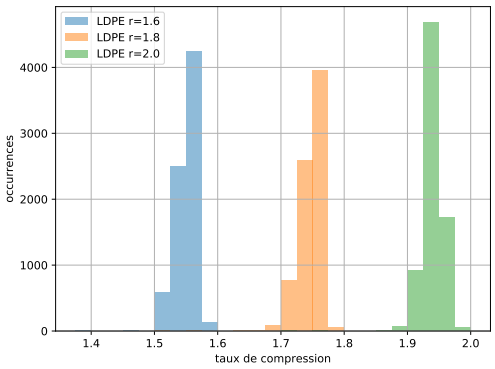
(e) LDPE $r \sim \mathscr{U}(0,5;1,5)$



(f) LDPE $r = 0,6;1,0;1,4$



(g) LDPE $r = 0,0;0,2;0,4$



(h) LDPE $r = 1,6;1,8;2,0$

FIGURE 4.5 – Histogrammes des taux de compression dans l'ensemble de test, transduit par LRPE et LDPE selon différentes modalités. La formule de LRPE n'est pas compatible avec $r = 0,0$.

Les principaux résultats sont présentés par les figures 4.4 et 4.5, qui illustrent les distributions de taux de compression obtenus, et par le tableau 4.3, qui donne des mesures portant d'une part sur la précision du contrôle de longueur, et d'autre part sur la validité des phrases engendrées.

4.5.1.1 Analyse selon le taux de compression

Les figures 4.4a, 4.4e, 4.5a et 4.5e montrent qu'en suivant le même échantillonnage de r dans $[0,5;1,5]$, la distribution des taux de compression obtenus est relativement conforme à celle présente dans le corpus d'entraînement (Figure 4.1) ; nous remarquons néanmoins une densité plus importante autour de $r = 1$ dans le cas de LenInit et de LRPE (de façon plus accentuée pour le second), qui semble témoigner d'une tendance à reproduire la phrase source. Nous observons dans les figures 4.4b, 4.4f, 4.5b et 4.5f qu'aux objectifs r fixes sont associées des distributions de taux de compression effectivement centrées autour d'une valeur. Avec les modèles LenInit et LRPE, la variance est toutefois plus importante, en particulier dans le cas $r = 1,4$, pour lequel la gaussienne obtenue est également davantage décalée par rapport au taux visé, probablement à cause de la difficulté pour l'encodeur-décodeur à manipuler les longues séquences. Enfin les figures 4.4c, 4.4d, 4.4g, 4.4h, 4.5c, 4.5d, 4.5g et 4.5h permettent d'appréhender la capacité des systèmes à généraliser leur action pour des taux de compressions non-vus dans les données d'entraînement. LenInit apparaît comme une exception pour cet aspect, ne produisant que peu de taux en dehors de l'intervalle $[0,5;1,5]$. Cela n'est pas intuitif, dans la mesure où LenInit encode la valeur absolue de la longueur visée, et non le taux de compression (Équation (4.3)). Une valeur de r en dehors de $[0,5;1,5]$ peut, selon la longueur de la phrase source, correspondre à une longueur cible (objectif) rencontrée lors de l'apprentissage. Cela laisse penser que le modèle de langue contenu dans le décodeur pourrait s'opposer et prévaloir face aux mécanismes de contrôle de longueur.

Ces observations sont pour l'essentiel corroborées par les erreurs EAM et REQM dans le tableau 4.3. Nous pouvons aussi noter que la plupart des modèles engendrent davantage de phrases valides pour $r = 0,6$ que pour $r = 1,4$. En ce qui concerne LRPE et LDPE, une observation des données laisse penser que dans le cas de la décompression, la gestion de l'objectif de longueur intervient souvent vers la fin du décodage, la première partie se caractérisant par une stricte copie de la source (voir les exemples du Tableau 4.2).

4.5.1.2 Architecture RNN p. opp. architecture Transformer

Dans cette section nous comparons les modèles selon que leur architecture est fondée sur les RNN ou sur *Transformer*. Du point de vue de la précision du contrôle de longueur,

Modèle	EAM	REQM	EA=0	EA<5 %	EA<10 %	corr.	S_{corr}	S_{inco}	S_{glob}
				$r \sim \mathscr{U}(0,5;1,5)$					
LenInit	8,4 %	0,108	9,6 %	35 %	67 %	16,4 %	0,35	0,44	0,42
LenInit2	6,1 %	0,086	18,5 %	53 %	82 %	22,2 %	0,39	0,47	0,45
LenInit3	7,7 %	0,119	14,7 %	45 %	74 %	19,8 %	0,36	0,45	0,43
LenEmb	1,7 %	0,020	75,0 %	99 %	100 %	25,1 %	0,40	0,50	0,47
LRPE	7,7 %	0,105	21,4 %	50 %	68 %	58,3 %	0,37	0,42	0,39
LDPE	3,4 %	0,037	40,3 %	85 %	100 %	46,2 %	0,44	0,42	0,43
Oracle (T)	6,1 %	0,080	17,0 %	51 %	80 %	100 %	0,37	-	0,37
				r fixé					
$RNN_{r=0,6}$	10,9 %	0,195	7,30 %	29 %	58 %	16,4 %	0,38	0,47	0,45
$LenInit_{r=0,6}$	8,6 %	0,102	6,0 %	29 %	64 %	15,5 %	0,52	0,56	0,56
$LenInit_{r=1,0}$	4,7 %	0,071	25,1 %	62 %	89 %	8,6 %	0	-	0
$LenInit_{r=1,4}$	9,9 %	0,131	10,3 %	31 %	59 %	6,1 %	0,54	0,55	0,55
$LenEmb_{r=0,6}$	0,7 %	0,009	60,8 %	100 %	100 %	12,0 %	0,57	0,62	0,61
$LenEmb_{r=1,0}$	0,0 %	0,001	100 %	100 %	100 %	27,8 %	0	-	0
$LenEmb_{r=1,4}$	2,5 %	0,027	60,8 %	98 %	100 %	19,3 %	0,57	0,57	0,57
$LRPE_{r=0,6}$	2,7 %	0,033	42,3 %	87 %	100 %	74,4 %	0,58	0,59	0,58
$LRPE_{r=1,0}$	2,0 %	0,035	55,3 %	85 %	99 %	53,8 %	0	-	0
$LRPE_{r=1,4}$	16,2 %	0,173	0,7 %	4 %	17 %	2,3 %	0,51	0,51	0,51
$LDPE_{r=0,6}$	2,4 %	0,027	63,0 %	97 %	100 %	75,2 %	0,57	0,59	0,57
$LDPE_{r=1,0}$	3,1 %	0,032	0,0 %	99 %	100 %	0,0 %	0	-	0
$LDPE_{r=1,4}$	4,4 %	0,047	6,0 %	72 %	100 %	0,0 %	0,47	0,53	0,53

TABLE 4.3 – Mesures sur la précision du contrôle de longueur et sur la validité des phrases produites (Section 4.3.1). corr., S_{corr}, S_{inco}, S_{glob} indiquent respectivement la proportion de phrases correctes, et le score moyen attribué aux phrases correctes, incorrectes, et à l'ensemble des phrases. L'oracle est le transducteur fini pondéré qui a engendré le corpus.

il est difficile de formuler une assertion globale : LRPE obtient de meilleurs résultats que LenInit, mais LDPE est moins précis que LenEmb (qui de façon générale est le premier système pour les métriques de précision). LRPE et LDPE engendrent plus de phrases correctes en moyenne, mais cela concerne en réalité surtout les compressions (le pourcentage de sorties valides est bien inférieur pour $r = 1,4$ que pour $r = 0,6$; LenInit et LenEmb sont plus équilibrés à cet égard). Pour S_{corr} et S_{glob}, LRPE et LenInit sont approximativement comparables (S_{corr} est légèrement plus élevé pour LenInit, mais S_{glob} est un peu plus haut pour LRPE). De même, il est difficile de départager LDPE et LenEmb sur les scores de validité : LenEmb reçoit des valeurs plus grandes pour les phrases correctes, mais celles-ci représentent une proportion plus petite de ses productions ; pour le score moyen sur la totalité des phrases (S_{glob}), LDPE est au dessus.

4.5.1.3 Encodage de la longueur totale p. opp. encodage de la longueur restante

Les différences les plus nettes sont obtenues selon la dichotomie entre les méthodes encodant la longueur totale (LenInit et LRPE) et celles encodant la longueur restante

(LenEmb et LDPE). LenEmb est plus précis sur la réalisation de la longueur visée que LenInit, et LDPE l'est davantage que LRPE de façon similaire (ce qui confirme les résultats présentés respectivement par Kikuchi et al. (2016) et Takase & Okazaki (2019)). LRPE a également une plus grande proportion de phrases correctes et a de meilleurs scores que LDPE (sur la tâche de génération de titre d'article, Takase & Okazaki (2019) trouvaient effectivement de meilleurs scores ROUGE pour LRPE ; de même dans les expériences de traduction de Lakew et al. (2019), LDPE enregistrait une sous-performance en BLEU par rapport aux autres systèmes testés). Sous cet angle le contraste entre les méthodes RNN est plus nuancé : le pourcentage de phrases valide est plus important pour LenEmb, alors que LenInit engendre des phrases plus probables d'après S_{corr} et S_{glob}. LenInit2 et LenInit3 sont quasiment à tous égards (précision, correction, scores) intermédiaires entre LenInit et LenEmb. Il apparaît en particulier qu'un encodage par plongement permet plus de précision dans le contrôle de longueur qu'un encodage par la norme (qui en outre, comme précisé plus haut, ne garantit pas la fonctionnalité pour des valeurs hors domaine). Toutefois, l'encodage mixte de LenInit2 se montre plus efficace (du point de vue de la précision et de la proportion de phrases correctes) que le plongement pur de LenInit3, ce qui suggère que l'encodage continu apporte tout de même un bénéfice pour les valeurs peu fréquentes.

L'impression générale qui résulte de ces observations est celle d'un compromis : l'encodage de la longueur totale (soit à l'initialisation du décodage pour LenInit, soit par l'encodage de position pour LRPE) permet d'obtenir une meilleure qualité pour les phrases engendrées, au prix d'un contrôle de la longueur moins précis ; et inversement, l'encodage de la longueur restante inséré à chaque position de sortie autorise d'approcher plus précisément la longueur visée, au détriment de la qualité des phrases.

À titre de comparaison nous avons aussi entraîné un encodeur-décodeur RNN classique sur un corpus comparable mais ne contenant que des phrases compressées pour $r = 0,6$. Les scores de validité indiquent que le RNN a appris un modèle de langue adapté, et nous notons une précision un peu moins importante que celle de $\text{LenInit}_{r=0,6}$.

4.5.2 Prédiction de la longueur future

Nous avons testé notre classificateur sur l'ensemble de test du corpus de prédiction de la longueur future (Section 4.3.2), qui contient $n = 2169$ états cachés échantillonnés. La figure 4.6a montre la distribution de la différence entre la longueur prédite par le modèle et la longueur de référence (récupérée expérimentalement lors de la création de l'ensemble de données). Cette distribution est approximativement centrée autour de 0 et nous avons

calculé les valeurs EAM et REQM comme suit :

$$\text{EAM} = \frac{1}{n}\sum_{i=1}^{n}|\hat{l}_i - l_i| = 3,08, \quad \text{REQM} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{l}_i - l_i)^2} = 4,94, \tag{4.13}$$

où $\hat{l}_i$ et l_i sont respectivement la longueur prédite et la longueur de référence pour le i-ème échantillon.

Notons que ce résultat confirme l'existence, dans un état caché du décodeur, de données qui représentent spécifiquement la longueur à venir, et qui ne peuvent dériver directement d'un reliquat d'information relatif à la phrase d'entrée. En effet, puisque le modèle LenInit qui a engendré les états cachés avait reçu des objectifs choisis aléatoirement (uniformément dans $[0,5;1,5]$), le classificateur ne devrait pas avoir pu établir de corrélation systématique entre la longueur d'entrée et la longueur de sortie.

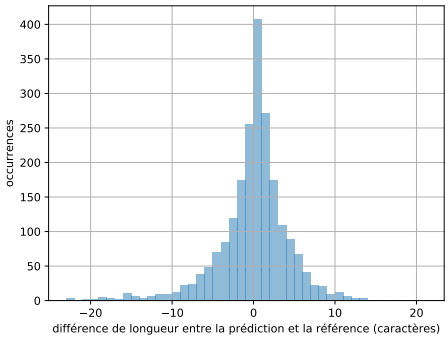
La figure 4.6b montre l'évolution de la mesure REQM en fonction de la position de l'état caché dans la séquence de sortie. L'indicateur d'erreur est calculé selon :

$$\text{REQM}(k) = \sqrt{\frac{1}{|I_k|}\sum_{i\in I_k}(\hat{l}_i - l_i)^2}, \tag{4.14}$$

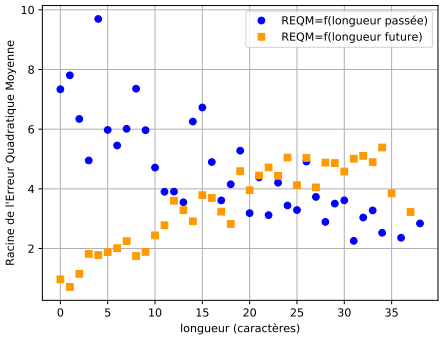
où I_k est l'ensemble d'indices qui décrit les instances de l'ensemble de test qui ont été échantillonnées à la position k dans leur phrase de sortie ; les deux séries sur la figure se distinguent en mesurant k soit depuis le début de la phrase, soit avant sa fin. La tendance suggère que la précision de la prédiction augmente au fil du décodage.

Similairement, la figure 4.6c montre l'évolution au cours du décodage d'une mesure de type REQM, cette fois calculée par rapport à la longueur visée : dans l'équation (4.14), l_i ne correspond plus à la longueur future de référence (celle obtenue en pratique), mais à la longueur future visée (celle qu'il faudrait produire pour atteindre l'objectif de compression donné par r). Les profils sont moins nets dans ce cas, mais nous constatons que l'erreur en fin de phrase est plus importante (ce qui découle logiquement de l'imprécision du contrôle exercé par LenInit), et que l'erreur en début de phrase est étonnamment haute, étant donné que la longueur visée est intégrée dans le premier état caché.

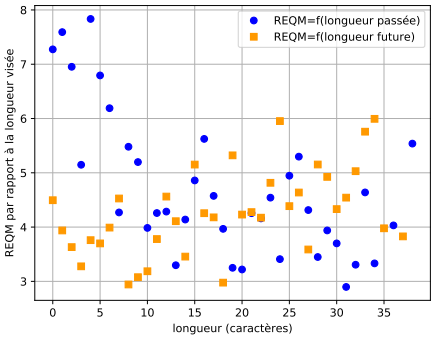
Ces observations peuvent être interprétées comme des signes que l'objectif de longueur plongé dans l'état initial coexiste durant le décodage avec une autre représentation de la longueur, propre au décodeur, qui est influencée par les caractères progressivement engendrés. Il faut considérer le fait que pour LenInit, aucune mesure n'est prise pour réaliser une *dissociation* (*disentanglement*) de l'information dans l'état initial ou les suivants, contrairement à certaines approches pour la génération contrôlée ou le transfert de style, souvent fondées sur l'apprentissage adverse, qui obligent le système à dépendre des valeurs d'at-



(a) Différence par rapport à la longueur de référence



(b) REQM par rapport à la longueur de référence



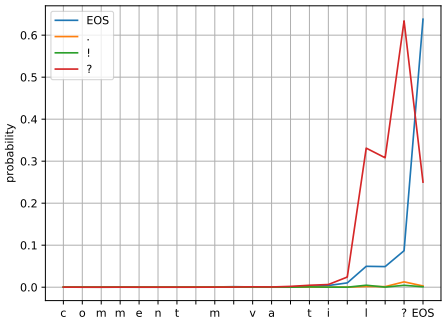
(c) REQM par rapport à la longueur visée

FIGURE 4.6 – (a) Histogramme de la différence $(\hat{l}_i - l_i)$, sur l'ensemble de test. (b),(c) Évolution au cours du décodage de l'erreur de la longueur prédite par rapport à la longueur de référence et par rapport à la longueur visée. Les marqueurs carrés correspondent à l'erreur en fonction du nombre d'étapes avant la génération de *fin-de-phrase*. Les marqueurs ronds correspondent à l'erreur en fonction du nombre d'étapes après *début-de-phrase*.

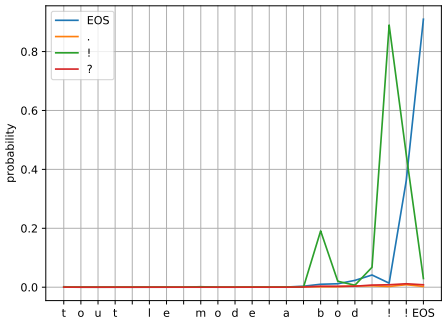
tributs fixés par l'utilisateur, en les faisant disparaître du reste de la représentation interne (Hu et al., 2017; John et al., 2019). L'applicabilité et l'utilité de ce genre d'approches pour le contrôle de longueur peuvent néanmoins être questionnées : une méthode telle que LenEmb est assez précise sans recourir à la dissociation, mais se montre relativement faible pour la qualité des phrases produites ; en outre il paraîtrait difficile de réellement supprimer l'information de longueur dans un encodage de séquence, dans la mesure où même une forme basique comme le *sac de mots* est capable de la capturer (Adi et al., 2016).

4.5.3 Évolution de la probabilité des caractères de fin de phrase

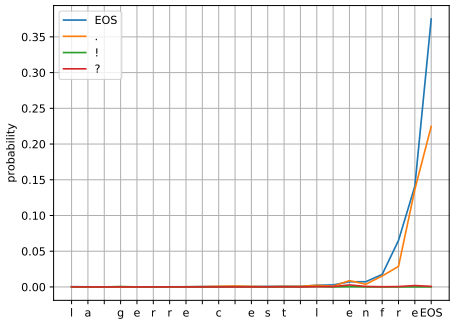
Après avoir analysé l'information de longueur présente dans les états cachés, nous nous sommes penchés sur ce que le modèle de langue du décodeur en faisait. La figure 4.7



(a) Source :
`comment tom va t il ?`



(b) Source :
`tout le monde a bord !`



(c) Source :
`la guerre c est l enfer .`

FIGURE 4.7 – Évolution au cours du décodage de la probabilité attribuée par LenInit$_{r=0,8}$ aux caractères « . », « ! », « ? » et *fin-de-phrase*, pour trois phrases exemples.

donne l'évolution de la probabilité attribuée par le modèle LenInit (appliqué avec un taux de compression visé $r = 0,8$) aux caractères « . », « ! », « ? » et *fin-de-phrase*, pendant le décodage de phrases provenant de Tatoeba (présentes dans l'ensemble de test du corpus de compression). Après examen d'un large éventail de tels graphes, il apparaît que la hausse qui provoque la génération de *fin-de-phrase* est généralement très abrupte, et qu'elle succède la plupart du temps à la génération d'une marque de ponctuation, qui elle même n'a pas eu lieu après une montée régulière ou étagée de probabilité. Ainsi, contrairement aux mesures de la Section 4.5.2, la probabilité des caractères de fin de phrase semble être un signal final à travers lequel il est difficile de suivre la progression du processus de contrôle de la longueur.

4.6 Conclusion

Nous avons étudié et comparé l'efficacité de plusieurs méthodes de contrôle de longueur pour une tâche de compression/décompression, effectuée sur un corpus de données

artificiel créé pour l'occasion. Les résultats mettent en lumière un compromis entre la précision avec laquelle sont atteints les objectifs de longueur, et la qualité des phrases de sortie, que nous avons mesurée selon la probabilité attribuée par le transducteur utilisé pour l'engendrement des exemples d'apprentissage. Parmi les modèles que nous avons testés, ceux qui encodent la longueur totale présentaient un avantage du point de vue de la qualité, tandis que ceux qui encodent la longueur restante se montraient meilleurs du point de vue de la précision. Il apparaît également qu'un encodage de la longueur par plongement est bénéfique pour l'exactitude du contrôle, par rapport à un encodage continu par la norme (les deux peuvent toutefois être combinés). Dans le cas spécifique de la méthode LenInit, qui est une de celles respectant le mieux la « grammaticalité » définie dans nos données, il semble que deux influences s'opposent au cours du décodage : d'une part l'objectif explicite de longueur, et d'autre part la contrainte de produire une phrase juste, en accord avec la source. Nous avons pu déceler la distinction entre ces influences dans l'information contenue dans les états cachés du décodeur. Nous avons aussi relevé une certaine faiblesse pour la tâche de décompression de la part des méthodes de contrôle de longueur reposant sur l'encodage positionnel de l'architecture *Transformer* ; cela reste toutefois à relativiser compte tenu des tailles très modestes retenues pour les modèles dans notre cadre expérimental. Enfin les probabilités attribuées par le décodeur s'avèrent ne pas être un signal permettant d'apprécier le cheminement menant à la décision de finir la phrase.

Au chapitre 6 nous utilisons à nouveau LRPE et LDPE, dans le cadre de l'adaptation de modèles de sous-titrage au genre télévisuel.

Chapitre 5

Corpus pour le sous-titrage d'émissions télévisées

5.1 Introduction

Nous avons choisi d'approcher la production de sous-titres en suivant la métaphore de la traduction, et en nous appuyant, à l'instar de nombreux travaux récents en simplification et en sous-titrage automatique (Zhang et al., 2017; Zhang & Lapata, 2017; Matusov et al., 2019; Karakanta et al., 2020a), sur des architectures neuronales encodeur-décodeur. Toutefois, ces méthodes demandent de grandes quantités de données parallèles représentant la transformation attendue pour pouvoir être mises en œuvre avec succès. Pour les applications de sous-titrage, les ressources de ce type sont encore relativement lacunaires (Karakanta et al., 2020b), particulièrement en français.

Nous décrivons dans ce chapitre un nouveau corpus associant des transcriptions automatiques (les échantillons en langue source, si l'on poursuit la métaphore de la traduction) et des sous-titres en français (les échantillons en langue cible), obtenu à partir du traitement automatique de programmes télévisés contemporains, et des fichiers de sous-titres professionnels fournis par leur diffuseur. Ce corpus est utilisé (Chapitre 6) pour mettre en place une architecture *en cascade* capable de produire sans intervention humaine le fichier de sous-titres correspondant à une entrée vidéo. Nous précisons cependant que ces données ne peuvent pas être partagées du fait des droits associés aux émissions [1].

Le processus de création de ce corpus est détaillé dans la section 5.2 ; les sections suivantes analysent les caractéristiques et la variabilité observées au sein du corpus, en contrastant notamment les sous-titres produits *en direct* et ceux produits *en différé* (Section 5.3), et en faisant la distinction entre les *genres télévisuels* auxquels les programmes appartiennent (Section 5.4).

Nos contributions pour ce chapitre sont les suivantes :

1. Les données appartiennent au diffuseur pour la partie sous-titre, la propriété des enregistrements étant répartie sur les multiples acteurs de la chaine de production.

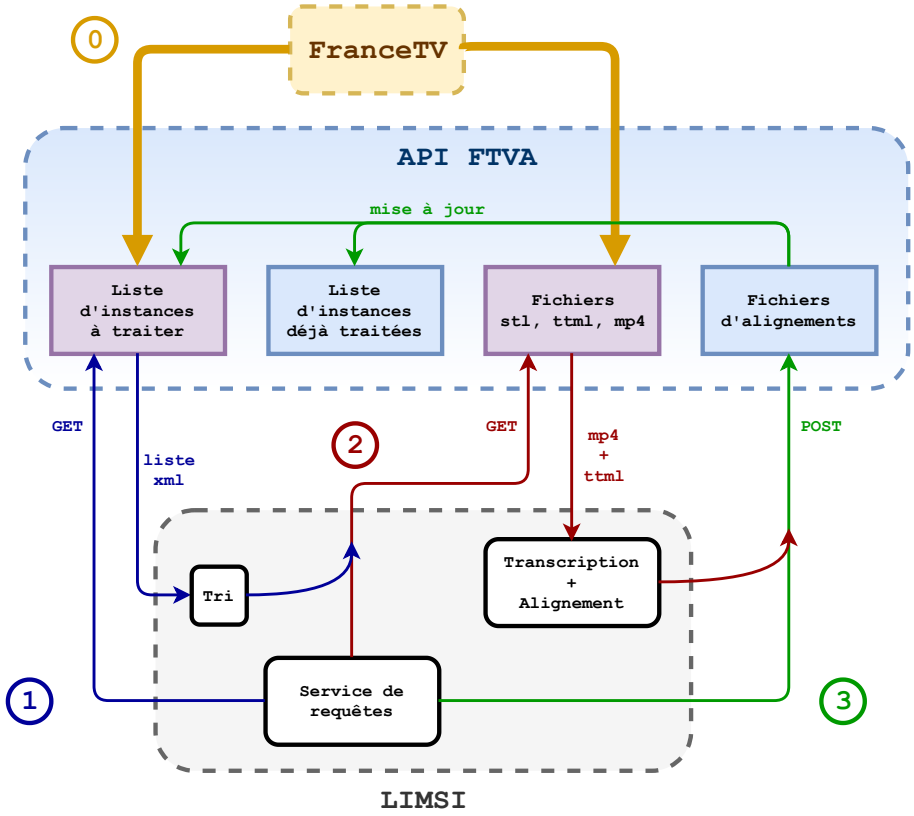


FIGURE 5.1 – Chaîne de traitement globale pour le recueil et la préparation des données du corpus de sous-titrage. Les chiffres indiquent les étapes du processus. En amont (0), les vidéos (au format `mp4`) des programmes télévisés ainsi que les sous-titres professionnels (aux formats `stl` et `ttml`) sont mis en ligne par france.tv access ; un registre tient compte des instances d'émissions disponibles. Régulièrement (1), une requête est effectuée auprès de l'API afin de récupérer la dernière version de ce registre. Dans le cas où des émissions non-encore traitées apparaissent, d'autres requêtes sont émises (2) pour collecter les données vidéo (`mp4`) et sous-titres (`ttml`) correpondantes. Après transcription et alignement (voir les Sections 5.2.1.2 et 5.2.1.3), les exemples parallèles obtenus sont en retour postés (3) sur l'API, dans des fichiers suivant un format convenu.

— Création d'un grand corpus pour l'apprentissage et l'évaluation de modèles de production de sous-titres.
— Analyse des caractéristiques de ce corpus, en étudiant notamment l'influence des stratégies de sous-titrage et des genres télévisuels.

5.2 Recueil et annotation de corpus

La création du corpus parallèle s'est déroulée d'octobre 2019 à novembre 2021 ; au cours de cette période, france.tv access a mis à notre disposition via une API un flux de données relatives aux émissions alors diffusées. Nous avons mis en place une chaîne de traitement (représentée par la Figure 5.1) afin d'intégrer les émissions au fur et à mesure

de leur mise en ligne .

5.2.1 Corpus pour l'apprentissage

5.2.1.1 Sélection des émissions

Le panel d'émissions destiné à la constitution du corpus d'apprentissage a été choisi de manière à représenter diverses catégories de programmes actuellement diffusés à la télévision, en concertation avec france.tv access, et les autres partenaires impliqués dans les lots 4 et 5 du projet ROSETTA [2]. Nous avons en particulier classé les programmes recueillis selon des genres identifiés par des experts métiers du domaine, qui correspondent aux catégories utilisées par les fournisseurs de données. Dans la partie parallèle du corpus pour l'apprentissage, les genres télévisuels recensés sont les suivants : dessin animé, documentaire, fiction, jeu, journal, magazine, politique, et vulgarisation. Dans la partie non-alignée du corpus (Section 5.2.1.5), deux genres supplémentaires sont présents : enseignement (vidéos de cours de collège et lycée), et programme court. La fréquence et la durée des programmes ont été prises en compte, en veillant à ce que le traitement des données relatives à cette sélection puisse suivre le rythme des diffusions : le volume hebdomadaire de vidéos collectées a ainsi initialement été estimé à environ 50 h (en pratique il s'est révélé être plutôt entre 30 h et 40 h). Nous avons également fait en sorte d'éviter un déséquilibre trop important entre les émissions de type `stock` et `direct` (leurs proportions prévisionnelles ayant été calculées à 60 % et 40 % respectivement). Étant donné les différences significatives existant entre ces stratégies de production (les sous-titres `stock` sont rédigés en amont de la diffusion, tandis que les sous-titres `direct` sont réalisés pendant la diffusion selon une stratégie de répétition des paroles, voir Section 5.3), nous avons souhaité étudier l'incidence qu'elles pouvaient avoir dans les données, et dans les résultats des traitements avals.

5.2.1.2 Transcription automatique

Les instances de programmes collectées, qui arrivent au fur et à mesure des diffusions, sont transcrites automatiquement (mot-pour-mot) en utilisant le système VoxSigma développé conjointement par Vocapia Research et le LISN [3]. Ce système comporte un modèle acoustique hybride HMM-TDNN (*Hidden Markov Model*, *Time Delay Neural Network*) et un modèle de langue standard 4-gramme, entraînés sur de grandes quantités de données. Il produit des transcriptions automatiques qui sont segmentées en phrases selon les

2. Lot 4 : module de traduction multilingue de sous-titres adaptés. Lot 5 : module de génération de la langue des signes française – LSF.

3. Voir `www.vocapia.com/speech-to-text-technology.html`

1er invité écrire est-ce trahir, hé bien c'est la question que se pose également, Jean- Luc Coatalem lorsqu'il cherche à briser le silence qui entoure la mort de son grand-père, un grand-père qui n'a pas connu, arrêté en 1943 puis déporté en Allemagne et dont on a toujours refusé de parler dans une famille qui considère comme une trahison toute tentative d'explication Jean-Luc.

FIGURE 5.2 – Résultats de la transcription automatique d'un extrait de l'émission *La grande librairie* du 23/10/2019. Cet exemple illustre la forme des transcriptions produites par l'outil, à savoir des phrases longues, portant des marques de l'expression orale, avec de possibles erreurs (« un grand-père qu['il] n'a pas connu »). La transformation attendue pour le système de transduction étant donc de convertir en un style écrit, de resegmenter, et si possible de corriger les erreurs de transcription.

WER	**Avec aide ST**		**Sans aide ST**	
Émission	strict	lax	strict	lax
Jeu [s]	53,7	38,8	55,5	41,2
Fiction [s]	44,2	32,5	46,4	35,3
Magazine [s/d]	35,4	22,3	37,3	24,6
Politique [s/d]	28,1	15,2	29,6	16,9
Journal [d]	23,8	11,5	25,7	13,7
Tout	34,4	21,7	36,3	23,9

TABLE 5.1 – Évaluation sur le corpus de test de la qualité des transcriptions automatiques produites dans des conditions « optimistes » (avec sous-titrage) et « réalistes » (sans sous-titrage). Nous calculons deux scores : le premier (strict) correspond à une évaluation stricte des correspondances entre mots, alors que la seconde (lax) ignore les distinctions majuscule / minuscule et les ponctuations.

tours de parole, identifiés après un processus utilisant des modèles de mélange gaussien et un algorithme de regroupement des segments de parole. Les transcriptions sont aussi ponctuées automatiquement par un modèle 4-gramme, et elles respectent les principales règles typographiques (majuscule en début de phrase, pour les noms propres, etc.). Ce système délivre des performances à l'état de l'art pour la transcription du français, avec un taux d'erreur de mots (*Word Error Rate*) variant entre 10 et 40 % environ selon les émissions du corpus [4] (voir Tableau 5.1) : les meilleurs scores correspondant à de la parole préparée (par exemple dans les journaux télévisés), et les moins bons à de la parole spontanée ou peu rédigée, potentiellement bruitée par l'environnement (par exemple dans les jeux télévisés).

Le système VoxSigma intègre aussi un mode de transcription aidée par un texte auxiliaire. Deux transcriptions automatiques sont donc calculées pour chaque émission : l'une

4. Ces taux d'erreur ont été calculés par rapport à une transcription humaine de référence, considérée comme une version « idéale » de la transcription automatique.

simule la situation favorable où des documents préparatoires à la transcription sont disponibles, qui sont utilisés pour adapter le système de transcription à l'émission. À cet effet, nous fournissons au décodeur de parole la compilation des informations textuelles extraites du fichier de sous-titre (en filtrant néanmoins celles que le fichier identifie comme indications sonores ou musicales). La seconde simule une situation moins favorable où aucune information sur l'émission n'est disponible et le décodeur n'est alors pas adapté. Dans nos développements, nous avons principalement utilisé la version adaptée du décodeur. Un contraste avec la version non-adaptée est donné à la section 6.4.6.

Un exemple de sortie du système de transcription automatique est reproduit à la figure 5.2 et les performances brutes de ce système sont présentées dans le tableau 5.1.

Les résultats du tableau 5.1 on été obtenus en comparant mot à mot, sur le corpus de test (voir Section 5.2.2), les résultats bruts de la transcription automatique (assistée ou non-assistée) avec une transcription humaine de référence produite par la société ELDA dans le cadre d'une prestation de sous-traitance. La réalisation de ces transcriptions de référence a été conduite en essayant de reproduire au mieux les conventions de transcription adoptées par le système automatique. La métrique utilisée est le *taux d'erreur de mots* ou *Word Error Rate* (WER), qui agrège les erreurs correspondant à des ajouts, des omissions, et des substitutions. Nous donnons ici des résultats moyennés par catégories d'émissions qui permettent d'apprécier la qualité générale de l'entrée qui sera traitée par le système de sous-titrage. Des résultats plus complets sont donnés dans le tableau 5.8.

5.2.1.3 Alignement et « parallèlisation » du corpus

Le texte de la transcription ainsi obtenu est alors aligné avec celui des sous-titres, afin de pouvoir reconstituer des paires de segments parallèles qui sont nécessaires à l'apprentissage et l'évaluation automatique du système. Cet alignement est principalement fondé sur la comparaison caractère par caractère des deux segments textuels sur la base des opérations d'édition usuelles (insertion, substitution, délétion, copie), et est réalisé en pratique par composition avec un transducteur fini pondéré d'édition [5] et application d'un algorithme de plus courte distance (Mohri, 2002). Nous mettons en place diverses heuristiques pour atténuer par exemple l'incidence des différences portant sur la casse ou sur le type de ponctuation. L'utilisation du transducteur d'édition [6] nous permet aussi notamment de favoriser les suites homogènes d'opérations (résultant en des plages d'alignement mieux définies), par la prise en compte du contexte de l'opération précédente (voir la Figure 5.3).

Nous avons décidé d'utiliser la segmentation calculée par le système de transcrip-

5. Nous suivons un calcul semblable à celui détaillé à la Section 4.3.1.2
6. À l'aide de la librairie *Pynini* (Gorman, 2016).

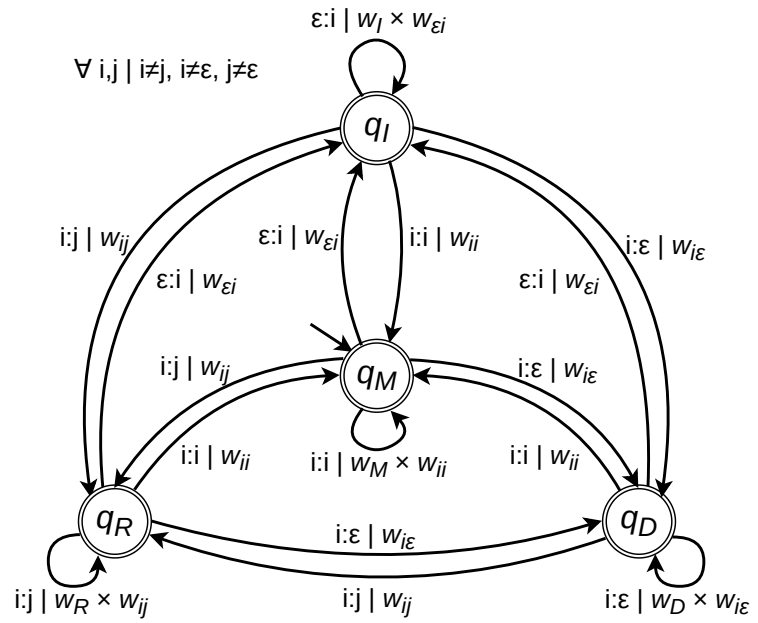


FIGURE 5.3 – Transducteur d'édition, permettant la comparaison des chaînes de caractères. Les états correspondent au contexte de l'opération précédente : insertion (q_I), substitution (q_R), suppression (q_D), copie (q_M). Le poids des transitions dépend des caractères mis en jeu (une table de pondération définit les w_{ij}, de façon strictement positive en général). Pour favoriser la succession d'opérations identiques, nous fixons $w_I, w_R, w_D, w_M \leqslant 1$ (les valeurs ont été testées pour accroître la proportion de plages de mots alignées à l'identique entre les deux chaînes).

tion automatique comme base de l'alignement. Ces segments sont assez longs (environ 40 mots en moyenne) et correspondent généralement à plusieurs tronçons de sous-titres (voir le Tableau 5.2) [7]. Les autres informations délivrées par le système de transcription (locuteurs, pauses, etc.) ne sont pas utilisées. Le processus de pré-traitement complet est représenté sur la figure 5.4.

À l'issue de l'alignement, une partie des phrases transcrites ne sont appareillées avec aucun sous-titre ; soit parce que le texte de sous-titres correspondant a été aligné avec le segment précédent ou suivant, soit parce que la phrase avait simplement été coupée lors du sous-titrage, comme pour l'exemple TR2-ST2 du tableau 5.3. Pour l'apprentissage des modèles de tels segments sont filtrés du corpus. De même, les paires présentant une trop grande dissimilitude [8] sont écartées avant l'apprentissage. Les nombres de segments retenus après filtrage sont donnés dans le tableau 5.4.

7. Nous avons développé les notations `<br>` et `<p>` avant d'avoir connaissance de celles de Karakanta et al. (2020b) ; nous les utilisons dans ce chapitre et dans le suivant. `<br>` équivaut à `<eol>`, et `<p>` équivaut à `<eob>`

8. Si la distance d'édition (Levenshtein) entre le segment transcrit et le segment sous-titre est supérieure à 40 %.

TR	Tout au long de la journée, des orages violents, de fortes pluies et quelles conséquences pour la population, faisons le point ce soir sur cette soudaine montée des eaux et sur les vents violents qui ont soufflé cet après-midi, dans les Bouches-du-Rhône à Marignane et je vous le disais sur la Côte-d Azur à à Valbonne Vence ou encore à Nice, Alexandre Christophe Larocca.
ST	Des orages violents, de fortes pluies et quelles conséquences pour <p> la population ? <p> Faisons le point sur cette soudaine montée des eaux et sur les vents <p> violents qui ont soufflé cet après-midi... <p>
TR	Le mot piperade vient du du basque bipède qui veut dire piment, j'insiste, une fois de plus, parce que, en fait, c'est bien que la base de cette recette est certes la tomate, mais surtout le piment donc le poivron, c'est une erreur pour la ratatouille.
ST	J'insiste une fois de plus car en fait, c'est bien que la base <p> de cette recette est certes la tomate <p> mais surtout le piment. Le poivron, c'est une erreur. <p> C'est pour la ratatouille.

TABLE 5.2 – Exemples de fragments de transcription automatique TR (source) et des blocs de sous-titres associés ST (cible), le premier issu d'un extrait de l'émission *Journal 20h00* du 03/11/2019, et le second issu d'un extrait de l'émission *Les carnets de Julie avec Thierry Marx* du 03/08/19. Les balises représentent la segmentation à l'affichage : saut de ligne au sein d'un bloc `<br>` et fin de bloc `<p>`.

5.2.1.4 Volume de données et versions du corpus d'apprentissage

Les systèmes de sous-titrage ont été construits avec des versions de taille croissante des données d'apprentissage, puisque l'acquisition de données s'est déroulée sans arrêt sur toute la longueur du projet. Quatre points d'étape ont été réalisés : notés V1 à V4, leurs tailles sont rapportées dans le tableau 5.4. Pour l'essentiel des expériences du chapitre 6, le corpus V4 a été utilisé. Nous donnons néanmoins un contraste entre des modèles de sous-titrage appris sur les quatre points d'étape du corpus dans la section 6.4.7.

La répartition par genre et type d'émissions pour la version la plus complète du corpus est donnée dans le tableau 5.5.

5.2.1.5 Sous-titres complémentaires et rétro-traductions

La production de données d'apprentissage est un processus coûteux qui implique l'exploitation d'un système de reconnaissance vocale. En complément, nous avons également choisi d'utiliser des *données pseudo-parallèles artificielles*, qui sont directement dérivées des fichiers de sous-titres sans qu'il soit besoin de traiter la piste son. Nous nous inspirons des méthodes de rétro-traduction qui ont fait leurs preuves en traduction automatique neuronale (Sennrich et al., 2016; Burlot & Yvon, 2018; Edunov et al., 2018) et qui consistent à « inverser » le processus de traduction de manière à construire des données associant

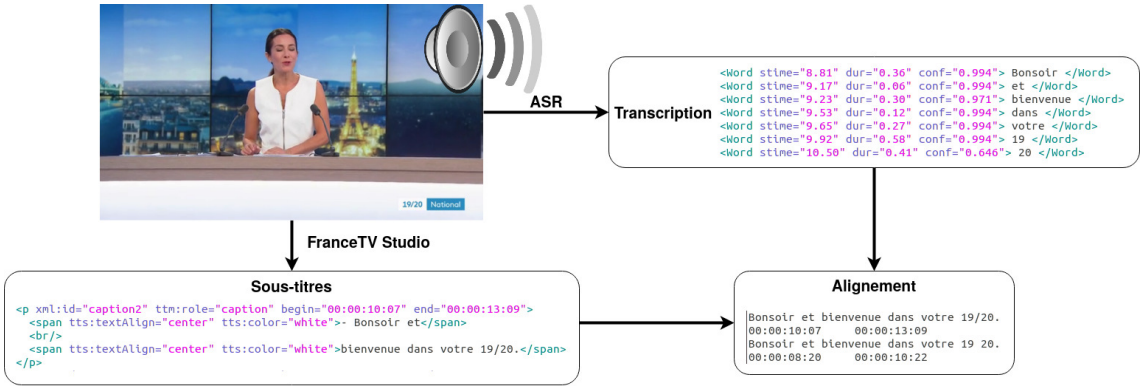


FIGURE 5.4 – Préparation des données alignées.

une transcription artificielle avec un sous-titre correct. Cette méthode permet en particulier d'améliorer l'apprentissage du décodeur du système de traduction. La rétro-traduction présente l'avantage (par rapport à d'autres méthodes de synthèse de données) de ne pas recourir à des données extérieures, et de produire des phrases cibles syntaxiquement correctes, dont le genre télévisuel est connu.

La génération des pseudo-transcriptions est mise en œuvre de la manière suivante. Pour chaque évolution du corpus, en exploitant l'intégralité des données parallèles disponibles, nous avons entraîné un système encodeur-décodeur qui inverse le processus de sous-titrage et produit des pseudo-transcriptions à partir des sous-titres. Ce système utilise la même architecture *Transformer* (nombre de couches, dimensions internes) que le système de sous-titrage correspondant (voir la Section 6.2.2). Comme le système de reconnaissance de parole tend à produire de longs segments par rapport à ceux présents dans le texte des sous-titres (une phrase transcrite correspondant généralement à plusieurs phrases de sous-titres ; voir le Tableau 5.2), nous concaténons aléatoirement les phrases de sous-titres en de plus longues séquences préalablement à la rétro-traduction. Le nombre de phrases à rassembler est échantillonné selon une loi normale centrée sur 3 [9]. À l'étape V2 du corpus, ce système obtient un score BLEU [10] de 64,7 sur les données de test (test-1) en comparant les énoncés artificiellement bruités aux sous-titres de référence. Cela suggère que les pseudo-transcriptions artificielles restent très proches des sous-titres de référence, et sont donc considérablement moins bruitées que les transcriptions réelles. Des exemples de pseudo-transcriptions sont donnés dans le tableau 5.6.

Pour nos expériences, nous n'avons pas rétro-traduit l'ensemble des sous-titres disponibles, mais avons effectué une sélection sur la base du genre des émissions. La répartition par genre et type d'émission des données rétro-traduites est dans le tableau 5.7. Notons

9. Valeur qui correspond au ratio observé en pratique entre les segments transcrits automatiquement et les phrases de sous-titres.
10. Les métriques sont décrites en détail à la section 2.4.

TR1	Alors là aussi je ne veux pas à chaque fois, repousser la question à la semaine prochaine, mais il y a des choses, ce travail, nous menons avec les régions sur l'offre de transport, ce que nous faisons, c'est que nous faisons, nous essayons, nous organisons le maximum de transports du quotidien partout dans les régions de manière à avoir.
TR2	Très concrètement, mais concrètement dans les transports jour pas si Djebbari concrètement, vous voyez, il y a, il y a les les départements.
TR3	Moi je réponds, je vous réponds quoi concrètement et sur le sur les couleurs Vert, rouge, nous réglons ce que nous voulons l'objectif du gouvernement, c'est que s'agissant des transports longue distance, nous ayons une offre qui soit relativement réduite et qui finalement satisfasse les besoins de transport essentiels, on l'a dit, les réseaux professionnels ou les motifs familiaux impérieux [...] donc il y aura toujours sur les transports longue distance des offres, ils seront contrôlées régulées, nous aurons des réservations.
ST1	Je ne veux pas à chaque fois repousser la question à la semaine prochaine, mais c'est un travail que nous menons avec les régions sur l'offre de transport. Nous organisons le maximum de transports du quotidien partout dans les régions.
ST2	-
ST3	Ce que nous voulons, c'est que s'agissant des transports longue distance, nous ayons une offre qui soit relativement réduite et qui concerne juste les transports essentiels comme les réseaux professionnels ou les motifs familiaux impérieux. Il y aura toujours sur les transports longue distance des offres qui seront contrôlées, régulées.

TABLE 5.3 – Exemples de fragments de transcription automatique TR1-2-3 (source) et des blocs de sous-titres associés ST1-2-3 (cible), issus d'un extrait de l'émission *Dimanche en politique* du 03/05/2020. Aucun sous-titre n'est associé à TR2. Les balises représentant la segmentation des sous-titres sont ici omises.

que nous n'avons mis en œuvre la rétro-traduction qu'à partir de l'étape V2 du corpus, et que le volume de données synthétiques est approximativement le même entre V2, V3, et V4 (voir Tableau 5.4) : au cours du développement de V3, pour des raisons pratiques de limitation de l'espace de stockage, l'API de france.tv access a restreint les émissions mises en ligne à la sélection de base du corpus.

5.2.2 Corpus de test

Nous avons choisi pour nos tests de sélectionner au hasard un petit nombre d'émissions, en collectant toutes les émissions traitées par notre chaîne de traitement les 3 août et 3 novembre 2019, ainsi que les 3 mars et 3 mai 2020. Notre objectif étant de construire un unique système multigenre, et pas un système par genre, nous avons utilisé la distribution naturelle des émissions dans nos corpus, qui optimise les performances sur des données de test représentant également une distribution naturelle (ayant été arbitrairement échantillonnées le 3 de chaque mois), qui pourrait simuler un scénario de production. Néanmoins, pour mieux couvrir le genre « fiction », trois épisodes de la série « Un si grand soleil » ont été inclus dans l'ensemble de test. Les titres d'émission du corpus

Taille	**V1** (11/2020)		**V2** (02/2021)		**V3** (07/2021)		**V4** (11/2021)	
	aln	psd	aln	psd	aln	psd	aln	psd
Heures (h)	1 253	–	1 620	1 276	2 304	1 278	2 878	1 278
Sous-titres	0,96M	–	1,63M	1,19M	2,31M	1,19M	2,92M	1,19M
Segments	0,32M	–	0,41M	0,29M	0,62M	0,29M	0,78M	0,29M
Segments*	0,16M	–	0,26M	0,28M	0,38M	0,28M	0,48M	0,28M
Mots TR	13,04M	–	17,04M	–	23,81M	–	29,88M	–
Mots ST	9,30M	–	12,20M	8,84M	17,00M	8,86M	21,55M	8,86M

TABLE 5.4 – Analyse statistique des points d'étape du corpus d'apprentissage, où nous distinguons les alignements avec transcription automatique (aln) et les données pseudo-parallèle obtenues via rétro-traduction (psd). Le corpus V1 ne comporte pas de données pseudo-parallèle. Le nombre de segments après filtrage est noté par (*).

Genre/Stratégie	**(h)**	**Sous-titres**	**Segments**	**Mots TR**	**Mots ST**	**%**
Dessin animé [s]	8	7K	2K	0,05M	0,04M	0,2
Documentaire [s]	162	145K	49K	1,26M	1,04M	4,8
Fiction [s]	143	134K	46K	1,05M	0,86M	4,0
Jeu [s]	586	549K	190K	5,22M	3,39M	15,7
Journal [d]	587	681K	157K	6,16M	5,36M	24,9
Magazine [s/d]	1 285	1 293K	317K	14,83M	9,98M	46,3
Politique [s/d]	104	104K	19K	1,26M	0,85M	3,9
Vulgarisation [s]	4	4K	1K	0,04M	0,03M	0,1
`direct`	1 217	1 284K	272K	13,65M	10,06M	46,7
`stock`	1 662	1 633K	509K	16,23M	11,49M	53,3
Tout	2 878	2 917K	780K	29,88M	21,55M	100

TABLE 5.5 – Distribution par genre et type d'émission des données du corpus V4 (11/2021). Les pourcentages sont calculés par rapport au nombre de mots dans les sous-titres.

de test font partie de ceux présents dans le corpus d'apprentissage, et leurs périodes de diffusion se chevauchent. Les instances d'émission elles-mêmes ont été retirées des données d'apprentissage et ont fait l'objet d'un traitement séparé, incluant en particulier une transcription manuelle de référence.

Le corpus de test est constitué de deux parties distinctes : l'une (test-1) destinée à être utilisée pour évaluer les évolutions successives de nos systèmes ; la seconde (test-2) destinée à n'être utilisée qu'une fois les développements réalisés, pour permettre de mesurer de manière plus juste la progression des performances du sous-titrage automatique, en calculant les scores automatiques rétrospectivement sur ce second corpus de test.

Le détail des émissions est dans le tableau 5.8, des statistiques concernant leur durée et

ST	Madame, Monsieur, bonjour. <p> Dans l'actualité de ce dimanche 15 mars, la France est désormais au <p> stade 3 de l'épidémie. <p> Nous verrons ce que cela change dans votre vie quotidienne. <p> En tout cas, depuis minuit, les lieux publics non indispensables à <p> la vie du pays sont fermés. <p>
TR̂	Madame monsieur, bonjour, bonjour, merci d'être avec nous dans l'actualité de ce dimanche 15 mars la France est désormais au stade 3 de l'épidémie, nous verrons ce que cela change dans votre vie quotidienne, en tout cas depuis minuit, les lieux publics non indispensables à la vie du pays sont fermés, on va voir tout cela avec vous, Valérie Heurtel et Jean-Pierre Magnaudet.
ST	Surtout pas Benjamin. <p> Mais je vous ai vus. Oui, tu nous as vus <p> en train de parler discrètement, à l'écart, peut-être, <p> mais pas parce qu'il y avait une histoire entre nous. <p>
TR̂	Surtout pas, Benjamin, mais je vous ai vu, oui, oui, oui, tu nous as vus en train de parler discrètement à l'écart, peut-être, mais pas parce qu'il y avait une histoire entre nous.

TABLE 5.6 – Exemples de pseudo-transcriptions obtenues par « rétro-traduction » (modèle appris sur le corpus V4).

leur taille sont dans le tableau 5.9.

5.3 Analyse selon la stratégie de sous-titrage

Un type de variabilité est directement lié à la stratégie de production des sous-titres, qui peuvent être selon les émissions réalisés en direct, c'est-à-dire au fur et à mesure que l'émission se déroule, ou bien durant une étape de post-production. Nous désignerons ces deux types de sous-titres par `direct` et `stock`. La production en direct est soumise à des contraintes temporelles et obéit à une très forte logique d'efficacité. Elle s'appuie sur des systèmes de transcription automatique [11] et de correction d'orthographe, ce qui conduit en particulier à des sous-titres qui « collent » au plus près du contenu sonore ; avec parfois des décrochages dus à l'impossibilité de suivre le rythme des échanges verbaux du direct (c'est le cas dans le passage présenté dans le Tableau 5.3, avec pour conséquence le problème d'alignement mis en exergue). En comparaison, la génération de sous-titres en post-production, qui subit d'autres contraintes (par exemple : économiques), peut prendre une plus grande distance avec le flux audio.

Ces différences sont objectivables et nous les illustrons dans le tableau 5.10 et la figure 5.5, qui donnent quelques résultats d'analyses lexicométriques du corpus d'apprentissage. Une première différence apparaît clairement entre les sous-titres `direct` et `stock` : les premiers sont plus verbeux dans l'absolu avec environ $11,2$ $kmots/h$, alors que le rythme moyen de parole pour le `stock` est seulement environ $9,8$ $kmots/h$. Toute-

11. Pour avoir de meilleures performances, la reconnaissance de parole est appliquée alors qu'un opérateur (dit « perroquet ») répète les paroles dans un environnement sans bruit.

Genre/Stratégie	**(h)**	**Sous-titres**	**Mots ST**	**%**
Dessin animé [s]	9	8K	43K	0,5
Documentaire [s]	92	78K	563K	6,4
Fiction [s]	57	55K	338K	3,8
Jeu [s]	54	56K	309K	3,5
Journal [d]	185	176K	1 425K	16,1
Magazine [s/d]	637	602K	4 623K	52,2
Politique [d]	14	12K	107K	1,2
Vulgarisation [s]	10	10K	77K	0,9
Enseignement [s]	221	189K	1 364K	15,4
Prgm court [s]	2	1K	12K	0,1
`direct`	782	737K	5 758K	65,0
`stock`	496	452K	3 102K	35,0
Tout	1 278	1 189K	8 860K	100

TABLE 5.7 – Distribution par genre et type d'émission des données rétro-traduites pour V3 et V4 (quasiment identique pour V2). Les pourcentages sont calculés par rapport au nombre de mots dans les sous-titres.

fois ces moyennes sont lissées sur la durée totale des émissions ; si les périodes de silence sont exclues du calcul, nous observons que la vitesse d'élocution est comparable, étant même légèrement supérieure pour `stock` (les émissions de `stock` reposent davantage sur l'image et contiennent davantage de silences, qui représentent 24 % de la durée, par opposition au `direct` où ils ne comptent que pour 8 % du temps total). Les sous-titres en `stock` correspondent aussi à un plus grand nombre de phrases, qui sont donc plus courtes (7,5 mots par phrase en moyenne contre 12,7 mots pour les sous-titres `direct`), et probablement plus travaillées. En effet, on remarque sur la figure 5.5 que les taux de compression (CpR) sont relativement proches (76,0 % pour `direct`, 74,6 % pour `stock`), alors que la distance d'édition normalisée (NED) est nettement plus grande pour `stock` (50,4 %) que pour `direct` (39,3 %). En termes d'opérations d'éditions, cela signifie que les sous-titres `direct` et `stock` réalisent une proportion comparable de pures suppressions (environ 25 % de la transcription), mais que ceux en `stock` opèrent davantage de ré-écriture, que ce soit par substitution ou par couple suppression/insertion [12]. Le score BLEU_{nb} (46,8 pour `direct`, 34,9 pour `stock`) confirme la plus grande proximité entre la transcription et les sous-titres dans le cas du `direct`. La transcription et les sous-titres sont globalement plus simples à la lecture pour les émissions de `stock` (le score de lisibilité FRE est plus élevé pour la transcription et les sous-titres `stock`) : cela correspond notamment au fait que certaines de ces émissions contiennent moins d'interventions spontanées,

12. Ici, taux de compression et distance d'édition étaient calculés au niveau des caractères

F5	03/08/19	C dans lair	D	Magazine	1	66	24,44
F3	03/08/19	Les carnets de Julie	S	Magazine	1	54	30,57
F5	03/08/19	Echappees belles	S	Magazine	1	90	29,37
F2	03/11/19	Tout le monde veut prendre sa place	S	Jeu	1	48	37,14
F2	03/11/19	Journal 20h00	D	Journal	1	30	17,32
F5	03/11/19	C politique	D	Politique	1	78	16,25
F3	03/11/19	1920 Journal national	D	Journal	1	24	11,93
F3	03/11/19	Dimanche en politique	D	Politique	1	42	19,49
F5	03/11/19	C politique la suite	D	Politique	1	54	18,8
F5	02/03/19	Echappees belles	S	Magazine	1	90	27,23
F2	25/02/19	Un si grand soleil	S	Fiction	2	24	33,36
F2	05/03/19	Un si grand soleil	S	Fiction	2	24	33,52
F2	07/03/19	Un si grand soleil	S	Fiction	2	24	38,73
F2	03/03/20	Journal 20h00	D	Journal	2	36	13,43
F2	03/03/20	Journal 13h00	D	Journal	2	42	14,15
F5	03/03/20	C dans l'air	D	Magazine	2	66	19,43
F2	03/03/20	Un si grand soleil	S	Fiction	2	24	37,1
F5	03/03/20	Passage des arts	S	Magazine	2	24	24,15
F3	03/03/20	Plus belle la vie	S	Fiction	2	24	33,65
F5	03/03/20	Le magazine de la sante	D	Magazine	2	54	15,58
F2	03/03/20	Ca commence aujourd'hui	S	Magazine	2	66	25,92
F3	03/03/20	Des chiffres et des lettres	S	Jeu	2	30	45,21
F3	03/05/20	Dimanche en politique	D	Politique	2	42	12,45
F5	03/05/20	C politique	D	Politique	2	78	17,38
F3	03/05/20	1920 Journal national	D	Journal	2	24	13,25
F2	03/05/20	Journal 20h00	D	Journal	2	48	11,99

TABLE 5.8 – Les jeux de tests : pour chaque émission, nous indiquons le type de sous-titrage ([S]tock ou [D]irect), le genre télévisuel, le test set, la durée (en minutes) et le taux d'erreur de mots (WER) de la reconnaissance vocale (sans aide des sous-titres, en ignorant la casse et la ponctuation).

qui forment des énoncés moins structurés et plus longs (les pauses étant plus fréquentes et plus appuyées dans les discours préparés, ce qui est important en particulier pour la détection de fin de phrase par le système de reconnaissance vocale). La différence de FRE (nettement plus grande pour le `direct`) entre la transcription et les sous-titres suggère une simplification plus importante dans le cas du `direct`. Elle découle en fait de la réduction plus importante de la longueur moyenne des phrases dans le `direct` (38 mots de moins en moyenne, p. opp. 24 mots de moins dans le `stock`), et de l'augmentation plus substantielle de la longueur des mots dans le `stock` (0,32 caractère, p. opp. 0,09 caractère dans le `direct`). Du point de vue des normes spatiale et temporelle, les sous-titres produits en `direct` s'avèrent de façon générale plus denses ; le nombre caractères par ligne (CPL) et le nombre de caractères par seconde (CPS) sont légèrement plus faibles pour le `stock`, et la recommandation de 15 car/s est bien moins souvent respectée dans les sous-titres

Taille	test-1	test-2
Heures (h)	9,6	10,5
Sous-titres	8 840	10 111
Segments	2 189	2 527
Mots TR	103 487	107 308
Mots ST	68 195	77 014

TABLE 5.9 – Statistiques des corpus de test.

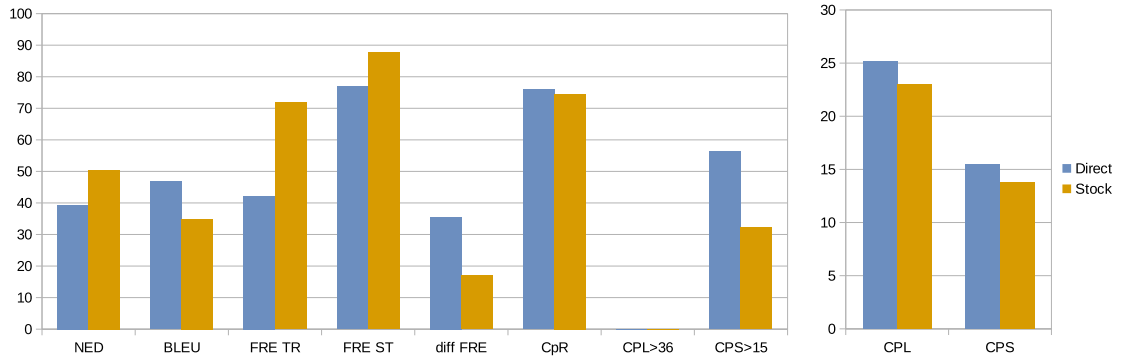


FIGURE 5.5 – Comparaison entre les données `direct` et `stock` au sein du corpus, selon plusieurs métriques (décrites aux Sections 2.4) et 3.5.

en `direct` (plus de la moitié des sous-titres produits dans ces conditions dépassent cette borne).

5.4 Analyse par genre télévisuel

Au delà de la stratégie de production, les sous-titres, ainsi que les transcriptions qui leur sont associées, sont affectés par des caractéristiques propres aux émissions et à leur format. La nature de certains programmes fait qu'une partie des phrases ont une structure assez spécifique : énoncé d'une question de culture générale ou réponse très courte d'un candidat pour les jeux télévisés, annonce des titres d'actualités ou d'un reportage dans un journal (on peut remarquer, dans le premier exemple du Tableau 5.2, les noms de journalistes énoncés à la fin de la phrase introduisant le sujet ; trait qui est d'ailleurs imité par le système de rétro-traduction dans le premier exemple du Tableau 5.7). Et les thèmes abordés de façon récurrente (sujets de politique, santé, gastronomie, sciences) ont une incidence sur le vocabulaire employé. De plus, une variabilité semble exister sur les standards attendus pour différents types de programmes. En effet la figure 5.6 montre que les sous-titres de journaux et d'émissions politiques se distinguent de ceux des autres genres en dépassant très fréquemment la recommandation de 15 car/s (63 % des sous-titres de journaux, et 55 % des sous-titres d'émissions politiques). Ces deux genres en

Stratégie/Genre	**Vrb**	**SpR**	**StR**$_{TR}$	**StR**$_{ST}$	**StL**$_{TR}$	**StL**$_{ST}$	**WdL**$_{TR}$	**WdL**$_{ST}$
`direct`	11,2k	12,6k	223	653	50,3	12,7	5,51	5,60
`stock`	9,8k	13,3k	306	919	31,9	7,5	5,21	5,53
Dessins animé [s]	6,3k	9,2k	264	1 057	23,9	5,4	5,15	5,55
Documentaire [s]	7,8k	10,4k	300	694	25,9	9,3	5,34	5,53
Fiction [s]	7,4k	10,4k	323	1 132	22,7	5,3	5,06	5,36
Jeu [s]	8,9k	13,9k	325	969	27,5	6,0	5,30	5,67
Journal [d]	10,5k	11,6k	268	698	39,2	13,1	5,63	5,63
Magazine [s/d]	11,5k	13,8k	246	774	46,8	10,0	5,26	5,51
Politique [s/d]	12,1k	13,8k	179	626	67,9	13,0	5,43	5,59
Vulgarisation [s]	10,2k	12,0k	182	820	55,8	10,0	5,46	5,60
Tout	10,4k	13,0k	271	807	38,3	9,3	5,35	5,56

TABLE 5.10 – Comparaison d'indices textuels selon les stratégies de sous-titrage et plusieurs genres télévisuels. Les catégories « Magazine » et « Politique » contiennent un mélange d'émissions `direct` et `stock`, alors que les autres sont soit exclusivement `direct` [d] ou `stock` [s]. La verbosité (Vrb) est mesurée par le nombre de mots prononcés rapporté à la durée totale de l'émission ; la vitesse d'élocution (SpR) est mesurée en rapportant le nombre de mots prononcés à la durée de parole effective (sans compter les périodes de silence). StR est le nombre de phrases par heure ; StL est la longueur moyenne des phrases en nombre de mots ; WdL est la longueur moyenne des mots en nombre de caractères.

particulier semblent bénéficier d'une tolérance concernant la fréquence d'affichage, au point que même l'indice CPS calculé en moyenne sur les sous-titres est au dessus de la limite des 15 car/s. Notons que la fréquence d'affichage est à mettre en lien avec la vitesse d'élocution réelle (SpR*) et le taux de compression pour les sous-titres (CpR). Les journaux correspondent à une vitesse d'élocution intermédiaire ($11,6$ $kmots/h$), mais font l'objet d'une compression relativement faible (CpR=87 %) ; peut-être parce que les informations sont déjà formulées efficacement dans la parole initiale. En comparaison, les émissions politiques partent d'une vitesse d'élocution plus élevée ($13,8$ $kmots/h$), mais arrivent à une fréquence d'affichage comparable en comprimant davantage les sous-titres (CpR=69 %).

D'autres divergences en relation avec le genre télévisuel concernent plus particulièrement le contexte de production de la parole, et ses conséquences sur la transcription automatique. Une différence essentielle est celle entre discours préparé et discours spontané. Une prise de parole préparée à l'avance tend à se rapprocher du style écrit, tandis que la parole spontanée suit des règles de syntaxe (p. ex. la surcharge *parce que, en fait* dans le deuxième exemple du Tableau 5.2) ou de macro-syntaxe [13] (p. ex. la succession

13. Le terme de « macro-syntaxe » désigne une organisation de la langue parlée qui ne repose pas sur des catégories grammaticales, mais sur délimitation d'unités, les « noyaux », par la prosodie (Blanche-

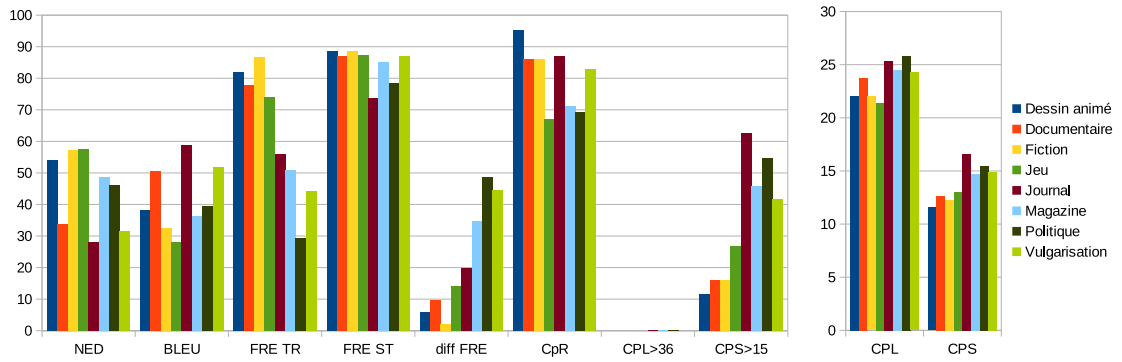


FIGURE 5.6 – Comparaison entre les données des genres télévisuels au sein du corpus d'apprentissage, selon plusieurs métriques (décrites aux Sections 2.4) et 3.5.

de noyaux autonomes dans le segment TR3 du Tableau 5.3 : *moi je réponds*, *je vous réponds quoi concrètement*) spécifiques à l'expression orale. La transformation requise pour la mise en sous-titres est ainsi moins complexe dans le premier cas que dans le second. D'autant que des propos non-préparés ont également propension à présenter des phénomènes tels que les hésitations, les erreurs et retouches, qui sont transcrits par le système de reconnaissance vocale comme le reste des mots. Sur la figure 5.6 nous pouvons voir que des émissions telles que les documentaires (`stock`), les programmes de vulgarisation (`stock`) et les journaux (`direct`), qui sont rédigées à l'avance, dans un registre généralement formel, montrent une grande proximité entre la transcription et les sous-titres (score BLEU_{nb} au dessus de 50, et distance d'édition en dessous de 34 %). Inversement, les jeux (`direct`) et les fictions (`stock`) qui sont constitués de parole spontanée (simulée, dans le cas des fictions) affichent une dissimilitude forte entre la transcription automatique et les sous-titres (NED=58 %, BLEU_{nb}=28 pour les jeux, et NED=57 %, BLEU_{nb}=32 pour les fictions).

Il faut également prendre en compte la qualité des prédictions proposées par le système de reconnaissance vocale, qui se répercute sur la qualité des sous-titres engendrés en aval (voir Section 6.4.6). La reconnaissance est notamment affectée par le débit de parole, la clarté de la prononciation, les dialogues avec recouvrement, et généralement la présence de bruits parasites. Par exemple, les jeux télévisés ont tendance à contenir beaucoup de séquences avec de la musique ou des rires, et des échanges rapides. En conséquence, ce genre de programmes présente un plus grand taux d'erreur de mots (WER) par rapport aux émissions politiques et aux journaux, comme le montre le tableau 5.1.

Remarquons pour finir que notre corpus présente une plus grande diversité de situations d'énonciation par rapport aux ensembles de données regroupant des conférences TED (p. ex. MuST-Cinema, voir Section 3.4) : celles-ci correspondent à un cadre rela-

Benveniste, 2010, p. 123).

tivement normalisé où dans lequel prend place une présentation préparée. En outre, les sous-titres de notre corpus ont été réalisés par des sous-titreurs professionnels, dans la perspective d'être utilisés par des personnes sourdes et malentendantes. En comparaison, les sous-titres de TED sont produits par des volontaires, et sont essentiellement destinés à la diffusion internationale (des sous-titres intralinguistiques sont très proches de la transcription mot à mot, et ne contiennent quasiment pas de simplification).

5.5 Conclusion

Dans ce chapitre nous avons présenté la création d'un corpus pour le sous-titrage automatique, dans le scénario d'une architecture en cascade. Nous avons conçu une chaîne de traitement pour (i) récupérer les données d'émissions télévisées par le biais d'une API mise en place par france.tv access, (ii) transcrire les vidéos à l'aide d'un système pré-existant (VoxSigma), et (iii) aligner les transcriptions et les sous-titres sur la base d'une comparaison au niveau des caractères. Quatre points d'étape ont été réalisés durant l'avancement de ce processus, qui correspondent à différentes tailles de corpus. Des données synthétiques ont été engendrées par rétro-traduction de sous-titres complémentaires. Le corpus final comprend environ 30 millions de mots transcrits (1278 heures de vidéos). Nous avons identifié deux classes majeurs (`stock`/`direct`) et huit genres télévisuels, qui d'après nos observations induisent des variations significatives dans les données, justifiant l'utilisation de méthodes d'adaptation pour le sous-titrage automatique (Chapitre 6).

À l'avenir il pourrait être intéressant de poursuivre l'analyse des exemples recueillis, en observant plus finement les types d'opération de simplification, ainsi que les erreurs de transcription commises, selon le genre télévisuel.

Chapitre 6

Production automatique de sous-titres

6.1 Introduction

Dans ce chapitre, nous étudions des stratégies de sous-titrage automatique en français inspirées des méthodes de traduction neuronales (Cho et al., 2014a; Bahdanau et al., 2015; Vaswani et al., 2017), la transcription vocale (automatique) jouant le rôle d'énoncé en langue source et le sous-titre (texte et segmentation) jouant le rôle d'énoncé en langue cible. En nous appuyant sur des données réelles, la principale question que nous essayons de traiter concerne *la variabilité des genres télévisuels et son impact sur la qualité des sous-titres automatiques*. Après avoir mesuré cette variabilité (Chapitre 5), nous comparons différentes méthodes, inspirées de travaux en traduction automatique, pour la prendre en charge à travers de modules de sous-titrage adaptés au genre. Trois approches sont implémentées et évaluées : une approche fondée sur la spécialisation par genre des taux de compression, une qui spécialise les représentations des énoncés sources (Kobus et al., 2017), une troisième, enfin, qui repose sur l'affinage (Freitag & Al-Onaizan, 2016) des modèles de traduction. Nos évaluations permettent alors de conclure que ces deux dernières méthodes améliorent la qualité des sous-titres produits, y compris lorsqu'elles sont combinées avec d'autres stratégies d'auto-apprentissage.

Nous présentons à la section 6.2 les méthodes utilisées pour réaliser l'adaptation du sous-titrage au genre. Le protocole expérimental est décrit à la section 6.3, en précisant en particulier la mise en place d'une évaluation humaine. La section 6.4 présente enfin les principaux résultats expérimentaux et diverses analyses complémentaires.

Nos contributions pour ce chapitre sont les suivantes :

— Nous mettons en place une chaîne entièrement automatisée de production de sous-titres utilisables.
— Nous montrons que les méthodes classiques d'adaptation au genre peuvent être transposées efficacement à la tâche de sous-titrage, et nous les comparons à une approche d'adaptation originale fondée sur le contrôle de longueur au cours du décodage.

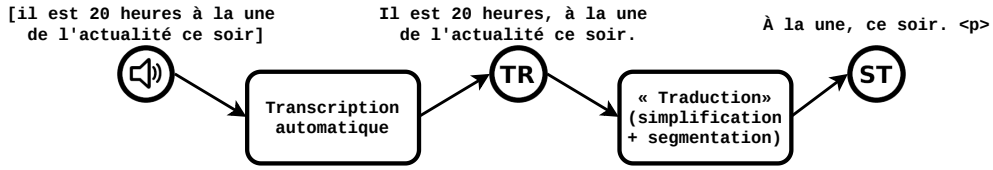


FIGURE 6.1 – Architecture globale pour le sous-titrage automatique. Nos expériences se concentrent sur la tâche de « traduction » de la transcription vers les sous-titres segmentés.

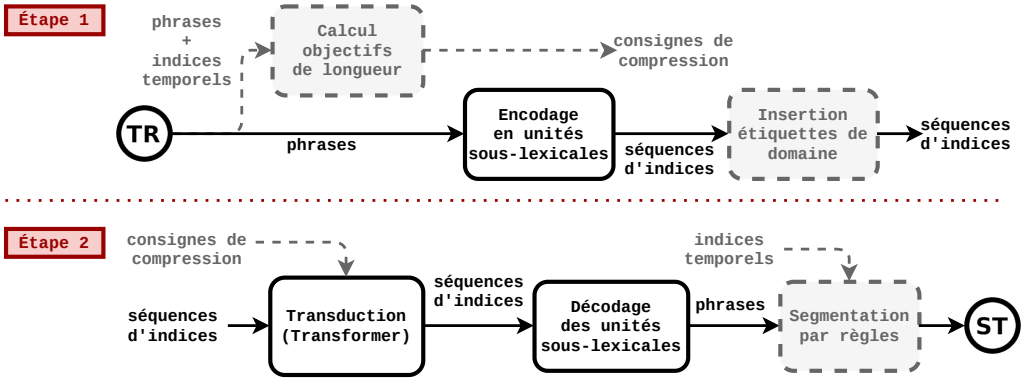


FIGURE 6.2 – Architecture détaillée pour la transduction de la transcription vers les sous-titres segmentés. En noir sont représentées les étapes partagées par tous les systèmes ; en gris et pointillé celles réalisées seulement par certains systèmes : le calcul de consignes de compression (introduites dans le *Transformer*) est spécifique aux systèmes fondés sur le contrôle de longueur (Sections 6.2.3.2 et 6.2.4.2) ; l'insertion d'étiquettes de domaine correspond à l'une des méthodes d'adaptation au genre (Section 6.2.4.1) ; la segmentation du contenu textuel des sous-titres par un module à règles intervient uniquement dans les systèmes de base qui n'effectuent pas conjointement simplification et segmentation (Section 6.2.3.1). Les indices temporels sont les périodes de prononciation des phrases (identifiées par l'outil de reconnaissance) ; les consignes de compressions sont des objectifs de longueur cible, modulés pour suivre un certain taux de compression (CpR), ou une fréquence d'affichage (CPS).

— Nous analysons de façon détaillée l'influence de plusieurs facteurs (usage de données synthétiques ou d'étiquettes de domaine, qualité de la transcription, taille du jeu d'apprentissage) sur les performances de nos systèmes.

6.2 Adaptation aux genres télévisuels

Nous décrivons dans cette section les méthodes utilisées pour construire nos systèmes de sous-titrage, dans leur version de base comme dans leur version adaptée. La figure 6.1 présente une vue générale de l'architecture en cascade commune à ces systèmes. La figure 6.2 apporte une vue détaillée sur la phase de transduction de la transcription vers les sous-titres segmentés.

6.2.1 Architectures pour le sous-titrage automatique

Un choix précoce de cette étude a été d'utiliser une architecture *en cascade* qui découple une phase de retranscription verbatim du contenu audio, de la phase d'élaboration et de génération des sous-titres. Cette décision est notamment motivée par la possibilité de disposer d'un outil de transcription vocale au meilleur état de l'art pour le français, dont l'utilisation nous a permis de nous focaliser sur la tâche de simplification / compression. Récemment, la traduction automatique de parole (Bérard et al., 2016; Karakanta et al., 2020a) ainsi que d'autres applications (Ghannay et al., 2018) ont vu la progression d'architectures *bout en bout* (aussi appelées *directes*), qui s'abstiennent d'utiliser une représentation symbolique intermédiaire. Il convient néanmoins de noter que l'approche en cascade obtient encore en général les meilleurs résultats, surtout si des données indépendantes pour la transcription et la traduction sont utilisées (comme expliqué à la Section 3.3.0.2).

Dans notre approche, la production de sous-titres repose sur la métaphore de la traduction, et s'appuie, à l'instar de nombreux travaux récents (Zhang et al., 2017; Zhang & Lapata, 2017; Matusov et al., 2019; Karakanta et al., 2020a), sur des architectures neuronales encodeur-décodeur. Selon cette métaphore, les échantillons de parole (pour les système de sous-titrage de bout en bout) ou leur retranscription automatique (pour les systèmes en cascade) sont encodés sous la forme d'une suite de vecteurs numériques, qui sont ensuite décodés pour engendrer de proche en proche les séquences de mots correspondant aux sous-titres.

La tâche de sous-titrage demande non-seulement de produire du texte, mais également d'engendrer des directives pour son affichage à l'écran et la resynchronisation avec la piste audio. Les indications de synchronisation portent sur des blocs entiers d'une ou deux lignes et correspondent à des indices temporels de début et de fin d'affichage du bloc : elles sont calculées dans notre processus à partir des périodes de parole identifiées par l'outil de transcription (en permettant à l'affichage de durer quelques secondes supplémentaires pendant les éventuels silences). Les directives d'affichage sont matérialisées par des balises qui sont insérées dans le flux textuel et signalent les fins de ligne (`<br>`) et les fins de blocs (`<p>`). Nous envisageons deux méthodes distinctes pour prédire la position des balises ; la première l'envisage comme une étape séparée du calcul du sous-titre et repose sur un module à base de règles détaillé ci-dessous ; la seconde méthode est *une méthode intégrée* qui permet d'engendrer simultanément le contenu linguistique et les marques de segmentation. Pour réaliser cet apprentissage de bout en bout, les balises de segmentation sont directement insérées dans le flux textuel lors de l'apprentissage et du décodage. Le système est alors libre de les produire comme il produirait n'importe

quelle autre unité de son vocabulaire de sortie. Une illustration de ces sorties enrichies est donnée au tableau 6.1. Un dernier composant de notre architecture rassemble le contenu textuel accompagné de consignes de segmentation et de synchronisation temporelle en un fichier au format `ttml` qui peut être directement utilisé dans des systèmes de visualisation de vidéos.

6.2.2 Un modèle encodeur-décodeur à base de Transformer

Nous nous appuyons sur l'architecture *Transformer* de Vaswani et al. (2017), qui constitue aujourd'hui l'état de l'art pour la traduction automatique comme pour de nombreuses autres tâches de traitement automatique du texte et de la parole. Nous avons ré-implémenté cette architecture en Python. Les hyperparamètres ont été choisis en partie par imitation de la littérature (l'implémentation originale du *Transformer* notamment), et en partie par expérimentation (essais avec d_{m} et d_{ff} plus petits, ou N nombre de couches plus grand, par exemple). Les variations de dimensionnement ont été testées sur un corpus de développement (test-1, voir Section 5.2.2) correspondant à 10 heures d'émissions (100 000 mots transcrits), échantillonnées aléatoirement dans nos données. La version qui est utilisée dans nos expérimentations utilise les paramètres suivants :

— dimension des représentations internes et des plongements lexicaux $d_{\text{m}} = 512$;
— dimension du perceptron multicouche $d_{\text{ff}} = 2048$;
— nombre de têtes d'attention $h = 8$;
— nombre de couches pour l'encodeur et le décodeur $N = 6$.

L'optimisation des paramètres du modèle est faite avec *Adam* (Kingma & Ba, 2015) en utilisant les paramètres suivants : $\beta_1 = 0,9$, $\beta_2 = 0,98$, $eps = 10^{-9}$. Nous avons également repris la méthode de variation du taux d'apprentissage proposée par Vaswani et al. (2017), en fixant le nombre d'étapes d'échauffement à 4000.

Afin de pouvoir traiter d'un vocabulaire ouvert pour le sous-titrage, les phrases (aussi bien en source qu'en cible) sont segmentées en unités sous-lexicales avec *SentencePiece* (Kudo & Richardson, 2018), en prenant un modèle *unigram* et un vocabulaire de 16 000 unités.

6.2.3 Contrôle de la segmentation : règles et contraintes

6.2.3.1 Les règles de segmentation pour les systèmes de base

Nous prenons comme architecture de référence un système qui utilise telle quelle la sortie de l'outil de reconnaissance vocale et qui calcule les balises de segmentation à l'aide d'un module à règles utilisant les heuristiques suivantes :

— une fin de phrase implique nécessairement un changement de bloc,

— chaque bloc contient au plus deux lignes,
— les lignes sont construites successivement en agrégeant progressivement les mots et les ponctuations,
— si la ligne en cours dépasse en longueur une borne inférieure et se termine par une virgule, cela déclenche la fin de ligne (ou le cas échéant la fin de bloc),
— si la ligne en cours s'apprête à dépasser en longueur une borne supérieure [1], cela déclenche la fin de ligne (ou le cas échéant la fin de bloc).

Un autre système de comparaison que nous avons testé réalise une simplification des transcriptions réalisée par un modèle *Transformer* simple, avant d'opérer la segmentation par règles.

6.2.3.2 Contrôler la verbosité du décodeur

Le cadre du sous-titrage imposant théoriquement des contraintes assez strictes en lien avec la taille du texte produit (les lignes ne doivent pas dépasser une certaine largeur, les sous-titres ne doivent pas être trop verbeux pour pouvoir être lus en synchronisation avec le son ou l'image), nous avons orienté une partie de nos expériences vers la compression avec contrôle explicite de la longueur, en utilisant les méthodes LRPE et LDPE (variations sur l'architecture *Transformer*), déjà présentées à la section 4.2.2. LRPE et LDPE permettent de fixer à l'inférence une valeur l correspondant à la longueur de phrase de sortie souhaitée. Dans nos expériences, nous avons modulé les objectifs de longueur afin de contraindre les modèles LRPE et LDPE à engendrer des phrases respectant soit un taux de compression constant CpR (auquel cas l est égale à la longueur de la phrase source multipliée par CpR), soit une fréquence d'affichage des caractères constante CPS (auquel cas l est égale à la durée allouée à l'affichage des tronçons de la phrase multipliée par CPS).

6.2.4 Méthodes d'adaptation au genre

Comme exposé en préambule, notre principal objectif dans cette partie est d'étudier l'apport de méthodes d'adaptation au domaine du système de sous-titrage. La question de l'adaptation au domaine est une question largement traitée en traduction automatique, que la visée soit d'adapter à un unique domaine cible (Chu & Wang, 2018; Saunders, 2021), ou bien de construire des systèmes multi-domaines (Pham et al., 2021). Nos travaux utilisent trois méthodes pour réaliser cette adaptation : l'utilisation d'étiquettes de domaine (Kobus et al., 2017), l'utilisation d'objectifs de longueur par domaine, et l'affinage des paramètres (Luong & Manning, 2015; Freitag & Al-Onaizan, 2016). Les autres méthodes présentes

1. L'intervalle d'acceptabilité de longueur est entre 13 et 32 caractères ; les bornes ont été choisies de manière à refléter la distribution de longueurs dans les véritables sous-titres.

TR	<stock> <jeu> Suite à votre passage dans l'émission comment les élèves à l'époque avait réagi.
ST	A votre passage dans l'émission, comment les élèves avaient réagi ? <p>

TABLE 6.1 – Exemple d'apprentissage constitué d'un segment transcrit automatiquement TR (source) et des sous-titres professionnels associés ST (cible), issu d'un extrait de l'émission *Tout le monde veut prendre sa place* du 03/11/2019. Les balises dans ST représentent la segmentation à l'affichage : `<br>` pour un saut de ligne au sein d'un bloc, `<p>` pour une fin de bloc (et changement d'écran). Les balises au début de TR indiquent la stratégie de sous-titrage et le genre de l'émission : ici `stock` et « jeu » (de telles balises ne sont présentes dans les exemples que pour les systèmes d'adaptation avec étiquettes).

dans la littérature reposent principalement soit sur des techniques d'apprentissage adverse qui neutralisent les différences entre genres au sein des représentations internes, soit sur l'affinage d'une sous-partie seulement du modèle, dont les paramètres sont adaptés à un domaine (Bapna & Firat, 2019).

6.2.4.1 Utilisation d'étiquettes de domaine

Cette méthode présente l'avantage d'être à la fois très simple, et relativement efficace dans de nombreuses situations expérimentales. Elle consiste à augmenter les représentations des segments sources par des informations relatives au domaine. Deux manières de procéder sont généralement considérées : soit insérer une balise de domaine en première position de chaque segment source ; soit injecter cette information plus directement dans les représentations de chaque unité source. Nous avons implémenté cette méthode en utilisant la première de ces deux approches et en distinguant, en plus des deux grands types d'émissions (`stock` et `direct`), les 8 domaines correspondant aux genres télévisuels (voir Section 5.4) : dessin animé, documentaire, fiction, jeu, journal, magazine, politique, et vulgarisation. Chaque segment source est donc préfixé par deux tokens spéciaux, qui permettent de spécialiser par domaine les représentations calculées par l'encodeur (voir Tableau 6.1).

6.2.4.2 Objectifs de longueur par domaine

Les modèles LRPE et LDPE attendent une consigne sur la longueur de la phrase à engendrer. Disposant de la longueur de la phrase initiale, ainsi que de la période d'affichage disponible (donnée par l'outil de transcription automatique), nous avons mis en place une modulation de l'objectif de longueur de manière à ce qu'il corresponde soit à un taux de compression CpR, soit à une fréquence d'affichage CPS (Section 6.2.3.2). En fixant des valeurs à suivre CpR et CPS spécifiques pour chaque domaine (choisies pour corres-

pondre aux valeurs observées en pratique pour les émissions du corpus), une information sur le flux de sortie attendu est fournie au système.

6.2.4.3 Affinage par genre

L'affinage consiste à pré-entraîner un système générique de sous-titrage avec un ensemble divers de segments parallèles, représentant un mélange de tous les genres télévisuels. Dans un second temps, l'apprentissage se poursuit en réduisant les données au seul genre d'intérêt. Les paramètres résultants conduisent souvent à de meilleures performances que les systèmes utilisant des balises, mais présentent l'inconvénient de conduire à l'apprentissage d'un modèle distinct pour chaque genre, au lieu d'un modèle unique capable de traiter tous les types d'émissions.

Dans nos expériences, nous avons utilisé les mêmes huit genres que pour les systèmes à base de balises, et implémenté l'affinage de la manière suivante : nous sauvegardons les paramètres obtenus à l'issue de l'apprentissage [2] d'un modèle *Transformer* classique et reprenons l'entraînement en restreignant les données au genre d'intérêt et en réduisant le taux d'apprentissage par un facteur 20. Nous appliquons les mêmes règles de convergence que pour le modèle de base.

6.3 Protocole et données expérimentales

6.3.1 Corpus

Nous avons utilisé pour nos expériences le corpus d'apprentissage décrit au chapitre 5, dans sa version la plus grande, correspondant au dernier point d'étape V4 (480K exemples d'apprentissage, voir Section 5.2.1.4). Le corpus test-1 (100K mots transcrits, voir Section 5.2.2) a servi d'ensemble de développement afin de valider le choix des hyperparamètres. Nous avions initialement effectué l'évaluation des systèmes sur les données du corpus test-2 (100K mots transcrits également), qui avait été réservé à cet effet. Cependant, les résultats obtenus sur test-2 étant semblables à ceux obtenus sur test-1, nous avons décidé de présenter dans la section 6.4 les scores calculés sur le corpus test, rassemblant test-1 et test-2, afin de leur donner plus de poids d'un point de vue statistique.

Les transcriptions automatiques figurant dans la partie source du corpus d'apprentissage et du corpus de test ont été engendrées selon le mode adapté du système VoxSigma, qui permet de guider le décodage avec un texte auxiliaire, dans notre cas le texte des sous-titres.

2. C'est-à-dire après convergence de l'apprentissage.

6.3.2 Méthodologies d'évaluation

6.3.2.1 Métriques automatiques

Pour l'évaluation de la segmentation des sous-titres, nous utilisons les métriques BLEU_{br} et TER_{br}. Pour évaluer la conformité aux normes superficielles pour l'affichage, nous utilisons les métriques CPL>36 et CPS>15 (ces métriques sont décrites plus en détail dans la Section 3.5). En cela, nous suivons les précédents de la littérature sur le sous-titrage automatique (Matusov et al., 2019; Karakanta et al., 2020b,a).

Concernant la qualité des phrases produites, nous appliquons des métriques standard pour la tâche de simplification de texte : BLEU (que nous notons BLEU_{nb} afin d'éviter l'ambiguïté) et SARI (voir Section 2.4).

Enfin, nous reprenons EAM et EA<10 % que nous avions déjà mis en place au chapitre 4 (Section 4.3.1.1) pour la mesure de la précision des systèmes reposant sur le contrôle de verbosité.

Pour les calculs BLEU et TER, nous utilisons l'implémentation *SacreBLEU* de Post (2018) [3] ; pour le calcul de SARI nous utilisons l'implémentation de la bibliothèque *EASSE* (Alva-Manchego et al., 2019).

6.3.2.2 Évaluations « métier »

Des évaluations « métier » ont été réalisées, conformément au processus de contrôle de qualité réalisé par les équipes de france.tv access, afin notamment de mesurer l'effet de certains aspects des systèmes de sous-titrage automatique : le volume de données d'apprentissage, l'utilisation de données complémentaires rétro-traduites, et l'adaptation au genre télévisuel via l'insertion d'étiquettes de domaine. Ces évaluations consistent à mesurer, pour un court extrait d'une émission, la qualité des sous-titres produits selon six critères, chacun noté entre 0 (le pire) et 5 (le mieux). Ces critères correspondent respectivement à l'évaluation de la forme, du contenu et des contraintes de lisibilités. Elles évaluent à quel point :

— l'orthographe des sous-titres est correcte,
— la ponctuation est respectée,
— le sens est correctement restitué,
— l'ensemble du contenu a été sous-titré,
— le sous-titrage est lisible (nombre de caractères à l'écran et vitesse de déroulement),
— la segmentation en blocs et lignes est correcte.

3. BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.14
TER+tok.tercom-nonorm-punct-noasian-uncased+version.1.4.14

Une note moyenne agrège ces différents critères.

Des expériences préliminaires menées en juin 2021 nous ont amenés à faire évoluer ce protocole de manière à

1. fournir également un décompte des différents types d'erreur, permettant de mesurer plus finement la qualité des systèmes selon cinq critères (recoupant en partie les précédents) :
   — fautes d'orthographes,
   — contresens,
   — mots manquants,
   — phrases manquantes,
   — erreurs de ponctuation ;
2. éliminer les émissions pour lesquelles les évaluations étaient systématiquement trop mauvaises (par exemple les émissions de jeux).

Pour la comparaison des évaluations nous calculons des taux d'erreur par type en rapportant les décomptes de fautes au nombre de blocs de sous-titres compris dans l'extrait considéré. Nous calculons également un taux d'erreur général (tous types confondus) pour l'extrait.

6.4 Résultats

6.4.1 Pré-traitements

Les données de test issues de la transcription automatique subissent quelques pré-traitements, permettant en particulier de prendre en charge les énoncés longs. En effet, les modèles à base de *Transformer* que nous avons implémentés sont limités à des séquences d'entrée de 150 unités au plus (après découpage par *SentencePiece*). Les phrases excédant cette longueur sont préalablement tronçonnées, puis ré-assemblées après traitement [4].

6.4.2 Comparaison aux systèmes de base

Comme indiqué à la section 6.2.3.1, la première approche mise en place pour insérer les balises `<br>` et `<p>` dans les sous-titres consiste à utiliser un module indépendant de segmentation par règles. Cette méthode a été testée d'une part avec la sortie d'un modèle de simplification *Transformer* (appris sur des données pour lesquelles les balises `<br>` et `<p>` avaient été filtrées), et d'autre part directement avec les segments transcrits automatiquement (résultant en un système qui se contente de segmenter les transcriptions automatiques). Le tableau 6.4 montre que l'ajout de l'étape de simplification entraîne un

4. Les virgules présentes dans transcription automatique sont privilégiées en tant que bornes pour les tronçons.

Systèmes	EAM	EA<10 %
+ BS + $\text{LRPE}_{\text{CpR=0,75}}$	17,2 %	13,4 %
+ BS + $\text{LRPE}_{\text{CPS=14,5}}$	21,4 %	25,2 %
+ BS + $\text{LDPE}_{\text{CpR=0,75}}$	18,1 %	12,0 %
+ BS + $\text{LDPE}_{\text{CPS=14,5}}$	21,9 %	24,3 %

TABLE 6.2 – Résultats de l'évaluation du contrôle de longueur des modèles LRPE et LDPE (moyennés sur le groupe d'émissions de test).

gain considérable pour toutes les métriques automatiques (le plus grand écart entre deux itérations de système pour les métriques BLEU_{nb} et SARI).

L'intégration des balises de segmentation dans le côté cible des données d'apprentissage ne change que très peu les scores BLEU_{nb} et SARI : la réalisation conjointe de la simplification et de la segmentation n'affecte pas la qualité de la simplification. Concernant l'apport pour la qualité de la segmentation, une amélioration peut être notée pour la métrique BLEU_{br}, ainsi que pour TER_{br} de façon moins sensible.

Contrairement aux modifications précédentes, l'ajout via LRPE ou LDPE de contraintes (non-adaptées au genre) sur la longueur des sous-titres produits ne permet pas, dans l'ensemble, d'améliorer les métriques automatiques. La précision du contrôle de longueur en elle-même est relative, puisque la différence entre la consigne de longueur et sa réalisation représente en moyenne entre 16 et 20 % de la longueur source (EAM) (voir Tableau 6.2) ; LRPE et LDPE sont ici comparables du point de vue de l'effectivité de ce contrôle [5]. Pour ce qui est de la qualité des phrases engendrées (mesurée par SARI, BLEU_{nb}, BLEU_{br}), LRPE est supérieur à LDPE [6], et la poursuite d'une fréquence de caractères constante semble préférable à l'application d'un unique taux de compression (ce qui paraît effectivement plus proche de ce que ferait un sous-titreur humain). Nous notons néanmoins un meilleur respect de la norme sur la fréquence d'affichage des caractères (CPS), en particulier lorsque l'objectif de longueur est modulé pour suivre une fréquence constante, cas dans lequel le score TER_{br} est aussi plus bas (ce qui indiquerait un meilleur positionnement des coupures dans les phrases).

6.4.3 Comparaison selon la stratégie de sous-titrage

Un des axes d'analyse que nous considérons est l'évaluation sur des émissions appartenant aux catégories `direct` ou `stock`, qui correspondent respectivement aux stratégies de sous-titrage en simultané et en différé. Les principaux résultats sont détaillés dans le

5. Une différence plus nette en faveur de LDPE avait été observée à la Section 4.5.1, cependant dans les expériences de ce chapitre, la longueur encodée est mesurée en unités sous-mots, résultant nécessairement en une précision moindre pour la longueur de sortie, mesurée elle en nombre de caractères.

6. Conformément aux résultats de la Section 4.5.1.

Systèmes	$BLEU_{br}$	$BLEU_{nb}$	SARI	TER_{br}	CPL>36	CPS>15
Évaluation sur les émissions de test de type `direct`						
+ BS	37,6	49,8	56,6	0,37	3,2	72,5
+ BS + RT	38,2	50,5	57,2	0,38	2,1	72,7
+ BS + BG	37,8	50,3	56,8	0,35	2,0	72,1
+ BS + RT + BG	39,0	51,5	57,9	0,34	1,7	69,7
+ BS + AF*	38,6	50,7	57,3	0,35	2,5	71,2
+ BS + RT + AF*	39,1	51,3	57,9	0,35	1,6	71,6
Évaluation sur les émissions de test de type `stock`						
+ BS	32,9	38,2	51,7	0,32	2,6	45,0
+ BS + RT	34,0	39,2	52,5	0,32	1,7	44,0
+ BS + BG	32,8	38,6	52,1	0,32	2,1	42,6
+ BS + RT + BG	33,3	38,5	52,5	0,31	1,7	41,0
+ BS + AF*	33,4	38,7	52,6	0,31	2,3	41,7
+ BS + RT + AF*	34,3	39,5	53,3	0,31	1,5	40,9

TABLE 6.3 – Résultats de l'évaluation de différents modèles sur les émissions de type `direct` ou `stock`. BS, RT et BG dénotent respectivement l'usage de balises de segmentation, de rétro-traduction, et de balises de genre. AF* indique que chaque émission a été traitée avec le modèle affiné sur le même genre (fiction, magazine, politique etc.)

tableau 6.3. Les variations entre systèmes sont similaires au sein de chaque évaluation : les méthodes d'adaptation au genre, par balisage ou affinage, obtiennent des scores relativement meilleurs. De même l'utilisation de données rétro-traduites est toujours bénéfique, sauf quand elle est combinée avec l'utilisation de balises de genre, pour l'évaluation sur les émissions `stock`. Cette différence est potentiellement liée à la distribution des programmes dans les données complémentaires. Comme le montre le tableau 5.7, la proportion d'émissions `stock` est plus faible dans le corpus rétro-traduit que dans le corpus d'apprentissage (35 % au lieu de 53 %). En outre nous avons introduit parmi les émissions `stock` un nouveau genre télévisuel (« enseignement », correspondant à des vidéos de cours de primaire, collège ou lycée), absent du corpus régulier, ce qui pourrait causer une inadéquation du système avec les exemples portant la balise <stock> à l'évaluation.

D'autres différences notables concernent la verbosité et la proximité par rapport à la référence : pour les sous-titres produits automatiquement pour les émissions `direct`, la norme sur le nombre de caractères par seconde est plus régulièrement dépassée (pour plus de 70 % des sous-titres), et la distance au texte de référence mesurée par $BLEU_{nb}$ et SARI est dans l'ensemble plus petite (ce qui se comprend dans la mesure où les conditions du direct contraignent les sous-titres de référence à être proches de la transcription dans les données d'apprentissage). Pour autant la qualité de segmentation représentée par TER_{br} paraît être meilleure pour l'évaluation sur `stock`.

Systèmes	BLEU_{br}	BLEU_{nb}	SARI	TER_{br}	CPL>36	CPS>15
Systèmes de base et références						
Transcription + règles	20,4	33,6	17,9	0,54	0,0	80,0
Transformer + règles	32,0	44,7	54,4	0,37	0,0	61,5
Référence + règles	70,6	100	100	0,13	0,0	31,0
Référence	100	100	100	0,0	0,0	44,4
Systèmes indifférents au genre (Transformer)						
+ BS	35,4	44,4	54,3	0,35	2,9	59,8
+ BS + RT	36,2	45,3	55,0	0,35	1,9	59,5
+ BS + $\text{LRPE}_{\text{CpR=0,75}}$	28,6	35,3	50,7	0,34	3,1	10,0
+ BS + $\text{LRPE}_{\text{CPS=14,5}}$	31,6	39,2	52,2	0,30	3,3	0,5
+ BS + $\text{LDPE}_{\text{CpR=0,75}}$	27,8	34,8	50,6	0,34	3,1	9,3
+ BS + $\text{LDPE}_{\text{CPS=14,5}}$	30,8	38,4	51,9	0,31	3,1	0,4
Systèmes adaptés au genre (Transformer)						
+ BS + BG	35,5	44,9	54,6	0,34	2,0	58,5
+ BS + RT + BG	36,4	45,5	55,4	0,33	1,7	56,4
+ BS + $\text{LRPE}_{\text{CpR}^*}$	29,8	36,8	51,3	0,32	3,2	11,1
+ BS + $\text{LRPE}_{\text{CPS}^*}$	32,1	40,1	52,6	0,30	3,2	7,7
+ BS + $\text{LRPE}_{\text{CPS}^*}$ + BG	33,3	41,6	53,2	0,30	2,2	12,6
+ BS + AF^*	36,2	45,2	55,1	0,33	2,4	57,6
+ BS + RT + AF^*	36,9	45,8	55,8	0,33	1,6	57,4

TABLE 6.4 – Résultats de l'évaluation de différents modèles (moyenne sur le groupe d'émissions de test). BS, RT et BG dénotent respectivement l'usage de balises de segmentation, de rétro-traduction, et de balises de genre. CpR^* et CPS^* indiquent que les consignes de longueur (selon CpR ou CPS respectivement) sont adaptées pour chaque domaine. AF^* indique que chaque émission a été traitée avec le modèle affiné sur le même domaine.

6.4.4 Effets de l'adaptation au genre télévisuel

Nous évaluons trois stratégies pour l'adaptation au genre, détaillées à la section 6.2.4 : l'introduction de balises de genre, l'introduction de consignes de longueurs spécifiques pour chaque type d'émission, enfin des stratégies d'affinage de systèmes. Les principaux résultats expérimentaux sont dans les tableaux 6.4 et 6.5. L'adaptation par balisage produit un effet positif modéré mais cohérent pour toutes les métriques (pour BLEU_{nb} environ 0,5 point en moyenne). À l'inverse, le contrôle des longueurs produit des résultats très dégradés par rapport à l'utilisation d'une unique valeur cible pour le taux de compression, le contrôle du CPS s'avérant toujours bien meilleur que le contrôle du CpR. En combinant les deux types de contrôle, on aboutit à un point de fonctionnement relativement équilibré sur l'ensemble des indicateurs, avec une perte en score BLEU_{nb}, mais un bien meilleur respect des contraintes de taille.

L'affinage des modèles produit des améliorations comparables, voire supérieures à celles obtenues avec des étiquettes de domaine, avec des variations en fonction des types de programme. Cette observation est cohérente avec les résultats de Pham et al. (2021) qui

Systèmes	BLEU_{br}	BLEU_{nb}	SARI	TER_{br}	CPL>36	CPS>15
Évaluation sur les émissions de test du genre « fiction »						
+ BS	31,1	34,2	48,7	0,33	2,6	42,2
+ BS + RT	32,6	35,5	49,7	0,33	1,9	41,5
+ BS + BG	30,8	34,4	49,2	0,32	2,6	40,8
+ BS + RT + BG	31,0	34,1	49,5	0,31	1,9	37,5
+ BS + AF^{fic}	31,5	34,4	49,7	0,31	2,8	41,1
+ BS + RT + AF^{fic}	32,2	35,2	50,6	0,32	1,7	40,2
Évaluation sur les émissions de test du genre « politique »						
+ BS	33,0	43,7	54,4	0,38	3,0	73,3
+ BS + RT	33,7	44,6	55,0	0,39	2,1	73,5
+ BS + BG	33,8	44,8	54,6	0,35	2,2	70,7
+ BS + RT + BG	35,5	46,1	55,8	0,35	2,4	67,3
+ BS + AF^{pol}	34,2	45,0	54,6	0,36	3,4	71,2
+ BS + RT + AF^{pol}	35,6	46,0	55,7	0,36	2,3	71,5
Évaluation sur les émissions de test du genre « magazine »						
+ BS	35,4	43,1	54,8	0,42	3,0	57,2
+ BS + RT	36,1	43,9	55,3	0,42	1,8	56,3
+ BS + BG	35,7	43,8	55,3	0,40	1,8	52,8
+ BS + RT + BG	36,6	44,3	55,8	0,39	1,5	51,4
+ BS + AF^{mag}	36,4	44,2	55,8	0,39	2,1	50,5
+ BS + RT + AF^{mag}	37,0	44,7	56,2	0,39	1,5	50,2
Évaluation sur les émissions de test du genre « journal »						
+ BS	44,8	60,8	60,3	0,24	3,3	74,5
+ BS + RT	45,6	61,8	61,3	0,24	2,0	74,7
+ BS + BG	44,1	60,6	60,3	0,25	2,0	79,4
+ BS + RT + BG	45,2	61,8	61,5	0,25	1,3	78,2
+ BS + AF^{jour}	45,3	61,1	61,0	0,24	1,9	77,8
+ BS + RT + AF^{jour}	45,8	61,9	61,9	0,24	1,3	77,8
Évaluation sur les émissions de test du genre « jeu »						
+ BS	24,5	28,2	48,6	0,36	2,2	36,9
+ BS + RT	23,9	27,6	48,1	0,36	1,1	36,1
+ BS + BG	24,5	28,4	49,0	0,35	1,0	32,4
+ BS + RT + BG	24,9	28,5	49,4	0,34	1,3	31,4
+ BS + AF^{jeu}	25,0	28,9	49,7	0,33	1,8	32,4
+ BS + RT + AF^{jeu}	24,8	28,2	49,0	0,35	1,1	33,3

TABLE 6.5 – Résultats détaillés de l'évaluation de modèles d'adaptation au genre. BS, RT, BG et AF dénotent respectivement l'usage de balises de segmentation, de rétro-traduction, de balises de genre et de l'affinage.

montrent que l'affinage, qui spécialise un système différent pour chaque genre télévisuel, est une stratégie difficile à surpasser lorsque l'on utilise un seul système multi-genre. Les résultats détaillés par genre télévisuel (Tableau 6.5) montrent des différences pouvant atteindre 1,3 points BLEU_{nb} pour les émissions politiques. Dans le cas des journaux (et au contraire des autres genres), la spécialisation du système augmente notablement la durée des sous-titres, ce qui est cohérent avec la verbosité observée par ailleurs pour ce genre (débit de parole élevé et faible niveau de compression).

Systèmes	BLEU_{br}	BLEU_{nb}	SARI	TER_{br}	CPL>36	CPS>15
Transf + BS + BG	35,5	44,9	54,6	0,34	2,0	58,5
Transf + BS + BG†	34,9	44,6	54,1	0,36	1,7	62,0
Transf + BS + BG‡	34,3	43,5	53,5	0,36	2,0	61,6
Transf + BS + BG†‡	34,0	43,7	53,4	0,37	1,7	61,7

TABLE 6.6 – Résultats de l'évaluation de modèles adaptés utilisant des étiquettes de domaine erronées (moyenne sur le groupe d'émissions de test). BS, et BG dénotent respectivement l'usage de balises de segmentation et d'étiquettes de domaine (stratégie de sous-titrage et genre télévisuel). Les modèles marqués par (†) ont été évalués sur des exemples pour lesquels l'étiquette indiquant la stratégie de sous-titrage était erronée ; les modèles marqués par (‡) ont été évalués sur des exemples pour lesquels l'étiquette indiquant le genre télévisuel était erronée (l'absence de ces symboles correspond au cas où les vraies étiquettes étaient en usage).

Le tableau 6.5 permet également de voir l'incidence de la probabilité a priori des genres, pour le modèle générique (Transf + BS) comme pour les modèles adaptés. Les genres les plus représentés ont tendance à avoir de bons scores (« journal » : 61 – 62 BLEU, 60 – 62 SARI ; « magazine » : 43 – 45 BLEU, 55 – 56 SARI) ; mais il n'y a clairement pas linéarité (ou même monotonie) pour la relation entre scores et représentation dans le corpus : « politique » ne compte que pour 4 % du corpus, mais est comparable à « magazine » (46 %) pour les résultats, tandis que « jeu » qui correspond à 16 % du corpus donne les pires performances (28 – 29 BLEU, 49 – 50 SARI).

Afin de vérifier la dépendance des modèles utilisant des étiquettes de domaine vis-à-vis de l'information fournie par les étiquettes, nous avons réalisé l'évaluation du système Transf + BS + BG en préfixant les segments sources par des balises indiquant de manière inexacte la stratégie de sous-titrage ou le genre télévisuel [7] (voir Tableau 6.6). Le remplacement des balises produit une dégradation pour l'ensemble des métriques, qui est plus marquée pour le cas des balises de genre télévisuel ($-1,4$ BLEU_{nb}, $-1,1$ SARI) : c'est un résultat logique, dans la mesure où le genre télévisuel est une information plus précise (huit catégories dans le corpus principal, à comparer aux deux stratégies de sous-titrage), qui met en relief des disparités plus importantes (voir Sections 5.3 et 5.4).

À titre indicatif, nous avons aussi évalué certains de nos systèmes sur des émissions de genres non-vus à l'apprentissage : pour des leçons de MOOC (44 minutes) le système générique (Transf + BS) le score BLEU est de $45,8$, et monte à $54,5$ avec l'ajout de l'étiquette « vulgarisation » (représentant $0,9$ % du corpus) ; pour une présentation TEDx (13 minutes) le système générique obtient BLEU $= 63,8$, mais descend à $62,2$ avec l'étiquette « vulgarisation ». Ces résultats témoignent d'une certaine robustesse des modèles appris,

7. Concernant le genre télévisuel, la balise est choisie aléatoirement parmi les possibilités incorrectes ; pour la balise de stratégie de sous-titrage, il n'y a qu'une possibilité.

TR	**Suite à** votre passage dans **l'émission** comment les élèves à l'époque **avait réagi.**
Transf + BS	Suite à votre passage dans **l'émission,** comment les élèves <p> **avaient réagi ?** <p>
+ $LRPE_{CPS=14,5}$	Suite à votre passage dans **l'émission.** <p> Comment ? <p>
ST	**A** votre passage dans **l'émission,** comment les élèves **avaient réagi ?** <p>
TR	**Donc nous** on monte les mâts **oui mais bon, on va laisser finir**.
Transf + BS	On monte les mâts. <p>
+ $LRPE_{CPS=14,5}$	On monte les mâts. <p> On va laisser finir. <p>
ST	**Nous,** on monte les mâts. <p> **Ah oui.** <p> **Stop.** On va **les** laisser finir. <p>
TR	En Alabama, les anti-avortement sont **majoritaires, 70 pourcent** des habitants se déclarent pour **une interdiction de l'IVG**.
Transf + BS	En Alabama, les anti-avortement sont **majoritaires.** <p> **70 %** des habitants se déclarent pour **une interdiction de l'IVG**. <p>
+ $LRPE_{CPS=14,5}$	En Alabama, les anti-avortement sont **majoritaires.** <p> **70 %** des habitants se déclarent pour **une IVG**. <p>
ST	En Alabama, les anti-avortement sont **majoritaires.** <p> **70%** des habitants se déclarent pour **une interdiction de l'IVG**. <p>

TABLE 6.7 – Exemples de phrases engendrées par les modèles Transf + BS et Transf + BS + $LRPE_{CPS=14,5}$, comparées à la transcription initiale (TR) et aux sous-titres de référence (ST).

notamment pour l'adaptation à des domaines proches.

Une dernière observation est que l'amélioration des performances obtenues par adaptation au genre reste dans tous les cas modeste. Ce résultat soulève la question de l'homogénéité réelle des émissions regroupées dans ces grandes catégories, qui, bien que relevant du même genre télévisuel, diffèrent sous de multiples autres aspects (contenus, thèmes abordés, intervenants, etc).

Le tableau 6.7 présente des exemples de transformations réalisées par les systèmes Transf + BS et Transf + BS + $LRPE_{CPS=14,5}$. Nous pouvons constater que les modèles ont appris à re-segmenter les phrases (après « émission » dans le premier exemple, « mâts » dans le deuxième, et « majoritaires » dans le troisième), et à reconnaître la forme interrogative (premier exemple). Occasionnellement, ils peuvent également corriger des erreurs de grammaire (« avait », premier exemple). Les conventions orthographiques et typographiques, telles que l'écriture de « % » (troisième exemple) sont globalement gérées efficacement. Le deuxième exemple montre que le modèle Transf + BS élague trop la phrase initiale par rapport à la référence. LRPE fait mieux dans ce cas, bénéficiant de l'information de longueur en rapport avec le temps d'affichage disponible. Toutefois, la contrainte de longueur induit souvent des suppressions abruptes, provoquant la perte du sens ori-

Transcription automatique Présenter avec les mains, le truc, vous en tant que metteur en scène, on n'est pas je ne sais pas je aime pas généralement je me méfie beaucoup d'élus. **Transcription idéale** Présenter avec les mains, le truc, vous en tant que metteur en scène, on n'est pas... Je ne sais pas, j'aime pas, généralement je me méfie beaucoup de l'illu[stration avec les mains.]
LRPE (CPS = 10) Vous en tant que metteur en scène,   je ne sais pas. **LRPE (CPS = 14)** Vous en tant que metteur en scène,   je ne sais pas. <p> Je me méfie. **LRPE (CPS = 18)** Vous en parlez tant que metteur   en scène, je ne sais pas. <p> Je me méfie beaucoup d'élus. <p>
BG = jeu, stock Présentez avec les mains. <p> En tant que metteur en scène,   on n'est pas... <p> Je me méfie beaucoup d'élus. <p> **BG = magazine, direct** Présenter avec les mains, le truc... <p> En tant que metteur en scène,   on n'est pas... <p> Je me méfie beaucoup d'élus. <p> **BG = journal, stock** Présenter avec les mains, le truc,   en tant que metteur en scène, <p> je ne sais pas. <p> Je me méfie beaucoup d'élus. <p>

TABLE 6.8 – Exemples de phrases engendrées par les modèles Transf + BS + LRPE_{CPS} et Transf + BS + BG, à partir de la même phrase transcrite automatiquement, en variant la consigne de longueur (pour LRPE, de façon à correspondre à une certaine fréquence d'affichage CPS), ou les étiquettes de domaine (stratégie de sous-titrage, et genre télévisuel).

ginal (troisième exemple) ou de la cohérence syntaxique. Outre l'abandon d'éléments importants, les erreurs fréquemment commises par les modèles comptent de mauvaises segmentations, et la conservation d'artefacts de la transcription automatique. En alignant les phrases produites par Transf + BS avec les références de notre ensemble de test, nous avons noté que les opérations d'édition les plus fréquentes étaient les suppressions de mots connecteurs ou introductifs : « et », « que », « c'est », « ça », « donc », « qui »...

Les exemples du tableau 6.8 permettent d'apprécier les effets du contrôle de longueur ou de genre, en décodant la même phrase avec des consignes (fréquence d'affichage visée ou étiquettes de domaine) différentes. Nous notons ainsi l'allongement de la sortie à mesure que la contrainte sur CPS est relâchée ; dans le cas CPS $= 18$, l'ajout « en parlez » ne figure même pas dans la transcription initiale. Nous pouvons aussi remarquer que l'exemple produit avec la balise `direct` ressemble davantage à la version orale, intégrant à deux reprises des points de suspension ; ce qui est cohérent puisque les sous-titres des émissions en direct sont en pratique plus proches de la parole, du fait des nécessités techniques (voir Section 5.3). En comparaison, l'exemple engendré avec les balises

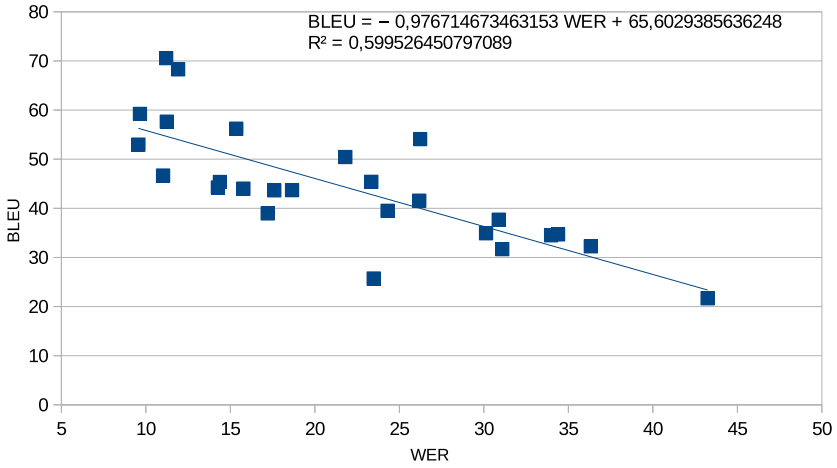


FIGURE 6.3 – Score BLEU_{nb} du système Transf + BS en fonction du WER de la transcription automatique (aidée), pour les émissions du corpus de test ; avec ajustement affine par régression linéaire.

« journal » et `stock` relève plus d'un style rédigé, typique de ce qui peut être rencontré dans les informations télévisées.

6.4.5 Utilité de la rétro-traduction

Les motivations initiales pour utiliser des données rétro-traduites en complément des données alignées étaient (a) d'améliorer la qualité des plongements lexicaux ; (b) d'apprendre à mieux distinguer les styles de sous-titres avec davantage d'exemples ; (c) de renforcer la génération de la segmentation, dans la mesure où le système aurait à sa disposition un plus grand ensemble de sous-titres de référence correctement segmentés.

Dans le tableau 6.4, nous pouvons voir que l'ajout de données rétro-traduites par rapport à un système *Transformer* de base apporte un gain en BLEU_{nb} ($0,9$ point) et en SARI ($0,7$ point). Il faut observer que l'utilisation de ces données complémentaires reste toujours bénéfique lorsqu'elle est combinée avec les méthodes d'adaptation au genre par balisage ou affinage : un gain de $0,6$ point BLEU_{nb} dans les deux cas, ce qui confirmerait une meilleure adaptation au genre.

Une autre tendance régulière qui se dégage est le respect plus strict de la contrainte sur le nombre de caractères par ligne (CPL>36) avec l'utilisation des données rétro-traduites, la quantité d'exemples de référence semblant bien importer pour cet aspect.

6.4.6 Influence de la qualité des transcriptions

Comme évoqué à la section 5.2.1.2, la qualité de la transcription automatique est soumise à des disparités selon les émissions, étant notamment affectée par des facteurs tels que la vitesse et la clarté de l'élocution, le recouvrement de paroles, et généralement la

Systèmes	test transcrit sans aide ST		test transcrit avec aide ST		test transcrit manuellement	
	$BLEU_{nb}$	**SARI**	**$BLEU_{nb}$**	**SARI**	**$BLEU_{nb}$**	**SARI**
Système appris avec des transcriptions non-aidées						
Transf + BS	42,0	54,2	44,1	54,4	46,2	55,9
Système appris avec des transcriptions aidées						
Transf + BS	41,4	53,1	44,4	54,3	46,5	55,8

TABLE 6.9 – Résultats d'évaluation comparés selon la qualité de la transcription (moyenne sur le groupe d'émissions de test). Le type de modèle évalué ici correspond à un *Transformer*, dont les données d'apprentissage comprennent des balises de segmentation (BS).

présence de bruit (une musique de fond ou des rires par exemple). Et l'évaluation de nos systèmes nous permet d'observer une dépendance assez claire entre la qualité de la transcription automatique et la qualité des phrases sous-titres produites : nous avons calculé un coefficient de corrélation linéaire de $-0,77$ entre le WER des émissions de test et les scores $BLEU_{nb}$ obtenus par le système Transf + BS ; de même pour SARI nous avons calculé un coefficient de $-0,74$ (les valeurs sont négatives puisqu'il s'agit d'une comparaison entre des scores et des taux d'erreur). La figure 6.3 illustre graphiquement la relation entre $BLEU_{nb}$ et WER.

Nous avons également analysé l'influence de la qualité générale de la transcription dans le corpus, et non plus seulement émission par émission. Pour cela, nous avons exploité le mode de transcription sans texte auxiliaire de l'outil VoxSigma [8], et la transcription manuelle réalisée pour les émissions de test. Sans l'appui apporté par le texte extrait des sous-titres, la transcription automatique s'éloigne de la référence humaine, comme en atteste le taux d'erreur de mots : de $21,7$ % avec l'aide des sous-titres, il passe à $23,9$ % sans l'aide des sous-titres. Nous avons alors entraîné une autre version du système Transf + BS (*Transformer*, pour lequel les balises de segmentation sont intégrées au flux textuel), en utilisant des transcriptions non-aidées pour les exemples d'apprentissage. Le tableau 6.9 montre les résultats d'évaluation des deux versions du système Transf + BS, sur trois version du corpus de test qui diffèrent par leurs sources : test transcrit automatiquement sans aide des sous-titres, test transcrit automatiquement avec aide des sous-titres, et transcription manuelle de référence (considérée comme une version parfaite des deux autres, c.-à-d.WER $= 0$ %).

Il apparaît d'une part que, pour les deux systèmes, les scores de qualité de phrase augmentent en accord avec la qualité de la transcription du test (l'écart maximum étant de 4 – 5 points en $BLEU_{nb}$, et de 2 – 3 points en SARI). Toutefois le gain obtenu en effec-

8. Dans le reste des expériences, la transcription automatique est toujours aidée par le texte des sous-titres (Section 5.2.1.2).

Systèmes	BLEU_{br}	BLEU_{nb}	SARI	TER_{br}	CPL>36	CPS>15
Systèmes appris sur le corpus V1 (11/2020)						
+ BS	34,3	43,2	52,7	0,33	5,2	51,0
+ BS + BG	33,8	42,9	52,0	0,34	4,8	57,5
Systèmes appris sur le corpus V2 (02/2021)						
+ BS	34,5	43,8	52,8	0,35	5,9	59,1
+ BS + RT	35,0	44,0	53,0	0,35	2,6	60,0
+ BS + BG	34,9	44,1	53,0	0,33	7,2	55,4
+ BS + RT + BG	35,8	44,6	53,3	0,33	2,9	56,7
Systèmes appris sur le corpus V3 (07/2021)						
+ BS	35,1	44,4	53,8	0,34	2,3	56,4
+ BS + RT	35,4	44,5	53,8	0,36	2,2	61,3
+ BS + BG	35,5	44,5	54,0	0,33	4,3	56,5
+ BS + RT + BG	36,4	45,2	55,0	0,32	2,5	55,8
Systèmes appris sur le corpus V4 (11/2021)						
+ BS	35,4	44,4	54,3	0,35	2,9	59,8
+ BS + RT	36,2	45,3	55,0	0,35	1,9	59,5
+ BS + BG	35,5	44,9	54,6	0,34	2,0	58,5
+ BS + RT + BG	36,4	45,5	55,4	0,33	1,7	56,4

TABLE 6.10 – Résultats de l'évaluation de modèles appris aux points d'étape du corpus (moyenne sur le groupe d'émissions de test). BS, RT et BG dénotent respectivement l'usage de balises de segmentation, de rétro-traduction, et de balises de genre.

tuant la transduction à partir de la transcription humaine de référence reste relativement modéré, avec une moyenne BLEU_{nb} plafonnant à $46,5$ (seulement $0,7$ point de plus que la meilleure méthode adaptée du Tableau 6.4). La figure 6.3 montre que les émissions de test aux WER les plus bas (~ 10 %) atteignent des scores BLEU_{nb} entre 50 et 70. Il peut en être déduit que la qualité de la transcription à elle seule ne garantit pas la performance des modèles de sous-titrages ; en pratique les sources d'erreur pour la reconnaissance de parole vont de pair avec d'autres caractéristiques rendant difficile la transformation vers les sous-titres (la parole non-préparée aura tendance par exemple à être mal reconnue, en ayant en plus une syntaxe éloignée de l'écrit).

L'autre constat que nous pouvons faire est que le système appris avec des transcriptions non-aidées obtient des résultats comparables à ceux du système appris avec des transcription aidées, voire significativement meilleurs pour l'évaluation sur le test transcrit sans-aide. La mesure selon SARI plus particulièrement semble bénéficier de la transcription sans aide : le système pourrait avoir pris l'habitude d'être moins conservateur au niveau de la simplification, à cause du bruit amplifié dans la source ; ce qui serait récompensé par cette métrique.

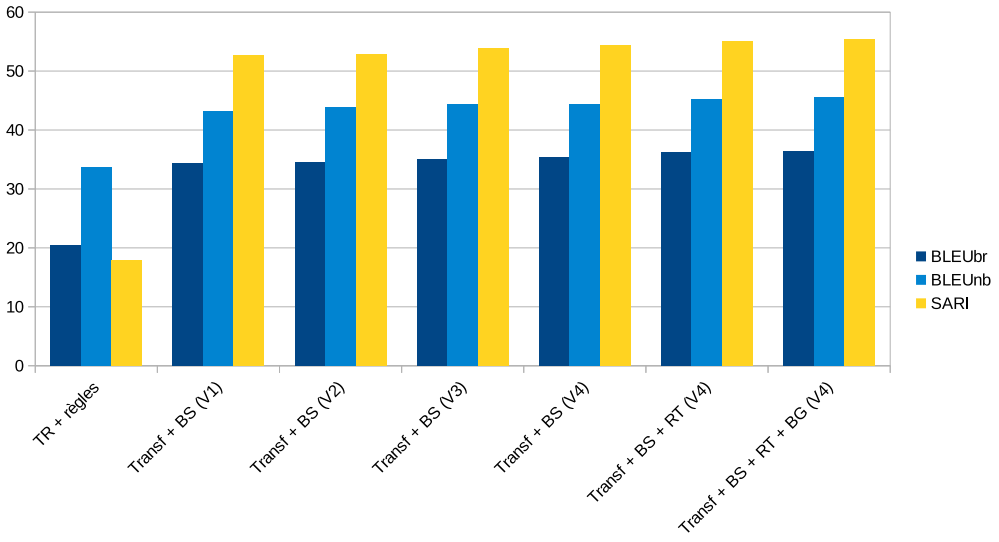


FIGURE 6.4 – Progression des scores BLEU_{br}, BLEU_{nb} et SARI (moyenne sur le groupe d'émissions de test) en fonction de la méthode et de la version du corpus d'apprentissage utilisées.

6.4.7 Rendement des exemples d'apprentissage

L'élaboration du corpus d'apprentissage s'étant étalée dans le temps, nous avons effectué l'évaluation des systèmes de sous-titrage à plusieurs points d'étape (V1 à V4), correspondant aux incréments suivants [9] :

— 60K segments supplémentaires pour V2 (29 % de la taille de V1),
— 112K segments supplémentaires pour V3 (42 % de la taille de V2),
— 105K segments supplémentaires pour V4 (28 % de la taille de V3).

Le tableau 6.10 présente les résultats d'évaluation aux quatre points d'étape, pour des systèmes *Transformer* utilisant éventuellement des données rétro-traduites (RT) et des étiquettes de domaine (BG) ; la figure 6.4 montre la progression générale des modèles évalués selon le volume du corpus d'apprentissage et les principales méthodes testées. Un accroissement modéré peut être noté pour les métriques de qualité des phrases : en moyenne (pour les systèmes comparables), $+0,9$ BLEU_{nb} et $+0,6$ SARI entre V1 et V2, $+0,5$ BLEU_{nb} et $+1,1$ SARI entre V2 et V3, $+0,4$ BLEU_{nb} et $+0,7$ SARI entre V3 et V4. Les gains pour SARI s'accordent ici davantage avec l'apport d'exemples d'apprentissage (plus important entre V2 et V3) que les gains pour BLEU_{nb}. Le respect de la contrainte sur la longueur des lignes bénéficie aussi de l'augmentation du volume du corpus d'apprentissage ; effet qui est accentué par l'ajout de données rétro-traduites. Il est également remarquable que l'utilisation de balises indiquant le genre télévisuel est

9. Calculés en tenant compte du filtrage selon la qualité d'alignement des exemples d'apprentissage (Section 5.2.1.3).

moins bénéfique avec les versions antérieures du corpus (étant même négative dans le cas de V1) ; il semblerait que cette approche d'adaptation ne soit pas optimale pour des ensembles d'apprentissage trop petits.

6.4.8 Évaluation « métier » de la qualité des sous-titres

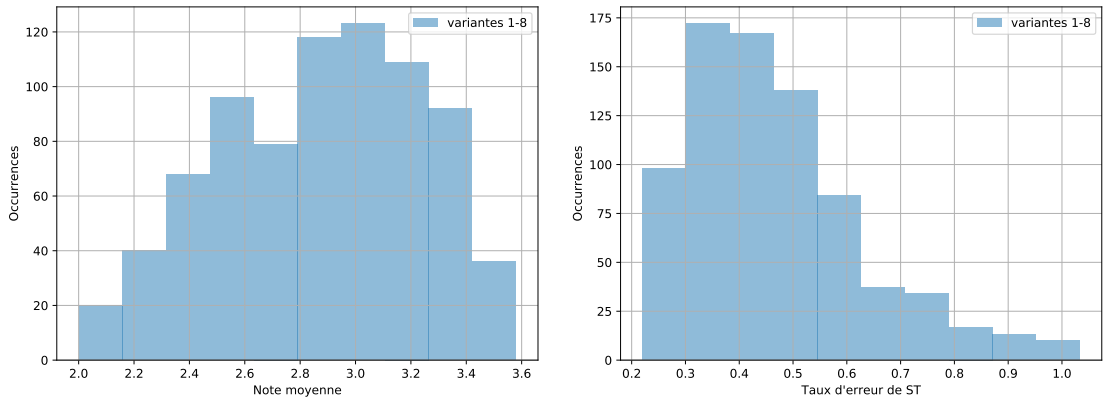
L'évaluation métier a été conduite entre juillet et octobre 2021. Elle a permis d'évaluer huit systèmes de sous-titrage, correspondant à des contrastes portant :

1. sur le nombre de données d'apprentissage du système : les variantes 1–4 utilisent le jeu d'apprentissage V2 comportant 411K segments parallèles (voir le Tableau 5.4), alors que les variantes 5–8 utilisent le jeu d'apprentissage V3 (618K segments parallèles) ;
2. sur l'utilisation de données rétro-traduites (voir la Section 5.2.1.5), les variantes 3, 4, 7, 8 ayant été apprises avec ces données en supplément des segments parallèles, et les variantes 1, 2, 5 et 6 sans ;
3. sur l'utilisation d'étiquettes portant sur le genre de l'émission, les variantes 2, 4, 6, 8 suivant cette méthode, contrairement aux variantes 1, 3, 5, 7.

Chacun des huit systèmes a été utilisé pour produire les sous-titres d'un ensemble aléatoire d'environ 100 émissions, en contrôlant que la proportion des types d'émissions était similaire pour chacun des systèmes. Un extrait aléatoire de chaque émission correspondant à une durée d'environ 5 minutes a ensuite été annoté selon la grille décrite à la section 6.3.2.2. Les principaux résultats peuvent être visualisés grâce aux figures 6.5 et 6.6.

La figure 6.5 montre la distribution des notes moyennes sur 5, ainsi que la distribution des taux d'erreur de sous-titres (nombre d'erreurs rapporté au nombre de blocs), pour l'ensemble des instances traitées par l'évaluation métier. La valeur médiane des notes moyennes est de $2,92$ (sachant que selon la grille d'évaluation utilisée par les annotateurs, les sous-titres acceptables pour diffusion seraient entre 4 et 5) ; tandis que le taux d'erreur médian vaut $0,43$ (soit moins d'une erreur tous les deux blocs).

Les figures 6.6a et 6.6c comparent les indicateurs statistiques des notes moyennes et des taux d'erreur pour les trois contrastes étudiés. Les résultats n'étant pas assez clairs pour tirer de conclusion a priori, nous avons réalisé une analyse de variance multifactorielle (*n-way ANOVA*, en prenant un intervalle de confiance de 95 %) de la note moyenne et du taux d'erreur de sous-titres, en considérant comme facteurs le jeu d'apprentissage (V2 ou V3), l'utilisation de données rétro-traduites (RT), l'utilisation de balises de genre (BG), et le titre des émissions (p. ex. *Journal 20h00*). Dans le cas de la note moyenne, l'interaction entre certains facteurs s'est révélée significative, rendant l'interprétation difficile. En revanche dans le cas du taux d'erreur, l'analyse a abouti à un modèle sans terme d'interac-



(a) (b)

FIGURE 6.5 – Distribution de la note moyenne sur 5 (a) et du taux d'erreur de sous-titre (b) pour l'ensemble des productions évaluées manuellement.

tion, pour lequel la version du jeu d'apprentissage et le titre d'émission sont significatifs. Les figures 6.6b et 6.6d montrent l'influence des émissions (groupées par genre télévisuel) sur la note moyenne et le taux d'erreur respectivement.

En complément, nous avons également mené une analyse de variance multifactorielle (avec les mêmes facteurs et le même intervalle de confiance) pour les six notes et les cinq taux d'erreur subsidiaires décrits à la section 6.3.2.2. La majorité de ces analyses ont fait émerger des modèles pour lesquels soit l'interaction entre facteurs prévalait, soit les facteurs d'intérêt n'étaient pas significatifs. Néanmoins, la version de corpus apparaît significative pour l'analyse du taux de contresens et du taux de faute d'orthographe, tandis que l'usage d'étiquettes de domaine est significatif pour l'analyse de la note de contresens. Les figures 6.6e et 6.6f présentent respectivement les indicateurs de position pour les notes de contresens et les taux de fautes d'orthographe observés.

En parallèle de l'évaluation métier, nous avons réalisé une évaluation automatique des mêmes sous-titres produits, en reprenant les métriques employées dans les autres expériences [10] (voir Section 6.3.2.1). Les coefficients de corrélation linéaire entre mesures manuelles et automatiques ont donc pu être calculés, et sont présentés dans la matrice de la figure 6.7a : les cases plus claires correspondent à des coefficients plus élevés, soit à une corrélation plus forte. Nous observons notamment une relation entre les métriques automatiques BLEU_{nb}, SARI et $1 - \text{TER}_{\text{br}}$. Le taux de compression CpR est lui lié aux notes sur la préservation du sens (`contresens`) et la conservation des mots importants (`mots_manquants`), ainsi qu'au score BLEU_{nb} : un taux de compression plus élevé correspondant à une compression plus faible (pour CpR = 1 il n'y a pas de com-

10. À l'exception de CPL>36 et CPS>15, les scores automatiques ont été calculés seulement sur les extraits annotés, et non pas sur les émissions entières.

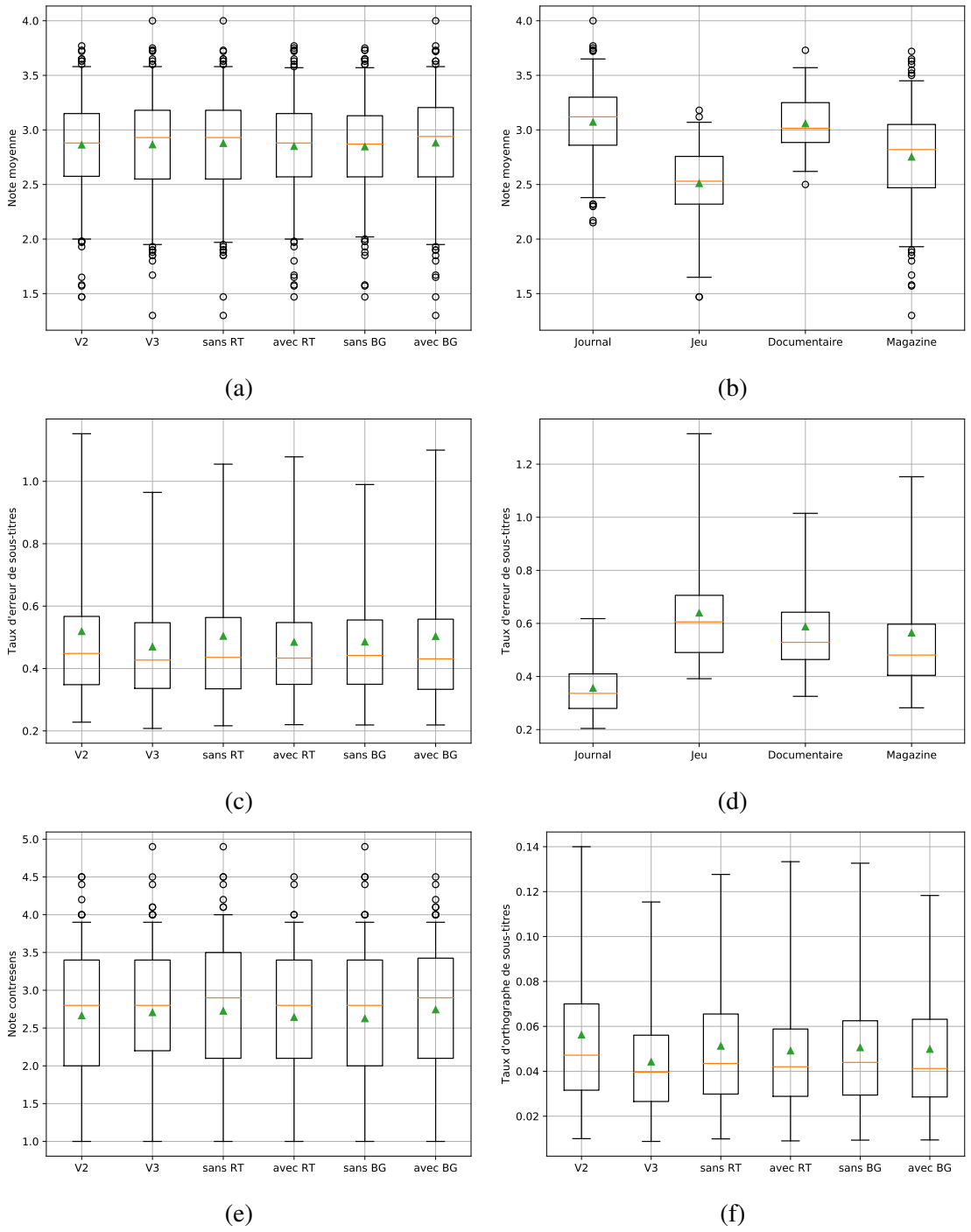
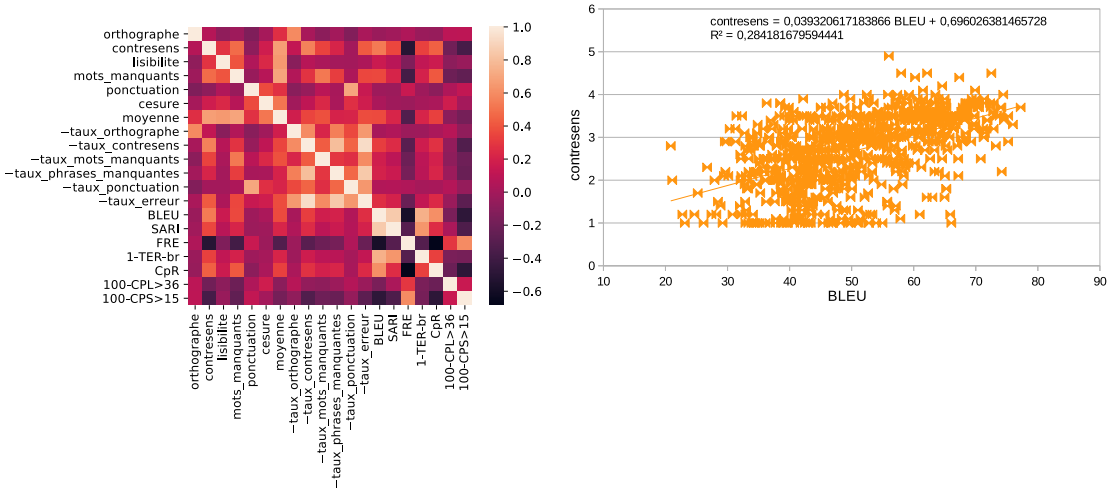


FIGURE 6.6 – Profils statistiques de mesures issues de l'évaluation métier. Plus précisément, les diagrammes présentent (a,b) les notes moyennes, (c,d) les taux d'erreur, (e) les notes de contresens, et (f) les taux de faute d'orthographe. Les séries rapportées sont définies soit par (a,c,e,f) le jeu d'apprentissage (V2 ou V3), l'utilisation de données rétro-traduites (RT), l'utilisation de balises de genre (BG), soit par (b,d) le genre télévisuel. Les pattes des diagrammes représentent les 2^{e} et 98^{e} centiles.



(a) (b)

FIGURE 6.7 – Corrélation entre l'évaluation métier et les métriques automatiques : (a) matrice des coefficients de corrélation de Pearson, (b) tracé des notes sur le critère de préservation du sens en fonction du score BLEU_{nb} (avec ajustement affine). Les métriques évaluant un niveau d'erreur ont été passées en négatif ou en complémentaire pour pouvoir être mises en relation avec les métriques évaluant un niveau de correction.

pression), le nombre de suppressions affectant des éléments importants de la phrase doit mécaniquement diminuer avec l'augmentation de CpR. Les métriques $100 - \text{CPL} > 36$, $100 - \text{CPS} > 15$ et FRE semblent aller à l'encontre des autres mesures sur la qualité (les lignes/colonnes en question apparaissant plus sombres), ce qui met en lumière l'opposition entre d'une part la lisibilité (exprimée par FRE et les contraintes sur CPL et CPS) et la qualité générale des phrases de sous-titres produites. La figure 6.7b montre le tracé des notes sur la préservation du sens en fonction du score BLEU_{nb} : plus que les autres mesures humaines, la note `contresens` semble être liée à la métrique BLEU_{nb} (le coefficient de corrélation linéaire entre les deux est de $0,53$).

6.5 Conclusion

Dans ce chapitre, nous avons présenté les travaux réalisés pour mettre en place un système entièrement automatisé de génération de sous-titres pour des émissions télévisuelles en langue française. Ce système prend appui sur un grand corpus associant transcriptions automatiques et sous-titres de références pour des émissions variées (Chapitre 5), qui sert à l'apprentissage d'un modèle de traduction de type encodeur-décodeur exploitant l'architecture *Transformer*. L'ajout de balises de segmentation explicites dans les textes générés permet de réaliser le sous-titrage en deux étapes (en comptant la reconnaissance de parole), sans dégradation des performances par rapport à la mise en cascade d'un module de segmentation distinct.

Partant du constat que les genres télévisuels qui composent le corpus d'apprentissage présentent de fortes disparités quant à leurs sous-titres, nous nous sommes particulièrement focalisés sur l'étude de méthodes d'adaptation des modèles aux types de sous-titres et aux genres télévisuels. Nos expériences confirment l'apport des méthodes classiques d'adaptation, telles que l'utilisation d'étiquettes de genre et l'affinage, notamment quand elles sont combinées avec une technique d'augmentation de données ; les méthodes fondées sur le contrôle de longueur se sont en revanche montrées peu performantes dans l'ensemble. Nous observons une corrélation importante entre le taux d'erreur de mots de la transcription et la qualité des phrases engendrées. Néanmoins nos essais avec une transcription manuelle, considérée idéale, suggèrent que la performance de la reconnaissance de parole n'est probablement l'obstacle principal à la progression des systèmes de sous-titrage. Dans ce contexte particulier les améliorations délivrées restent modestes, variables selon les genres et les types de sous-titres. Ceci laisse à penser que la ventilation des données par genre télévisuel est loin de capturer toutes les sources de variation présentes dans les données et que des distinctions plus fines devraient être opérées pour tirer le meilleur parti des méthodes d'adaptation. L'évaluation par des experts métier montre que l'amélioration due à l'augmentation du volume de données et à l'adaptation par étiquettes de domaine est toutefois sensible, notamment dans la mesure de la préservation du sens original.

Dans nos travaux futurs, nous comptons poursuivre l'étude des méthodes d'adaptation en essayant d'exploiter au mieux la richesse de notre corpus d'apprentissage, pour lequel nous disposons de méta-données très riches (par exemple : le nom de l'émission, la date de télé-diffusion, l'identité des principaux intervenants). Ceci permet en particulier d'explorer la construction de modèles adaptés par émission, ou bien encore adaptés temporellement pour ce qui concerne en particulier les journaux. Nous avons aussi l'intention de poursuivre l'étude de l'adaptation multi-genre à travers d'autres techniques, notamment les modules d'adaptation (*adapter layers*) (Bapna & Firat, 2019). Une autre question importante concerne les évaluations réalisées, qui s'appuient ici uniquement sur des métriques automatiques reflétant soit la similarité avec des références, soit la conformité avec la charte du CSA : étudier également l'utilité des sous-titres automatiques du point de vue de leur utilisation par des sous-titreurs professionnels (en post-édition) ou des spectateurs est également une perspective importante.

Chapitre 7

Évaluation de la segmentation des sous-titres pour des systèmes bout en bout

7.1 Introduction

La production automatique de sous-titres est une tâche complexe, qui va bien au-delà de la transcription automatique : en effet les sous-titres doivent non seulement refléter le contenu des propos, mais aussi satisfaire de multiples exigences formelles en rapport avec la position à l'écran, la longueur et la couleur du texte, la durée d'affichage et la synchronisation avec la parole, etc. (voir Section 3.2.1) En outre, une bonne segmentation du texte (qu'il soit dans la langue d'origine ou traduit) doit répondre à des contraintes syntactiques et sémantiques, puisque le respect des unités linguistiques facilite la compréhension et mène à des sous-titres plus lisibles (Perego, 2008; Rajendran et al., 2013).

Dans ce chapitre [1], nous étudions les manières d'évaluer la qualité des segmentations produites par des systèmes de sous-titrage bout en bout, qui sont désormais de plus en plus fréquemment utilisés (Lakew et al., 2019; Liu et al., 2020). Contrairement aux systèmes en cascade, qui contiennent généralement un module de segmentation indépendant qui peut être testé en tant que composant autonome en mesurant simplement sa capacité à reproduire une segmentation de référence pour un texte de référence (*texte parfait*), les systèmes bout en bout engendrent directement un texte segmenté, qui peut ne pas correspondre à celui des sous-titres de référence (*texte imparfait*). Faire la part entre les erreurs relatives au texte et celles relatives à la segmentation devient alors difficile. Récemment, des métriques telles que BLEU_{br} et TER_{br} (Karakanta et al., 2020a), qui incluent les balises de segmentation dans le calcul de la qualité globale de la sortie, ont été proposées pour pallier ce problème. Cependant, leur capacité à évaluer précisément les erreurs de la segmentation automatique n'a jamais été étudiée. Nous réalisons ici un examen systéma-

1. Les travaux présentés dans ce chapitre ont été réalisés en collaboration avec Alina Karakanta, dans le cadre de l'article : Karakanta et al. (2022).

tique des métriques de segmentation de sous-titres, dans le but de mieux comprendre leur comportement par rapport aux erreurs textuelles.

Nos contributions pour ce chapitre sont les suivantes :

— Une comparaison des métriques de segmentation existantes, pour évaluer la qualité des sous-titres dans la situation idéale d'un contenu textuel *parfait*, correspondant à celui de la référence (Section 7.3.1).

— Un nouveau score *Sigma*, dérivant d'une estimation de majorant de BLEU_{br}, qui isole l'information liée à la segmentation indépendamment de la qualité du texte potentiellement imparfait (Section 7.3.2).

— Une méthode de projection de frontière qui rattache les points de coupure de l'hypothèse vers la référence, et permet d'appliquer les mesures de segmentation classiques y compris dans le cas de textes *imparfaits* (Section 7.3.3).

— *EvalSub* : Un outil pour pour le calcul des scores à base de référence pour les sous-titres automatiques. [2]

7.2 Produire et évaluer la segmentation des sous-titres

7.2.1 Cadre du problème

Pour ce travail, nous nous concentrons seulement sur l'évaluation de la segmentation, en considérant que la sortie du système est constituée de texte entrecoupé par des symboles indiquant les limites des segments. Nous supposons de plus que ces symboles sont de deux types : `<eol>`, qui indique un changement de ligne sur le même écran, et `<eob>`, qui indique la fin d'un bloc sous-titre et le changement d'écran consécutif [3]. La figure 7.1 montre un exemple de deux sous-titres (blocs) ré-écrits avec ces notations. Selon les directives de sous-titrage courantes en anglais (BBC, 2021; Netflix, 2021; TED, 2021), il ne devrait pas y avoir plus de deux lignes par écran, et chaque ligne devrait contenir au plus 40 caractères (avec des variations en pratique selon l'audience visée). Pour faciliter la lecture, les fins de ligne et de sous-titre devraient être positionnées de manière à préserver autant que possible les unités syntaxiques et sémantiques (Carroll & Ivarsson, 1998). En outre, la durée d'affichage de chaque sous-titre devrait s'adapter au nombre de caractères à l'écran (pour respecter la vitesse de lecture), tout en restant synchrone avec le contenu parlé.

Observer ces contraintes est une tâche complexe, et les sous-titreurs professionnels

2. Notre code permettant de reproduire les expériences est disponible via le dépôt `https://github.com/fyvo/EvalSubtitle`.

3. Nous reprenons dans ce chapitre les symboles proposés par Karakanta et al. (2020b) ; dans les chapitres 5 et 3 nous utilisons les balises `<br>` et `<p>`, qui correspondent respectivement à `<eol>` et `<eob>`.

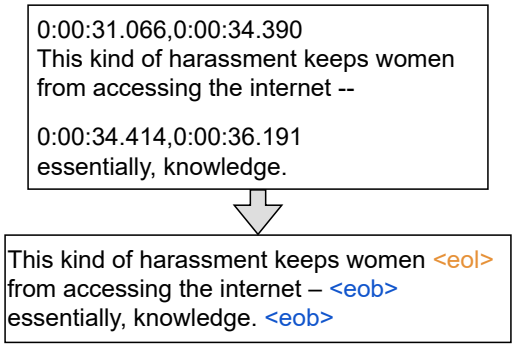


FIGURE 7.1 – Exemple d'utilisation des balises `<eol>` et `<eob>` pour représenter la segmentation de deux blocs sous-titres.

doivent souvent chercher un compromis entre les différentes règles. De plus, la question de la délimitation des segments ne dépend pas uniquement de la syntaxe ou du sens du texte (Diaz Cintas & Remael, 2007, p. 172), mais aussi de facteurs multimodaux tels que les tours de parole, les pauses dans le discours, et les changements de plan. Cela signifie qu'une importante expertise technique, linguistique et extra-linguistique est impliquée dans le processus de segmentation, et qu'à ce titre il y a beaucoup à apprendre des ressources de véritables sous-titres.

7.2.2 Métriques pour la segmentation

De façon générique, la tâche de segmentation consiste à diviser une séquence en plaçant des frontières entre les unités. Ces unités, et les segments qu'ils constituent, peuvent avoir différentes formes et granularités selon l'application considérée : des paragraphes peuvent être regroupés par passages ou thèmes dans un document (Hearst, 1997), une suite continue de caractères peut être découpe en mots (Okabe et al., 2021) etc.

Pour ce qui est de l'évaluation automatique de la segmentation des sous-titres, il existe deux approches : l'une est d'évaluer séparément le respect de chacune des consignes précédemment évoquées, puis d'en déduire un score agrégé ; l'autre consiste à essayer de reproduire des segmentations humaines de référence. Les deux présentent des avantages et des inconvénients : la première est difficile à mettre en œuvre du fait du besoin de réaliser une analyse syntaxique et sémantique des sous-titres, et d'accorder un poids juste à chaque norme ; tandis que la seconde est délicate parce que les métriques habituelles de comparaison de chaînes ne sont pas appropriées pour les sous-titres. Par exemple, un changement mineur (en termes de distance d'édition) comme l'ajout d'une balise `<eol>` supplémentaire peut donner un affichage invalide avec trois lignes. À l'inverse, l'absence d'une balise `<eol>` peut créer une ligne trop longue pour tenir même dans la largeur de l'écran. Concernant la position des frontières, il peut aussi arriver que déplacer une balise de trois mots soit moins dommageable (du point de vue des groupes syntaxiques et

sémantiques) que de la déplacer d'un mot seulement.

Nous portons ici notre attention sur les *métriques à base de références*, et exprimons qu'une bonne métrique pour la segmentation des sous-titres devrait pouvoir :

— prendre en compte plusieurs types de frontières ;
— s'adapter au scénario où plusieurs références humaines sont disponibles ;
— traiter les différences de contenu textuel avec la référence ;
— dissocier l'effet d'un mauvais contenu textuel de celui d'une mauvaise segmentation ;
— aboutir à un équilibre raisonnable entre les différentes contraintes formelles et structurelles.

Ainsi nous menons trois expériences dans lesquelles

1. nous analysons l'application des métriques classiques de segmentation à l'évaluation de la segmentation des sous-titres, en discutant du degré auquel elles remplissent les critères exprimés ci-dessus, dans le cas de textes parfaits (Section 7.3.1) ;
2. nous proposons *Sigma*, un nouveau score fondé sur BLEU_{br}, qui permet de séparer les effets respectifs de la qualité du texte et de la segmentation (Section 7.3.2) ;
3. nous comparons toutes les métriques sur des données réelles de sous-titrage automatique, engendrées par des systèmes de traduction automatique bout en bout (Section 7.3.3).

7.3 Protocole expérimental

7.3.1 Sensibilité ou robustesse des métriques

Dans la première expérience, nous examinons le comportement des métriques de segmentation standard et des métriques précédemment utilisées pour l'évaluation de la segmentation des sous-titres (BLEU_{br} et TER_{br}, voir Section 3.5), pour des textes parfaits produits artificiellement en contrôlant le niveau de dégradation de la segmentation. Les métriques testées sont les suivantes :

— **Précision, rappel et F1** (Álvarez et al., 2016) : La *précision* est définie comme la proportion de frontières dans l'hypothèse qui sont vérifiées dans la référence ; tandis que le *rappel* est la proportion de frontières dans la référence qui sont correctement prédites dans l'hypothèse. *F1* est la moyenne harmonique de la précision et du rappel.
— **Métriques fondées sur des fenêtres** : P_k (Beeferman et al., 1999) attribue une pénalité à chaque position d'une fenêtre glissante [4] si l'hypothèse et la référence

4. Dans ce contexte les frontières entre segments ne sont pas considérées comme des symboles faisant

sont en désaccord sur l'appartenance de ses deux extrémités à un même segment, tandis que *WindowDiff* (WD) (Pevzner & Hearst, 2002) attribue une pénalité si le nombre de frontières de segment à l'intérieur de chaque fenêtre diffère entre la référence et l'hypothèse.

— **Métriques fondées sur la distance d'édition** : *Segmentation similarity* (SegSim) (Fournier & Inkpen, 2012) calcule la proportion de frontières qui ne sont pas transformées quand les segmentations sont comparées en utilisant la distance d'édition en tant que fonction de pénalité. *Boundary similarity* (BoundSim) (Fournier, 2013) est une adaptation de *Segmentation similarity*, qui modifie les poids assignés à chaque opération d'édition, ainsi que la méthode de normalisation. Ces deux métriques prennent en compte les transpositions sur une certaine distance (typiquement un mot), qui réduisent la pénalité pour les quasi-erreurs (c.-à-d. les frontières prédites qui sont proches de positions de référence). Pour TER_{br} (Karakanta et al., 2020a), tous les mots à l'exception des balises de segmentation sont masqués (avec un unique symbole) dans chaque paire hypothèse-référence, préalablement au calcul du TER (Snover et al., 2006) sur ces séquences.

— **$BLEU_{br}$** (Karakanta et al., 2020a) : Score BLEU calculé sur le texte entrecoupé avec les symboles spéciaux représentant la segmentation d'affichage des sous-titres. Cette mesure a souvent été rapportée conjointement avec $BLEU_{nb}$ (*no breaks*) , version du score calculée en retirant les balises frontières de l'hypothèse et de la référence.

Afin d'étudier la sensibilité/robustesse des métriques de segmentation pour la tâche de sous-titrage, nous apportons des modifications à la référence de manière contrôlée. Plus précisément, nous modifions de manière contrôlée la référence :

— déplacer une frontière d'une, deux ou trois positions, vers la droite ou vers la gauche (`shift.1`, `shift.2`, `shift.3`);

— ajouter une frontière à une position précédemment inoccupée, c'est-à-dire entre deux mots ordinaires (`add`);

— supprimer une frontière existante (`delete`);

— changer le type d'une frontière, par exemple remplacer une balise `<eol>` par une balise `<eob>` (`replace`).

Pour chaque type d'opération, nous augmentons graduellement le pourcentage de frontières affectées (20 %, 40 %, 60 %, 80 % et 100 %). Par exemple, le scénario noté `shift.1.20` correspond au décalage d'une position de 20 % des frontières de référence, alors que `delete.80` signifie la suppression de 80 % des frontières de référence. Les pourcentages pour les cas d'addition de frontières se réfèrent au nombre initial de balises,

partie de la séquence.

REF	the car has just left Paris <eol> for its destination London <eob> where it will arrive next Sunday <eol> if all goes well . <eob>
HYP	the car has just left Paris <eol> for his destination : London <eob> where he arrives <eol> next Sunday if <eol> all goes well . <eob>

Le bigramme « left Paris » et le trigramme « all goes well » sont tous deux comptés par BLEU_{nb} et BLEU_{br} ; le bigramme « London <eob> » et le trigramme « Paris <eol> for » sont comptés par BLEU_{br} mais pas par BLEU_{nb} ; inversement le bigramme « if all » et le trigramme « arrive next Sunday » sont comptés par BLEU_{nb} mais pas par BLEU_{br}.

FIGURE 7.2 – Comparaison entre BLEU_{nb} et BLEU_{br}.

et non au nombre de positions libres (entre deux mots ordinaires) ; ainsi, `add.100` double le nombre de frontières. Pour finir, les métriques sont calculées entre les ensembles test modifiés et la référence.

7.3.2 BLEU_{br} : effets liés au contenu et à la segmentation

Dans la deuxième expérience, nous étudions si BLEU_{br} permet de saisir la qualité de la segmentation. Les deux formes de BLEU, avec (BLEU_{br}) et sans (BLEU_{nb}) frontières, ont parfois été rapportées ensemble (p. ex. dans (Karakanta et al., 2020a) et (Buet & Yvon, 2021a)), dans le but de mettre distinctement en relief le niveau du contenu textuel (par la valeur de BLEU_{nb}) et celle de la segmentation (par la valeur de BLEU_{br} relativement à celle de BLEU_{nb}). Cependant, la relation entre les deux scores suggère que cette interprétation est excessivement simplificatrice, incitant à mener une analyse plus profonde.

BLEU_{br} est calculé sur des séquences plus longues que BLEU_{nb}, ce qui implique un plus grand nombre de n-grammes à faire concorder. Puisque prédire correctement le nombre et le type des balises de segmentation est généralement plus simple que de prédire les mots eux-mêmes, BLEU_{br} a habituellement une meilleure précision unigramme, qui peut à son tour affecter les précisions d'ordre supérieur [5]. Cela laisse entendre que la différence absolue ou relative entre les deux scores n'est pas un signal exact de la qualité de la segmentation en elle-même : il peut être correct d'interpréter $\text{BLEU}_{\text{br}} > \text{BLEU}_{\text{nb}}$ comme un indice de bonne segmentation, mais l'intensité de ce signal ne peut pas être évaluée de façon réaliste à partir de ces deux mesures seules. Comment alors comparer BLEU_{nb} et BLEU_{br} ? Quand $\text{BLEU}_{\text{nb}} = 100$, dans le cas de textes parfaits, BLEU_{br} ne peut évidemment pas être plus grand ; dans cette situation, une baisse de BLEU_{br} reflète directement des erreurs de segmentation. Avec des textes imparfaits toutefois, plus BLEU_{nb} est bas, plus il aisé d'observer des valeur de BLEU_{br} supérieures à celles de BLEU_{nb}.

5. Dans la quasi-totalité de nos simulations, BLEU_{br} a une précision unigramme plus élevée que BLEU_{nb}.

Les différences entre BLEU_{br} et BLEU_{nb}, comme le montre la figure 7.2, résultent de correspondances obtenues pour les n-grammes contenant une balise de segmentation (dans la suite nommés n-tagrammes), puisque ceux ne contenant pas de balise (p. ex. « left Paris ») sont comptés de la même façon. Une première façon possible de distinguer l'effet de la segmentation serait alors de calculer séparément une autre paire de scores : BLEU_{nb} et BLEU_{em}, où la version BLEU_{em} mesure uniquement les scores de précision par rapport aux n-tagrammes, la précision unigramme ne comptant que les balises de segmentation, la précision bigramme comptant les 2-tagrammes tels que « m `<bal>` » ou « `<bal>` m », etc. Néanmoins, BLEU_{em} reste fortement corrélé avec BLEU_{nb}. La raison pour cela est que les correspondances n-tagrammes pour BLEU_{em} dépendent directement de la précision $(n-1)$-gramme pour BLEU_{nb}. Par exemple, « m_1 m_2 `<bal>` ou « m_1 `<bal>` m_2 » ne peut être correct que si « m_1 m_2 » l'est aussi ; ce qui implique que le score n-gramme définit un majorant pour la précision $(n+1)$-tagramme.

Nous écartons en conséquence BLEU_{em} et considérons plutôt un majorant de BLEU_{br}, noté $\text{BLEU}_{\text{br}}{}^{+}$, et calculé comme suit. p_1, p_2, p_3, p_4 dénotent respectivement les précisions modifiées unigramme, bigramme, trigramme et quadrigramme calculées par BLEU_{nb}, α est le rapport entre le nombre de balises et le nombre de mots ordinaires, et $p'_1, \ldots, p'_4$ désignent les précisions modifiées correspondantes pour BLEU_{br}. En supposant que les frontières sont majoritairement correctes, le nombre d'unigrammes corrects attendus dans un texte de l mots enrichi avec des balises est $p_1 \times l + \alpha \times l$, ce qui revient à $p'_1 = \frac{p_1+\alpha}{1+\alpha}$. Pour les ordres de n-grammes supérieurs, le calcul exact est plus compliqué (nous le détaillons dans l'Annexe A), mais un majorant simple peut être exprimé de la manière suivante :

$$p'_n \leq \frac{(1-(n-1)\alpha) \times p_n + n\alpha \times p_{n-1}}{1+\alpha} \tag{7.1}$$

Cette majoration est valide en supposant que : (a) chaque balise fait partie de n n-tagrammes, au sein desquels elle est entourée par des mots ordinaires [6] ; (b) les séquences sont suffisamment longues pour pouvoir faire l'approximation $\frac{l}{l+1} \approx 1$. Nous déduisons aisément $\text{BLEU}_{\text{br}}{}^{+}$ majorant de BLEU_{br}, que nous obtenons en pratique à l'aide des calculs intermédiaires de BLEU_{nb}. Cette valeur peut servir d'estimation de la meilleure valeur BLEU_{br} atteignable, étant donné un certain score BLEU_{nb}. Nous définissons ainsi notre nouveau score *Sigma* (S) comme :

$$S = 100 \times \frac{\text{BLEU}_{\text{br}}}{\text{BLEU}_{\text{br}}{}^{+}} \tag{7.2}$$

Les valeurs proches de 100 devraient indiquer une bonne segmentation (BLEU_{br} approchant de son maximum), et les valeurs les plus basses une mauvaise segmentation, cela

6. En théorie il pourrait y avoir plusieurs balises dans un même n-tagramme, mais cela impliquerait des lignes de sous-titres d'un ou deux mots, ce qui n'arrive que rarement dans nos références.

indépendamment de la valeur de $BLEU_{nb}$.

Nous vérifions empiriquement les postulats ci-dessus en deux étapes. Tout d'abord nous explorons la relation entre BLEU_{br} et BLEU_{nb} pour des sorties de système imparfaites, en laissant la segmentation inchangée. Pour simuler ces productions, nous introduisons du bruit à l'intérieur du texte de référence, sans déplacer ou changer le type des frontières. Le processus de bruitage consiste à appliquer une combinaison d'opérations d'édition (insertion, suppression, substitution, à parts égales) correspondant à un certain pourcentage du nombre de balises (de 0 à 90, avec un pas de 10). Ensuite nous passons au cas où les erreurs textuelles des textes imparfaits sont conjuguées avec des erreurs de segmentation. Nous effectuons les changements de segmentation (combinaison d'opérations sur les frontières, en suivant la même procédure que pour les mots) dans les références bruitées de la première étape, et nous comparons le comportement de notre nouveau score *Sigma* par rapport à BLEU_{br}, pour différentes valeurs de BLEU_{nb}.

7.3.3 Projection de frontières et données réelles

Dans la troisième expérience, nous sortons du cadre de la dégradation contrôlée du texte et de la segmentation pour examiner l'utilité de *Sigma* lors de l'évaluation de sorties réellement engendrées par des systèmes de sous-titrage bout en bout. Dans ce but, nous comparons *Sigma* à BLEU_{br} et TER_{br}, ainsi qu'aux métriques de segmentation classiques. Pour pallier le fait que ces métriques usuelles ne peuvent pas être calculées pour des textes imparfaits, nous mettons en place une méthode de projection de frontières fondée sur l'alignement entre référence et hypothèse, comme illustré par la figure 7.3. Étant donné $\text{Ref}(1,..,i)$ et $\text{Hyp}(1,..,j)$ une paire de séquences référence-hypothèse, où i et j sont respectivement les nombres de lignes de sous-titres dans la référence et l'hypothèse, nous découpons chaque séquence au niveau des balises, de façon à obtenir i et j sous-segments de part et d'autre. Alors les sous-titres de référence sont alignés avec ceux de l'hypothèse selon l'algorithme MWER (Matusov et al., 2005). Après quoi est obtenue une nouvelle référence $\text{Ref}_{\text{proj}}(1,..,j)$, qui contient le texte de référence segmenté aux positions de balise projetées depuis l'hypothèse. Comme Hyp et Ref_{proj} ont le même nombre de lignes de sous-titres, les types des balises de Ref_{proj} sont simplement copiés sur ceux de l'hypothèse. Ce transfert de frontières de l'hypothèse vers la référence nous permet d'appliquer les métriques de segmentation standard entre la référence projetée Ref_{proj} et la vraie référence Ref, de la même façon que dans la section 7.3.1.

Les sorties réelles à évaluer proviennent de précédents travaux : nous mettons en œuvre la projection de frontières pour les productions des quatre systèmes proposés par Karakanta et al. (2020a) dans la direction En→Fr. Ces systèmes sont, respectivement, un modèle de traduction automatique neuronale (`NMT`), un modèle de traduction de parole en

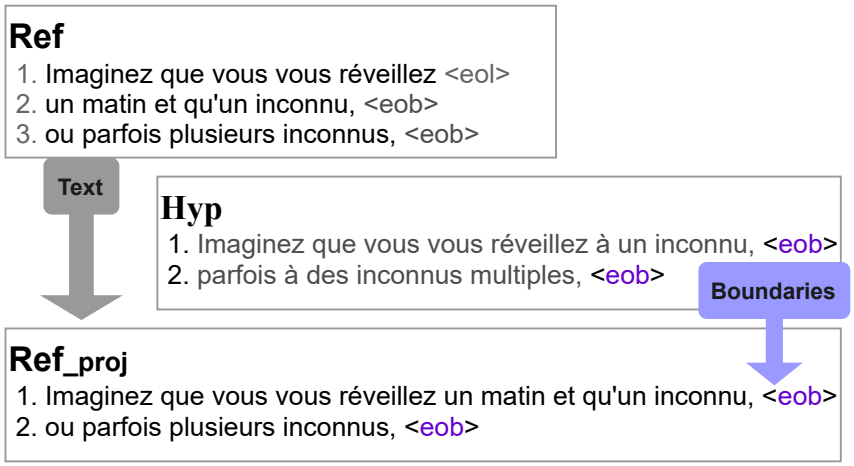


FIGURE 7.3 – Projection de frontières de l'hypothèse vers la référence fondée sur l'alignement des sous-titres.

cascade (`Cas`), et deux modèles de traduction de parole bout en bout : $\texttt{e2e}_{base}$, entraîné uniquement sur MuST-Cinema, et $\texttt{e2e}_{pt}$, pré-entraîné sur de grandes quantités de données de traduction de parole et affiné sur MuST-Cinema. Nous effectuons les mesures sur la segmentation et discutons du classement des systèmes en comparant *Sigma* avec : (a) les métriques classiques de segmentation appliquées après projection des frontières ; (b) les valeurs BLEU_{br} et TER_{br} calculées directement sur les sorties (sans projection).

7.3.4 Données et implémentation

Les données de sous-titrage utilisées dans les expériences proviennent du corpus MuST-Cinema (Karakanta et al., 2020b). L'ensemble test de MuST-Cinema est une compilation des fichiers de sous-titres de neuf « TED talks », totalisant 545 phrases entrecoupées de symboles spéciaux qui marquent les frontières de sous-titres. Pour les expériences des sections 7.3.1 et 7.3.2 nous utilisons la partie anglaise des paires anglais-français, tandis que pour la méthode de projection de frontières nous utilisons la partie française.

Le code pour réaliser les mesures sur la segmentation est implémenté en python, et repose sur des librairies existantes. Les métriques fondées sur des fenêtres (P_{k}, WindowDiff), de même que *Segment Similarity* et *Boundary Similarity*, sont calculées à l'aide du paquet *SegEval*[7] (Fournier, 2013). BLEU et TER sont calculés avec *SacreBLEU*[8] (Post, 2018).

7. `https://pypi.org/project/segeval/`
8. `BLEU|#:1|c:mixed|e:no|tok:13a|s:exp| v:2.0.0`
`TER|#:1|c:lc|t:tercom|nr:no|pn:yes|as:no| v:2.0.0`

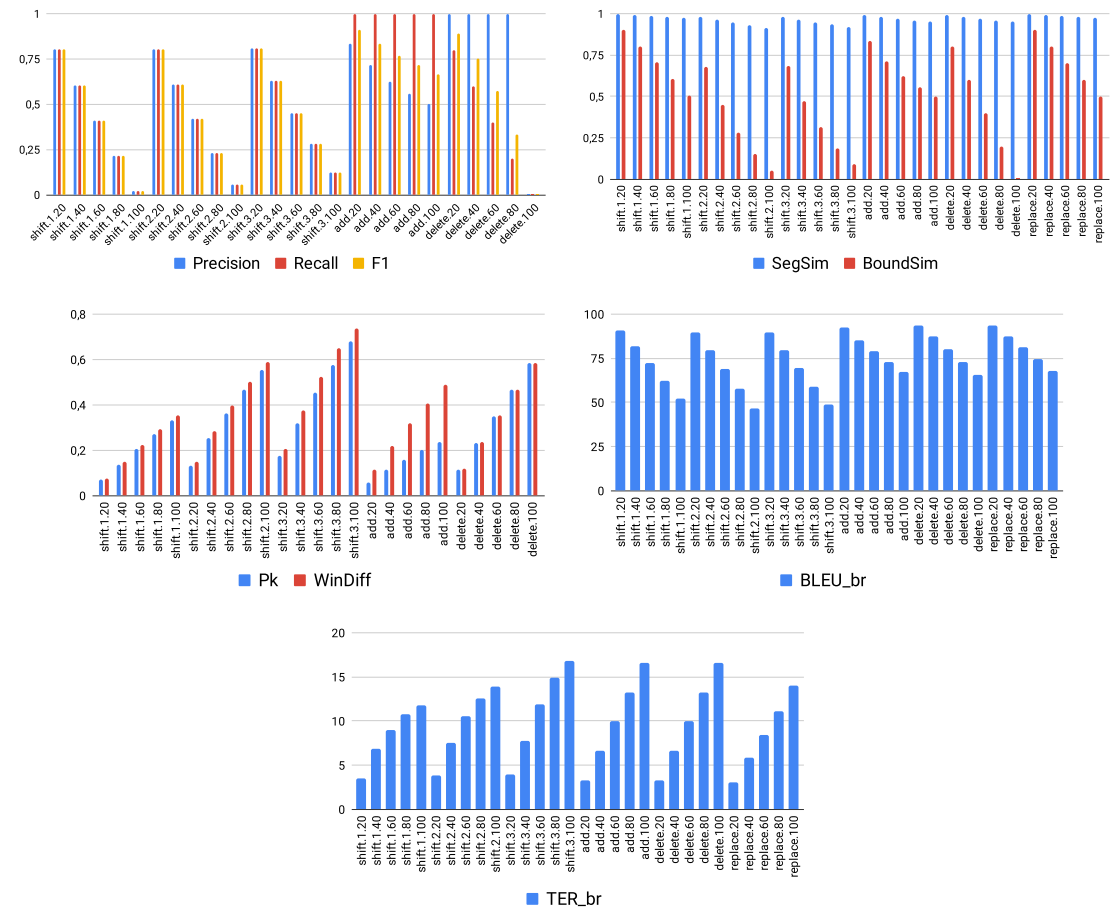


FIGURE 7.4 – Comportement des métriques de segmentation lorsque la segmentation de référence est graduellement perturbée en appliquant sur les frontières les opérations `shift.1`, `shift.2`, `shift.3`, `add`, `delete`, et `replace`. Les métriques qui ne traitent pas la distinction entre différents types de frontières n'ont pas été calculées pour le cas `replace`.

7.4 Résultats

7.4.1 Sensibilité aux changements de segmentation

Nous observons ici l'impact de différents types et niveaux de bruit sur les métriques de segmentation présentées à la section 7.3.1. Outre les souhaits exprimés à la section 7.2.1, nous formulons certaines attentes concernant la pénalisation des perturbations, en fonction de leur nature et de leur incidence sur l'expérience des usagers. Les déplacements (`shift`) correspondent à avoir le bon nombre de sous-titres, mais segmentés de manière non-optimale pour la compréhension. Dans la situation d'un décalage d'une seule position, il est d'autant plus vraisemblable que l'intégrité des unités syntaxiques adjacentes soit compromise (puisque les sous-titres de référence sont supposés respecter un décou-

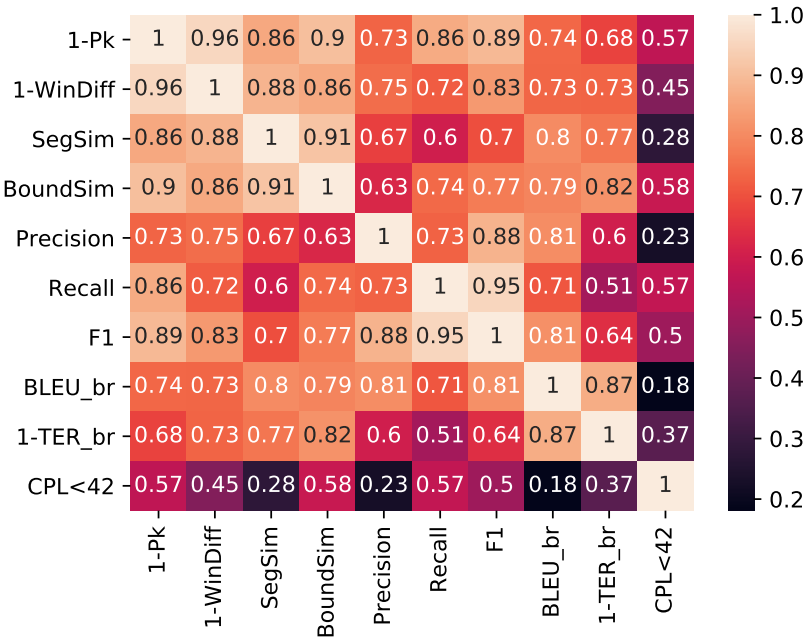


FIGURE 7.5 – Matrice de corrélation linéaire entre les métriques de segmentation. Les coefficients ont été calculés avec les valeurs mesurées au cours de l'expérience décrite section 7.3.1. CPL<42 est le pourcentage de lignes de sous-titres conformes à la norme de longueur maximale de 42 caractères.

page syntaxique cohérent, et que les syntagmes unitaires ne sont pas majoritaires). Pour cette raison, une métrique adaptée à la segmentation des sous-titres ne devrait pas être moins sensible aux quasi-erreurs. Quant aux opérations d'addition (`add`), de suppression (`delete`) et de substitution (`replace`), chacune peut mener à une erreur critique ; une suppression impliquera un sous-titre trop long, une addition provoquera un sous-titres trop court, et un remplacement pourra résulter en un trop grand nombre de lignes. En l'absence d'étude montrant clairement l'effet de chaque opération sur l'expérience utilisateur, nous préférons qu'une métrique pénalise de manière égale la sur-génération et la sous-génération de frontières, ainsi que l'interversion du type de frontière. Les résultats des scénarios de dégradation sont illustrés par la figure 7.4. La corrélation entre les métriques est analysée au travers de la matrice des coefficients de Bravais-Pearson, dans la figure 7.5.

Pour la précision et le rappel, le déplacement des balises est à l'origine des plus fortes baisses, bien que le nombre de frontières soit préservé. De façon intéressante, les déplacements d'une position sont plus défavorables que ceux de deux positions, eux-mêmes pires que ceux de trois positions (ce qui correspond au fait que les lignes de deux ou trois mots sont plus fréquentes que celles qui sont unitaires). La différence se voit davantage pour les pourcentages au-delà de 60 %. F1 est plus bas pour les suppressions que pour les additions, du fait d'une détérioration plus importante du rappel.

Par définition, l'erreur mesurée par P_k est toujours moins importante que celle mesurée par WindowDiff (une erreur pour P_k est une erreur pour WindowDiff, mais l'inverse n'est pas vrai). Comme l'avaient noté et critiqué Pevzner & Hearst (2002), P_k pénalise les faux négatifs plus lourdement que les faux positifs (dans notre expérience, les faux négatifs correspondent aux suppressions, et les faux positifs aux additions). Ainsi P_k semble être plus orientée vers le rappel que les autres métriques, ce que confirment les coefficients de corrélation. Comme la précision et le rappel, P_k et WindowDiff sont plus sensibles aux suppressions qu'aux additions. Cependant pour ces métriques, et à l'inverse de la précision et du rappel, la pénalisation des déplacement augmente avec leur taille.

SegSim calcule des valeurs toujours assez hautes dans l'absolu (comme mentionné par Fournier (2013)), ce qui peut être gênant pour l'interprétation par défaut de lisibilité. La nouvelle normalisation introduite pour BoundSim résout notamment ce problème. De même que les métriques fondées sur les fenêtres, SegSim et BoundSim sont plus sensibles aux suppressions qu'aux additions, et pénalisent moins les quasi-erreurs (ici `shift.1`). Cela s'explique par le fait que le décalage de frontière par une position est comptabilisé comme une transposition, tandis que les déplacements plus longs coûtent une addition et une suppression du point de vue de la distance d'édition. Au contraire de SegSim et BoundSim, TER_{br} est relativement indifférent au type d'erreur, montrant un équilibre entre `add`, `delete` et `shift.3`. Toutes les métriques fondées sur la distance d'édition sont cependant moins sensibles aux substitutions.

$BLEU_{br}$ reste globalement dans l'intervalle 45 – 100, à cause du contenu textuel qui n'est pas bruité (il s'agit de la seule métrique à le prendre en compte). Il n'y a pas vraiment de sensibilité par rapport à la taille des déplacements, les décalages par une, deux ou trois positions obtenant approximativement les mêmes scores ; néanmoins les déplacements sont davantage sanctionnés que les autres types de dégradation. Bien qu'étant une métrique fondée sur la précision, $BLEU_{br}$ est ici en partie robuste au type d'erreur, pénalisant de manière égale les suppressions, les additions, ainsi que les substitutions. Il faut considérer le fait qu'une frontière omise à tort (faux négatif) affectera la précision n-gramme (pour $n > 1$), quoique de façon moins importante qu'une frontière incorrectement ajoutée. Ainsi, cet équilibre entre la réponse aux additions et aux suppressions pourrait être attribué à l'effet de la pénalité de brièveté (*brevity penalty*), qui diminue le score des segments auxquels manquent des balises.

Enfin, pour revenir aux conditions que nous avions définies pour une bonne métrique de segmentation : la capacité à prendre en compte différents types de frontières est présente dans SegSim, BoundSim, $BLEU_{br}$, et TER_{br}. $BLEU_{br}$ et TER_{br} peuvent exploiter plusieurs références humaines. Concernant l'équilibre entre les contraintes formelles et structurelles, alors que toutes les métriques se corrèlent entre elles assez fortement, la

corrélation avec la conformité à la norme de longueur (CPL<42 dans la figure 7.5) est en général relativement faible, avec un coefficient supérieur à $0,5$ uniquement pour P_k, BoundSim et le rappel. Les métriques précision-rappel peuvent donner une idée de la nature des erreurs commises (sur-segmentation ou sous-segmentation), mais elles devraient être calculées séparément pour chaque type de balise puis combinées. Cela suggère qu'il n'existe pas encore de métrique qui incorpore réellement les contraintes formelles et structurelles.

Remarquons aussi qu'une tolérance spécifique pour les quasi-erreurs n'est pas une propriété souhaitable pour une métrique de segmentation de sous-titres (même si elle pourrait être pertinente dans un autre contexte). Au contraire des mesures fondées sur les fenêtres ou la distance d'édition, $BLEU_{br}$ est en phase avec nos attentes au sujet de la distance de déplacement des frontières. $BLEU_{br}$ réalise aussi une certaine symétrie entre les opérations d'addition, de soustraction, et de remplacement. Ainsi $BLEU_{br}$ semble avoir des traits correspondant à nos critères pour une bonne métrique de segmentation. Toutefois nos présomptions sur la relation entre les métriques et l'expérience des usagers doivent encore être validées à la lumière de jugements humains ; seule une étude recueillant des avis d'utilisateurs permettra de conclure définitivement sur la qualité des métriques.

7.4.2 Que mesure vraiment $BLEU_{br}$?

Malgré ses défauts, $BLEU_{nb}$ reste une métrique incontournable pour la recherche sur la traduction et le sous-titrage automatiques. Pour ce qui concerne son rapport avec la qualité de la segmentation, nous avons montré à la section 7.4.1 que pour un texte parfait ($BLEU_{nb} = 100$), les valeurs de $BLEU_{br}$ sont effectivement cohérentes avec le niveau de dégradation (voir le quatrième graphe de la Figure 7.4), ayant même l'avantage d'être sensible aux différences de type de frontière, tout autant qu'aux ajouts et aux suppressions. La question vers laquelle nous nous tournons désormais est celle de l'utilité de $BLEU_{br}$ *pour des textes imparfaits*. Nous voudrions évaluer indépendamment la qualité du contenu textuel (avec $BLEU_{nb}$) et la qualité de la segmentation dans la prédiction du système. Les mesures $BLEU_{br}$ (utilisée dans ce sens dans les travaux précédents) et *Sigma* (Equation (7.2)) peuvent-elles remplir ce rôle ?

Tout d'abord, dans le cas d'une segmentation « parfaite » mais d'un texte imparfait, la figure 7.6 montre que la relation entre $BLEU_{br}$ et $BLEU_{nb}$ est linéaire. Cela confirme l'hypothèse selon laquelle les deux métriques sont trop corrélées pour que leur différence puisse indiquer clairement la qualité de la segmentation. Il apparaît également que $BLEU_{br}$ ne peut pas dépasser (puisque dans cette configuration la segmentation n'a pas été affectée par le bruit) une borne supérieure qui est fortement liée à $BLEU_{nb}$. Étant donné la dépendance entre $BLEU_{br}$ et $BLEU_{nb}$, rapporter les deux scores n'est pas instructif, et

le signal de la segmentation devrait être recherché dans une autre relation.

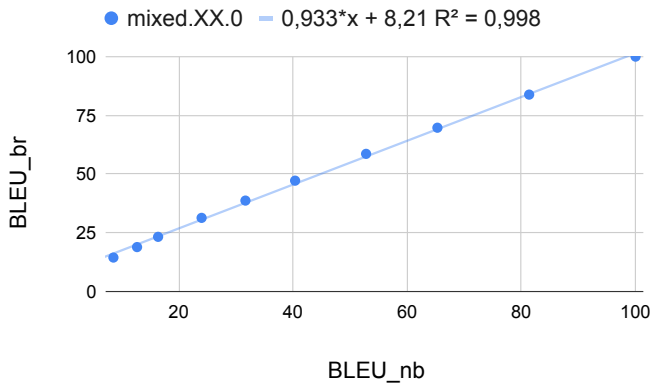
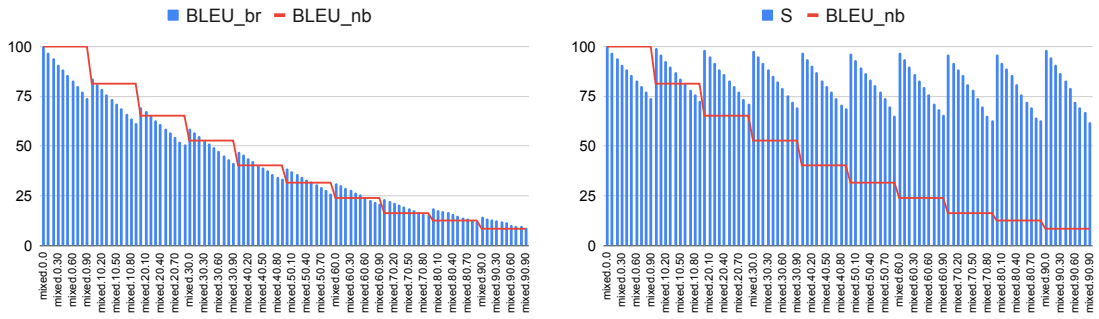


FIGURE 7.6 – Linéarité de BLEU_{br} par rapport à BLEU_{nb}, pour des instances où seul le texte a été bruité (voir Section 7.3.2). La régression linéaire donne un coefficient de détermination de $0,998$, et une erreur type de $1,2$.

Ensuite, dans le cadre d'un texte et d'une segmentation imparfaits, la figure 7.7a confirme le postulat selon lequel les occurrences où BLEU_{br} dépasse BLEU_{nb} sont plus nombreuses pour les petites valeurs de BLEU_{nb}. Et par opposition à BLEU_{br}, *Sigma* conserve le même domaine indépendamment de la valeur de BLEU_{nb}. Ceci est illustré par la figure 7.7b, dans laquelle *Sigma* est tracé pour diverses valeurs de BLEU_{nb}. *Sigma* réagit linéairement à la quantité de bruit apporté à la segmentation, mais nous observons une dérive de son intervalle de valeurs lorsque BLEU_{nb} décroît (de $[74,3\,;100]$ pour `mixed.0` à $[63,2\,;95,8]$ pour `mixed.90`). Néanmoins, dans un scénario réaliste, BLEU_{nb} serait typiquement contraint à un intervalle entre 25 et 55 (correspondant aux quatre séries centrales de la Figure 7.7b). De plus, l'incidence de cette dérive serait d'autant plus limitée quand serait comparée la capacité à segmenter de deux systèmes proches en termes de valeurs BLEU_{nb}. Il en résulte que *Sigma* peut être une bonne estimation pour capturer la qualité de segmentation, quelle que soit la qualité du contenu textuel engendré.

7.4.3 Évaluation de vraies productions

Nous nous intéressons maintenant à l'évaluation de la segmentation pour de véritables sorties de systèmes bout en bout. Comme les métriques classiques ne peuvent pas être appliquées avec des textes imparfaits, nous utilisons la méthode de projection de frontières afin de comparer la hiérarchie des systèmes selon ces métriques et selon *Sigma*. Les scores sont rapportés dans les tableaux 7.1 et 7.2. *Sigma* classe le système `NMT` comme le meilleur avec un score de $89,2$, suivi par `Cas` avec $83,1$, puis par les systèmes directs qui sont assez proches aux alentours de $81,5$. Les métriques classiques calculées sur la référence projetée (Tableau 7.1) ainsi que BLEU_{br} et TER_{br} (Tableau 7.2) s'accordent



(a) (b)

FIGURE 7.7 – Valeurs de BLEU_{br} (a) et *Sigma* (b) après avoir appliqué du bruit à la segmentation, pour différents seuils de BLEU_{nb} (10 niveaux de bruitage de la segmentation pour chaque seuil BLEU_{nb}, 10 seuils BLEU_{nb} de 100 à 8,5). (a) montre que BLEU_{br} décroît avec BLEU_{nb}, tandis que dans (b) *Sigma* reste stable.

clairement avec *Sigma* sur le fait que le système `NMT` produit les meilleures segmentations parmi celles examinées. Pour les conférences TED, les sous-titres traduits (qui sont nos références dans le corpus MuST-Cinema) sont habituellement rédigés en prenant comme patron ceux en langue originale ; par conséquent il n'est pas surprenant qu'un modèle prenant en entrée les sous-titres en langue source puisse atteindre un haut niveau de similitude avec la référence (il s'agit essentiellement de copier correctement les balises). Cependant, l'accord général entre métriques ne tient plus pour ce qui concerne les trois systèmes de traduction de parole, `Cas`, $\texttt{e2e}_{base}$ et $\texttt{e2e}_{pt}$. L'architecture en cascade semble ici avoir l'avantage, obtenant six fois une valeur sensiblement supérieure (pour WindowDiff, SegSim, la précision, BLEU_{br} et les deux versions de TER_{br}), et étant dans trois cas comparable avec $\texttt{e2e}_{pt}$, qui est meilleur d'après BoundSim et le rappel. Les résultats relativement proches pour beaucoup de mesures, en particulier *Sigma*, ne permettent pas de départager clairement les deux systèmes directs. Cela pouvait être attendu, puisque $\texttt{e2e}_{pt}$ était pré-entraîné sur du texte non-segmenté, ce qui améliore la qualité de la traduction, mais n'apporte pas de données de segmentation supplémentaires par rapport à $\texttt{e2e}_{base}$. La longueur de la sortie engendrée paraît avoir aussi de l'importance : les métriques fortement corrélées avec la conformité de longueur des lignes (CPL<42) placent $\texttt{e2e}_{pt}$ devant `Cas` ($\text{CPL} < 42 = 95\ \%$ pour $\texttt{e2e}_{pt}$, et $\text{CPL} < 42 = 91\ \%$ pour `Cas`). Dans l'ensemble, les deux systèmes fondés sur la traduction de parole semblent des erreurs de types différents (`Cas` a une plus grande précision, $\texttt{e2e}_{pt}$ a un meilleur rappel), mais pour la plupart des métriques les valeurs sont si proches qu'il est difficile de dire quelle segmentation est à préférer.

Les résultats montrent que, quoique les métriques soient capables de récompenser justement les bonnes productions, il est difficile dans un cadre d'évaluation réel de faire la

Systèmes	P_k	WD	SSim	BSim	Préc.	Rapp.	F1	$BLEU_{br}$	TER_{br}
NMT	0,192	0,208	0,979	0,637	0,711	0,735	0,723	83,18	6,87
Cas	0,252	0,270	0,970	0,519	0,639	0,667	0,653	76,14	8,91
$e2e_{base}$	0,257	0,277	0,969	0,515	0,601	0,667	0,632	75,00	9,29
$e2e_{pt}$	0,252	0,276	0,969	0,525	0,610	0,702	0,653	74,89	9,24

TABLE 7.1 – Résultats de l'évaluation de la segmentation pour quatre systèmes bout en bout, en utilisant la méthode de projection de frontières (Section 7.4.3). Les meilleurs scores parmi les systèmes prenant de la parole en entrée sont soulignés.

Systèmes	$BLEU_{br}$	TER_{br}	*Sigma*
NMT	32,16	19,38	89,2
Cas	26,34	23,23	83,1
$e2e_{base}$	22,53	24,48	81,8
$e2e_{pt}$	26,36	23,52	81,5

TABLE 7.2 – Résultats de l'évaluation de la segmentation pour quatre systèmes bout en bout, avec des métriques supportant les textes imparfaits. Les meilleurs scores parmi les systèmes prenant de la parole en entrée sont soulignés.

distinction entre des sorties dont la qualité de segmentation est similaire, cela demandant des mesures suffisamment précises. Néanmoins, *Sigma* s'accorde relativement bien avec la majorité des métriques concernant le classement du modèle en cascade au sein des systèmes de traduction de parole. La projection de frontières est utilisée ici comme un moyen pour dissocier les effets de la qualité du texte et de la segmentation, mais les scores obtenus en suivant cette méthode sont affectés par la performance de l'algorithme d'alignement, notamment pour les sorties de pauvre qualité. D'un autre côté, les métriques calculées directement sur des textes imparfaits ($BLEU_{br}$ et TER_{br}) sont fortement influencées par la qualité de la traduction, comme le suggèrent entre autres les différences d'évaluation lorsque $BLEU_{br}$ est appliqué pour une référence projetée (Tableau 7.1) ou pour l'hypothèse elle-même (Tableau 7.2). *Sigma* n'est contraint par aucune de ces limitations, et fournit une solution claire et aisément exécutable pour l'évaluation de la segmentation de textes imparfaits.

Enfin nous faisons le lien avec le chapitre 6, en évaluant avec *Sigma* les systèmes de sous-titrage intralingues avec architecture en cascade qui y avaient été présentés : simplification et segmentation séparées (Transformer + règles), simplification et segmentation conjointes grâce aux balises de segmentation (Transformer + BS), adaptation au genre par étiquettes de domaine (+ BG) ou par contrôle de la longueur/fréquence d'affichage (+ $LRPE_{CPS^*}$), et utilisation de données rétro-traduites complémentaires (+ RT). Les résultats se trouvent dans la figure 7.8. Nous remarquons en premier lieu que de façon éton-

nante, le score *Sigma* n'augmente pas nécessairement avec la taille du corpus d'apprentissage ; effectivement le système Transformer + BS obtient bien un meilleur score en étant entraîné sur le point d'étape V4 du corpus (la dernière et plus grande version), mais obtient une valeur *Sigma* plus haute avec V1 qu'avec V2 ou V3, alors que du point de vue de la quantité de données V1<V2<V3. Nous pouvons également voir que la segmentation est mieux réalisée conjointement avec la simplification (grâce aux balises de segmentation), que séparément avec le module à règles (la différence *Sigma* entre Transformer + règles et Transformer + BS est $7,7$, alors que l'écart en BLEU_{nb} n'est que de $0,3$). Une influence positive peut être notée pour l'usage de données rétro-traduites ; les exemples d'apprentissage engendrés par rétro-traduction sont peut-être particulièrement bénéfiques pour la segmentation parce que la principale transformation qu'ils exhibent est l'insertion des balises d'affichage (tandis les vrais exemples présentent davantage de différences au niveau du contenu textuel). Concernant l'adaptation au genre télévisuel, la méthode par étiquettes de domaine ne permet pas d'améliorer la valeur *Sigma* (une baisse sensible est même visible entre Transformer + BS et Transformer + BS + BG). Au contraire, la méthode de contrôle de longueur LRPE obtient ici le meilleur résultat ($\textit{Sigma} = 71,14$), alors qu'elle réalise le plus bas BLEU_{nb} $(40,1)$ parmi les système comparés sur la figure 7.8 (le plus haut score BLEU_{nb} étant de $45,5$, pour Transformer + BS + RT + BG). Ainsi, les approches qui avaient apporté des améliorations à la qualité du texte simplifié ne se révèlent pas systématiquement avantageuses pour la segmentation ; dans le futur l'expérimentation avec les méthodes de contrôle de longueur pourraient être approfondies, en relâchant notamment la contrainte sur la fréquence d'affichage afin d'atténuer la dégradation selon les métriques BLEU_{nb} et SARI.

7.5 Conclusion

Nous avons analysé des métriques et des méthodes pour évaluer avec référence une segmentation de texte correspondant à des sous-titres mis en forme pour l'affichage. Notre expérience d'ajout de bruit artificiel à la segmentation montre que pour des textes parfaits, BLEU_{br} remplit nos critères pour ce que doit être une bonne métrique de segmentation de sous-titres. Toutefois, dans la situation de textes imparfaits, BLEU_{br} est fortement corrélé avec le score BLEU habituel (BLEU_{nb}) ; ainsi l'information sur la segmentation ne peut pas être extraite d'une simple différence entre BLEU_{br} et BLEU_{nb}. Nous introduisons donc *Sigma*, une nouvelle mesure pour la segmentation des sous-titres, qui correspond au rapport entre BLEU_{br} et une estimation de sa valeur potentielle maximale. Afin de comparer *Sigma* avec les métriques de segmentation classiques pour l'évaluation de vraies sorties de systèmes bout en bout, nous proposons en outre une méthode qui transfère les balises de sous-titres de l'hypothèse vers la référence. Nous notons que dans un cadre réel

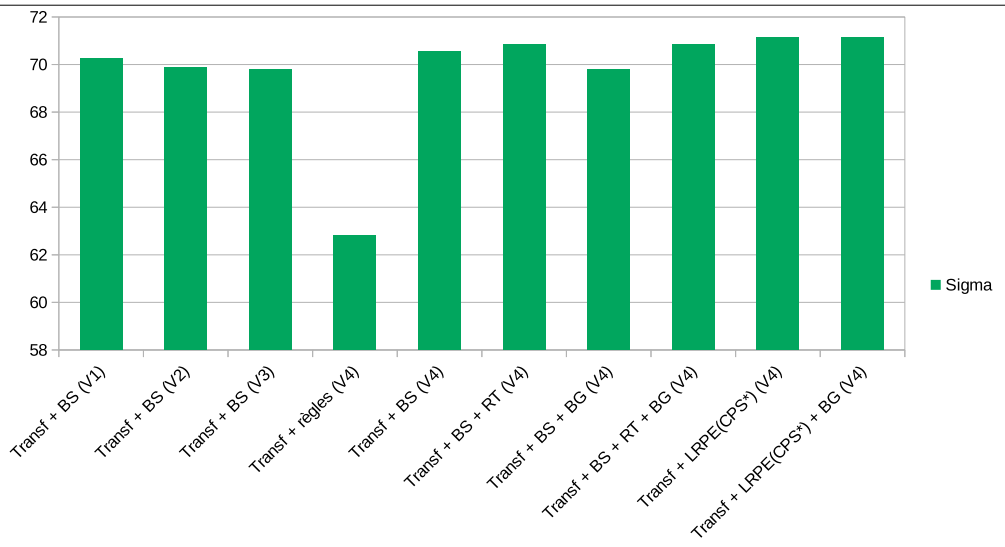


FIGURE 7.8 – Comparaison des scores *Sigma* (moyenne sur le groupe d'émissions de test du corpus de sous-titrage en français, voir Chapitre 5) pour différents systèmes de sous-titrage intralingues en cascade, évoqués au chapitre 6. BS, RT et BG dénotent respectivement l'usage de balises de segmentation (p. opp.à un module de segmentation à règles), de rétro-traduction, et de balises de genre. CpR* et CPS* indiquent que les consignes de longueur selon CPS sont adaptées pour chaque domaine. V1 – 4 indique le point d'étape du corpus d'apprentissage.

d'évaluation, les métriques existantes ne s'accordent pas toujours sur le classement entre les productions. Nous pensons qu'une conclusion définitive sur l'exactitude et la précision des méthodes d'évaluation de la segmentation ne sera possible qu'après considération de la corrélation avec le jugement humain. Néanmoins, l'analyse présentée dans ce chapitre met en lumière des aspects critiques de l'évaluation de la segmentation de sous-titres, qui seront à prendre en compte pour la conception d'études sur l'expérience des usagers, ou pour l'amélioration du calcul de *Sigma*.

Chapitre 8

Conclusion

Dans cette thèse nous avons étudié l'utilisation de méthodes neuronales de simplification automatique pour la production de sous-titres intralinguistiques. Nos travaux se sont d'abord portés sur l'analyse des mécanismes de contrôle de longueur dans les modèles encodeur-décodeur, qui permettent d'appliquer différents degrés de compression aux phrases engendrées (Chapitre 4). Nous avons ensuite procédé à la création d'un corpus d'apprentissage pour le sous-titrage d'émissions télévisées en français, en recueillant les vidéos et les sous-titres des programmes fournis par france.tv access, dans le cadre du projet ROSETTA (Chapitre 5). Cet ensemble de données nous a permis d'entraîner des systèmes de sous-titrage automatique, que nous avons adaptés aux genres télévisuels, notamment en utilisant des méthodes de compression contrôlée (Chapitre 6). Enfin nous avons examiné le problème de l'évaluation de la segmentation d'affichage des sous-titres, qui ne peut pas être effectuée avec les métriques classiques pour la segmentation si le système réalise conjointement la génération du texte et le placement des coupures : cette lacune nous conduit à proposer une nouvelle métrique, *Sigma*.

Au chapitre 4, nous avons effectué une étude comparative de plusieurs méthodes de contrôle de longueur, pour la compression et la décompression de phrase. Les résultats reflètent l'existence d'un compromis entre l'exactitude du contrôle et la qualité du texte produit. Les modèles qui encodent la longueur totale visée présentent un avantage du point de vue de la qualité, tandis que ceux qui encodent la longueur restante à chaque étape de décodage sont plus précis sur la réalisation de l'objectif. Il apparaît aussi qu'un encodage de la longueur par plongement est préférable à un encodage continu par la norme en ce qui concerne la précision du contrôle. Un sondage des états cachés du décodeur semble indiquer que l'objectif de longueur fourni extérieurement y est encodé distinctement de la représentation interne de la longueur (expliquant la relativité du contrôle).

Au chapitre 5 nous avons présenté la création du corpus pour le sous-titrage automatique d'émissions de télévision françaises, à partir de données (vidéos et sous-titres professionnels) collectées au cours du projet ROSETTA. Le traitement de ces données comprenait la récupération des fichiers au travers d'une API exposée par france.tv access, la transcription des émissions avec un logiciel commercial, et l'alignement des sous-titres

avec les transcriptions automatiques. La version finale du corpus contient près de 30 millions de mots transcrits, qui correspondent à environ 3 millions de sous-titres (1278 heures de vidéos). L'examen des données par diverses métriques révèle des variations significatives en fonction du mode production des sous-titres (`stock` ou `direct`), ainsi que du genre télévisuel (qui implique certains thèmes et certains types de langue parlée).

Le chapitre 6 était centré sur l'apprentissage et l'évaluation de modèles de sous-titrage en cascade, en utilisant le corpus du chapitre 5. L'architecture de nos systèmes prévoit une étape de transcription automatique (avec le logiciel commercial que nous n'avons pas modifié), et une étape de simplification et de segmentation conjointes (réalisée avec *Transformer*, et rendue possible par l'intégration de balises symbolisant les coupures parmi les mots). Du fait de la forte variabilité constatée dans ce corpus, nous nous sommes en particulier concentrés sur l'application de méthodes d'adaptation au genre télévisuel. Nos expériences confirment l'apport des méthodes classiques d'adaptation (étiquettes placées en tête des entrées, affinage des paramètres sur un domaine) ; cependant la méthode d'adaptation par contrôle de longueur que nous avons proposée se montre dans l'ensemble peu concluante (permettant tout de même une amélioration notable du respect des normes sur la segmentation des sous-titres). Les résultats suggèrent également que la performance de la reconnaissance de parole n'est probablement pas l'obstacle principal à la progression des systèmes de sous-titrage. L'évaluation automatique a été complétée par un jugement humain de la part d'experts métier, qui corrobore le bénéfice entraîné par l'augmentation du volume de données et par l'adaptation avec étiquettes de domaine.

Au chapitre 7 nous avons mené une analyse des métriques pour l'évaluation de la segmentation, plus spécifiquement dans un contexte correspondant à la mise forme pour l'affichage (lignes et blocs) de sous-titres. Les métriques de segmentation classiques supposent que le contenu textuel (c.-à-d. la séquence d'unités à laquelle est associée la subdivision) soit identique entre l'hypothèse et la référence. Or pour certains systèmes de sous-titrage (comme ceux du Chapitre 6) la segmentation n'est pas réalisée par un module séparé, mais est intégrée sous la forme de balises dans le processus de décodage des mots, et en conséquence le contenu textuel de l'hypothèse est en général différent de celui de la référence. Pour contourner cet obstacle, nous introduisons *Sigma*, une nouvelle métrique fondée sur des calculs de BLEU (en enlevant ou en gardant les balises dans les phrases). Nos observation préliminaires (qui n'incluent pas de mise en corrélation avec le jugement humain) attestent que *Sigma* a le potentiel de permettre de comparer des segmentations dont les contenus textuels diffèrent.

Perspectives

Un des principaux problèmes abordés dans cette thèse concernait le conditionnement de la génération selon des contraintes strictes. Dans le cadre du sous-titrage, la synchronisation avec l'information audio-visuelle et la vitesse de lecture imposent une limite sur la longueur du texte associé aux énoncés ; de plus, ce texte doit être segmenté pour l'affichage en observant des règles clairement définies (au maximum deux lignes par bloc, et 36 caractères par ligne). Nos expériences ont montré que des méthodes de contrôle de longueur comme LRPE et LDPE sont capables d'imposer assez précisément le respect de ces normes. Toutefois les modèles fondés sur ces méthodes ont aussi obtenu des scores de qualité de phrase plutôt faibles, en comparaison avec les autres modèles adaptés. Les travaux sur la production contrôlée de texte devraient non seulement s'assurer de l'effectivité de la maîtrise sur les paramètres considérés, mais aussi chercher un bon compromis avec la préservation de la grammaticalité et du sens.

Comme expliqué au chapitre 3, la forme des sous-titres est conventionnée sous de multiples angles : outre les contraintes spatiale et temporelle évoquées ci-dessus, il existe des normes pour la couleur et le placement des sous-titres, l'indication du nom du locuteur (p. ex. « -F.Moreau : ... »), le respect des changements de plan, les indications sonores et musicales (p. ex. « [applaudissements] »). Si actuellement la plupart des études (dont les nôtres) ignorent nombre de ces caractéristiques, le développement récent de ressources et de méthodes pour la traduction de parole, traduction multimodale, et la description audio peut laisser présager une orientation progressive de la recherche vers des architectures de sous-titrage automatique de plus en plus complètes.

Une conséquence de la transition vers des approches bout en bout est le renforcement de l'aspect « boîte noire » associé aux modèles appris : alors que les différents composants d'une architecture en cascade peuvent être examinés individuellement, une architecture bout en bout agglomère les différents traitements et transformations en un seul bloc. Il devient alors plus ardu d'interpréter les résultats d'un modèle, et de faire la distinction entre les différentes sources d'erreurs. Ce constat incite à analyser les systèmes dans la logique des travaux pour la compréhension de réseaux de neurones [1]. Dans le contexte de la production de sous-titres, nous avons par exemple été confrontés à la difficulté d'évaluer la segmentation séparément de la qualité du texte. La solution pour laquelle nous avons opté au chapitre 7, avec la métrique *Sigma*, consiste à utiliser un calcul qui neutralise l'influence des divergences entre le texte de l'hypothèse et celui de la référence. Une alternative serait de réaliser un décodage forcé de la phrase cible, en ne laissant au modèle que la liberté d'insérer les balises frontières, imposant ainsi la génération d'une

1. `https://blackboxnlp.github.io/`

segmentation avec le même contenu textuel que la référence.

Pour terminer, nous souhaitons revenir sur le sujet de la génération contrôlée de texte, afin de réaffirmer son intérêt pour la tâche de sous-titrage. Les demandes des utilisateurs concernant le niveau de simplification sont en effet très variables, notamment parmi les personnes sourdes et malentendantes (ce que nous ont confirmé nos échanges au sein du projet ROSETTA). Il est ainsi possible qu'à l'avenir, une production de sous-titres multi-complexité soit à envisager.

Bibliographie

2005-102 (2005). Loi n° 2005-102 du 11 février 2005 pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées (1). Consultable en ligne : https://www.legifrance.gouv.fr/loda/id/JORFTEXT000000809647/. Dernier accès : 04/07/2022. [Cited on pages 17 and 35.]

2010/13/UE (2010). Directive n° 2010/13/UE du Parlement et du conseil du 10 mars 2010 visant à la coordination de certaines dispositions législatives, réglementaires et administratives des États membres relatives à la fourniture de services de médias audiovisuels (directive «Services de médias audiovisuels»). Consultable en ligne : https://eur-lex.europa.eu/legal-content/FR/AUTO/?uri=celex:32010L0013. Dernier accès : 04/07/2022. [Cited on page 35.]

86-1067 (1986). Loi n° 86-1067 du 30 septembre 1986 relative à la liberté de communication (Loi Léotard). Consultable en ligne : https://www.legifrance.gouv.fr/loda/id/JORFTEXT000000512205/. Dernier accès : 04/07/2022. [Cited on page 35.]

Abdelali, A., Guzman, F., Sajjad, H., & Vogel, S. (2014). The AMARA corpus : Building parallel language resources for the educational domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, (pp. 1856–1862). Reykjavik, Iceland : European Language Resources Association (ELRA).
URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/877_Paper.pdf [Cited on page 47.]

Abdul Rauf, S., Ligozat, A.-L., Yvon, F., Illouz, G., & Hamon, T. (2020). Simplification automatique de texte dans un contexte de faibles ressources. In C. Benzitoun, C. Braud, L. Huber, D. Langlois, S. Ouni, S. Pogodalla, & S. Schneider (Eds.) *6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelles*, (pp. 332–341). Nancy, France : ATALA.
URL https://hal.archives-ouvertes.fr/hal-02784783 [Cited on page 28.]

Abend, O., & Rappoport, A. (2013). Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, (pp. 228–238). Sofia, Bulgaria : Association for Computational Linguistics.
URL https://aclanthology.org/P13-1023 [Cited on page 32.]

Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., & Goldberg, Y. (2016). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *CoRR*, *abs/1608.04207*.
URL http://arxiv.org/abs/1608.04207 [Cited on pages 53, 61, and 70.]

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., & Mohri, M. (2007). Openfst : A general and efficient weighted finite-state transducer library. In J. Holub, & J. Zdárek (Eds.) *Implementation and Application of Automata, 12th International Conference, CIAA 2007, Prague, Czech Republic, July 16-18, 2007, Revised Selected Papers*, vol. 4783 of *Lecture Notes in Computer Science*, (pp. 11–23). Springer.
URL https://doi.org/10.1007/978-3-540-76336-9_3 [Cited on page 62.]

Alonzo, O., Seita, M., Glasser, A., & Huenerfauth, M. (2020). *Automatic Text Simplification Tools for Deaf and Hard of Hearing Adults : Benefits of Lexical Simplification*

and Providing Users with Autonomy, (p. 1–13). New York, NY, USA : Association for Computing Machinery.
URL https://doi-org.ins2i.bib.cnrs.fr/10.1145/3313831.3376563 [Cited on pages 16 and 40.]

Alva-Manchego, F., Martin, L., Scarton, C., & Specia, L. (2019). EASSE : easier automatic sentence simplification evaluation. *CoRR*, *abs/1908.04567*.
URL http://arxiv.org/abs/1908.04567 [Cited on page 97.]

Alva-Manchego, F., Scarton, C., & Specia, L. (2019). Cross-sentence transformations in text simplification. In *Proceedings of the 2019 Workshop on Widening NLP*, (pp. 181–184). Florence, Italy : Association for Computational Linguistics.
URL https://aclanthology.org/W19-3656 [Cited on page 22.]

Álvarez, A., Balenciaga, M., del Pozo, A., Arzelus, H., Matamala, A., & Martínez-Hinarejos, C.-D. (2016). Impact of automatic segmentation on the quality, productivity and self-reported post-editing effort of intralingual subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, (pp. 3049–3053). Portorož, Slovenia : European Language Resources Association (ELRA).
URL https://aclanthology.org/L16-1487 [Cited on page 118.]

Anastasopoulos, A., Bojar, O., Bremerman, J., Cattoni, R., Elbayad, M., Federico, M., Ma, X., Nakamura, S., Negri, M., Niehues, J., Pino, J., Salesky, E., Stüker, S., Sudoh, K., Turchi, M., Waibel, A., Wang, C., & Wiesner, M. (2021). FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, (pp. 1–29). Bangkok, Thailand (online) : Association for Computational Linguistics.
URL https://aclanthology.org/2021.iwslt-1.1 [Cited on page 45.]

Anastasopoulos, A., & Chiang, D. (2018). Tied multitask learning for neural speech translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, (pp. 82–91). New Orleans, Louisiana : Association for Computational Linguistics.
URL https://aclanthology.org/N18-1008 [Cited on page 46.]

Angerbauer, K., Adel, H., & Vu, N. T. (2019). Automatic compression of subtitles with neural networks and its effect on user experience. In G. Kubin, & Z. Kacic (Eds.) *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, (pp. 594–598). ISCA.
URL https://doi.org/10.21437/Interspeech.2019-1750 [Cited on page 43.]

Aziz, W., de Sousa, S. C. M., & Specia, L. (2012). Cross-lingual sentence compression for subtitles. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, (pp. 103–110). Trento, Italy : European Association for Machine Translation.
URL https://www.aclweb.org/anthology/2012.eamt-1.33 [Cited on pages 43 and 52.]

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Y. Bengio, & Y. LeCun (Eds.) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
URL http://arxiv.org/abs/1409.0473 [Cited on pages 24 and 90.]

Bapna, A., & Firat, O. (2019). Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing, (pp. 1538–1548). Hong Kong, China. [Cited on pages 95 and 114.]

BBC (2021). Subtitling guidelines, version 1.1.9. Consultable en ligne : https://bbc.github.io/subtitle-guidelines/. [Cited on pages 40 and 116.]

Beeferman, D., Berger, A., & Lafferty, J. (1999). Statistical models for text segmentation. *Machine learning*, *34*(1-3), 177–210. [Cited on page 118.]

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, *3*, 1137–1155.
URL http://jmlr.org/papers/v3/bengio03a.html [Cited on page 62.]

Bentivogli, L., Cettolo, M., Gaido, M., Karakanta, A., Martinelli, A., Negri, M., & Turchi, M. (2021). Cascade versus direct speech translation : Do the differences still make a difference ? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, (pp. 2873–2887). Online : Association for Computational Linguistics.
URL https://aclanthology.org/2021.acl-long.224 [Cited on page 45.]

Bérard, A., Besacier, L., Kocabiyikoglu, A. C., & Pietquin, O. (2018). End-to-End Automatic Speech Translation of Audiobooks. In *ICASSP 2018 - IEEE International Conference on Acoustics, Speech and Signal Processing*. Calgary, Alberta, Canada.
URL https://hal.archives-ouvertes.fr/hal-01709586 [Cited on page 45.]

Bérard, A., Pietquin, O., Besacier, L., & Servan, C. (2016). Listen and Translate : A Proof of Concept for End-to-End Speech-to-Text Translation. In *NIPS Workshop on end-to-end learning for speech and audio processing*. Barcelona, Spain.
URL https://hal.archives-ouvertes.fr/hal-01408086 [Cited on pages 45 and 92.]

Bigand, F., Prigent, E., & Braffort, A. (2019). Animating virtual signers : The issue of gestural anonymization. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, IVA '19, (p. 252–255). New York, NY, USA : Association for Computing Machinery.
URL https://doi-org.ins2i.bib.cnrs.fr/10.1145/3308532.3329410 [Cited on page 35.]

Björnsson, C.-H. (1968). *Lesbarkeit durch LIX*. Pedagogiskt centrum, Stockholms skolförvaltn. [Cited on page 29.]

Blanche-Benveniste, C. (2010). *Approches de la langue parlée*. Éditions Ophrys. [Cited on pages 38, 39, and 87.]

Botha, J. A., Faruqui, M., Alex, J., Baldridge, J., & Das, D. (2018). Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (pp. 732–737). Brussels, Belgium : Association for Computational Linguistics.
URL https://aclanthology.org/D18-1080 [Cited on page 27.]

Braffort, A. (2016). *La Langue des Signes Française (LSF) : modélisations, ressources et applications*. ISTE Group. [Cited on page 41.]

Buet, F. (2020). Analyse de la régulation de la longueur dans un système neuronal de compression de phrase : une étude du modèle LenInit. In C. Benzitoun, C. Braud, L. Huber, D. Langlois, S. Ouni, S. Pogodalla, & S. Schneider (Eds.) *6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en*

Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, (pp. 57–70). Nancy, France : ATALA.
URL https://hal.archives-ouvertes.fr/hal-02786187 [Cited on page 19.]

Buet, F., & Yvon, F. (2021a). Toward Genre Adapted Closed Captioning. In *Interspeech 2021*, (pp. 4403–4407). Brno (virtual), Czech Republic : ISCA.
URL https://hal.archives-ouvertes.fr/hal-03329488 [Cited on pages 19 and 120.]

Buet, F., & Yvon, F. (2021b). Vers la production automatique de sous-titres adaptés à l'affichage. In P. Denis, N. Grabar, A. Fraisse, R. Cardon, B. Jacquemin, E. Kergosien, & A. Balvet (Eds.) *Traitement Automatique des Langues Naturelles*, (pp. 91–104). Lille, France : ATALA.
URL https://hal.archives-ouvertes.fr/hal-03265891 [Cited on page 19.]

Burlot, F., & Yvon, F. (2018). Using monolingual data in neural machine translation : a systematic study. In *Proceedings of the Third Conference on Machine Translation*, (pp. 144–155). Belgium, Brussels : Association for Computational Linguistics.
URL http://www.aclweb.org/anthology/W18-64015 [Cited on page 79.]

Campione, E., & Véronis, J. (2001). Etiquetage prosodique semi-automatique des corpus oraux. In *Actes de la 8ème conférence sur le Traitement Automatique des Langues Naturelles. Articles longs*, (pp. 122–131). Tours, France : ATALA.
URL https://aclanthology.org/2001.jeptalnrecital-long.10 [Cited on page 39.]

Carroll, J., Minnen, G., Pearce, D., Canning, Y., Devlin, S., & Tait, J. (1999). Simplifying text for language-impaired readers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*. Bergen, Norway : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/E99-1042 [Cited on pages 15 and 23.]

Carroll, M., & Ivarsson, J. (1998). *Code of Good Subtitling Practice*. Simrishamn : TransEdit. [Cited on page 116.]

Cendrars, B. (1926). *L'ABC du cinéma*. Les Écrivains réunis. [Cited on page 34.]

Cettolo, M., Girardi, C., & Federico, M. (2012). WIT3 : Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, (pp. 261–268). Trento, Italy : European Association for Machine Translation.
URL https://aclanthology.org/2012.eamt-1.60 [Cited on page 47.]

Chandrasekar, R., Doran, C., & Srinivas, B. (1996). Motivations and methods for text simplification. In *COLING 1996 Volume 2 : The 16th International Conference on Computational Linguistics*.
URL https://www.aclweb.org/anthology/C96-2183 [Cited on page 23.]

Chandrasekar, R., & Srinivas, B. (1997). Automatic induction of rules for text simplification. *Knowl. Based Syst.*, *10*(3), 183–190.
URL https://doi.org/10.1016/S0950-7051(97)00029-4 [Cited on page 23.]

Cherry, C., Arivazhagan, N., Padfield, D., & Krikun, M. (2021). Subtitle translation as markup translation. In H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, & P. Motlícek (Eds.) *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, (pp. 2237–2241). ISCA.
URL https://doi.org/10.21437/Interspeech.2021-744 [Cited on page 50.]

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014a). On the properties of neural machine translation : Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, (pp. 103–111). Association for Computational Linguistics.
URL http://aclweb.org/anthology/W14-4012 [Cited on page 90.]

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014b). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 1724–1734). Doha, Qatar : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/D14-1179 [Cited on pages 24, 54, and 62.]

Chu, C., & Wang, R. (2018). A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, COLING 2018, (pp. 1304–1319). Santa Fe, New Mexico, USA.
URL http://aclweb.org/anthology/C18-1111 [Cited on page 94.]

Cohn, T., & Lapata, M. (2008). Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics*, (COLING 2008), (pp. 137–144). Manchester, UK : Coling 2008 Organizing Committee.
URL http://www.aclweb.org/anthology/C08-1018 [Cited on page 21.]

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., & Baroni, M. (2018). What you can cram into a single $& !#* vector : Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, (pp. 2126–2136). Association for Computational Linguistics.
URL http://aclweb.org/anthology/P18-1198 [Cited on page 53.]

Corston-Oliver, S. (2001). Text compaction for display on very small screens. In *Proceedings of the NAACL Workshop on Automatic Summarization*, (pp. 89–98). Association for Computational Linguistics.
URL https://pdfs.semanticscholar.org/73c5/947cb8fae6f7d52755a2be237aa396aab45a.pdf [Cited on page 23.]

Coster, W., & Kauchak, D. (2011a). Learning to simplify sentences using wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, (pp. 1–9). Association for Computational Linguistics.
URL http://aclweb.org/anthology/W11-1601 [Cited on page 24.]

Coster, W., & Kauchak, D. (2011b). Simple English Wikipedia : A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, (pp. 665–669). Portland, Oregon, USA : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/P11-2117 [Cited on page 27.]

CSA (2011). Charte relative à la qualité du sous-titrage à destination des personnes sourdes ou malentendantes. Consultable en ligne : https://www.csa.fr/Reguler/Espace-juridique/Les-relations-de-l-Arcom-avec-les-editeurs/Chartes-et-autres-guides/Charte-relative-a-la-qualite-du-sous-titrage-a-destination-des-personnes-sourdes-ou-malentendantes-Decembre-2011. Dernier accès : 04/07/2022. [Cited on page 37.]

Daelemans, W., Höthker, A., & Tjong Kim Sang, E. (2004). Automatic sentence simplification for subtitling in Dutch and English. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal : European Language Resources Association (ELRA).
URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/697.pdf [Cited on

pages 16 and 43.]

Dale, E., & Chall, J. S. (1948). A formula for predicting readability : Instructions. *Educational Research Bulletin*, *27*(2), 37–54.
URL http://www.jstor.org/stable/1473669 [Cited on page 29.]

De Belder, J., & Moens, M.-F. (2010). Text simplification for children. In *Proceedings of the SIGIR workshop on accessible search systems*, (pp. 19–26). ACM ; New York. [Cited on page 15.]

de Loupy, C., Guégan, M., Ayache, C., Seng, S., & Moreno, J.-M. T. (2010). A french human reference corpus for multi-document summarization and sentence compression. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. European Languages Resources Association (ELRA).
URL http://www.lrec-conf.org/proceedings/lrec2010/pdf/919_Paper.pdf [Cited on page 28.]

Di Gangi, M. A., Cattoni, R., Bentivogli, L., Negri, M., & Turchi, M. (2019). MuST-C : a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, (pp. 2012–2017). Minneapolis, Minnesota : Association for Computational Linguistics.
URL https://aclanthology.org/N19-1202 [Cited on page 47.]

Diaz Cintas, J., & Remael, A. (2007). *Audiovisual Translation : Subtitling*. Translation practices explained. Routledge. [Cited on pages 16, 38, and 117.]

Edunov, S., Ott, M., Auli, M., & Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (pp. 489–500). Brussels, Belgium : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/D18-1045 [Cited on page 79.]

Elhadad, N., & Sutaria, K. (2007). Mining a lexicon of technical terms and lay equivalents. In *Biological, translational, and clinical language processing*, (pp. 49–56). Prague, Czech Republic : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/W07-1007 [Cited on pages 15 and 23.]

Etchegoyhen, T., Arzelus, H., Gete, H., Alvarez, A., Torre, I. G., Martín-Doñas, J. M., González-Docasal, A., & Fernandez, E. B. (2022). Cascade or direct speech translation ? a case study. *Applied Sciences*, *12*(3).
URL https://www.mdpi.com/2076-3417/12/3/1097 [Cited on page 45.]

Evans, R., Orăsan, C., & Dornescu, I. (2014). An evaluation of syntactic simplification rules for people with autism. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, (pp. 131–140). Gothenburg, Sweden : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/W14-1215 [Cited on page 15.]

Favre, B. (2007). *Résumé automatique de parole pour un accès efficace aux bases de données audio*. Theses, Université d'Avignon.
URL https://tel.archives-ouvertes.fr/tel-00444105 [Cited on page 22.]

Flesch, R. (1948). A new readability yardstick. *Journal of applied psychology*, *32*(3), 221. [Cited on page 52.]

Fournier, C. (2013). Evaluating text segmentation using boundary edit distance. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, (pp. 1702–1712). Sofia, Bulgaria : Association for Compu-

tational Linguistics.
URL https://aclanthology.org/P13-1167 [Cited on pages 119, 123, and 126.]

Fournier, C., & Inkpen, D. (2012). Segmentation similarity and agreement. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, (pp. 152–161). Montréal, Canada : Association for Computational Linguistics.
URL https://aclanthology.org/N12-1016 [Cited on page 119.]

Freitag, M., & Al-Onaizan, Y. (2016). Fast domain adaptation for neural machine translation. *CoRR*, *abs/1612.06897*.
URL http://arxiv.org/abs/1612.06897 [Cited on pages 90 and 94.]

Gadet, F. (1997). La variation, plus qu'une écume. *Langue française*, (115), 5–18. [Cited on page 39.]

Gadet, F. (2021). Variation. *Langage et societe*, (HS1), 331–336.
URL https://doi.org/10.3917/ls.hs01.0332 [Cited on page 39.]

Gala, N., Tack, A., Javourey-Drevet, L., François, T., & Ziegler, J. C. (2020). Alector : A Parallel Corpus of Simplified French Texts with Alignments of Misreadings by Poor and Dyslexic Readers. In *Language Resources and Evaluation for Language Technologies (LREC)*. Marseille, France.
URL https://hal.archives-ouvertes.fr/hal-02503986 [Cited on page 28.]

Ganitkevitch, J., Van Durme, B., & Callison-Burch, C. (2013). PPDB : The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, (pp. 758–764). Atlanta, Georgia : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/N13-1092 [Cited on page 28.]

Ghannay, S., Caubrière, A., Estève, Y., Camelin, N., Simonnet, E., Laurent, A., & Morin, E. (2018). End-to-end named entity and semantic concept extraction from speech. In *IEEE Spoken Language Technology Workshop*. Athens, Greece.
URL https://hal.archives-ouvertes.fr/hal-01987740 [Cited on page 92.]

Glickman, O., Dagan, I., Daelemans, W., Keller, M., & Bengio, S. (2006). Investigating lexical substitution scoring for subtitle generation. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, (pp. 45–52). New York City : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/W06-2907 [Cited on page 43.]

Gontier, F., Serizel, R., & Cerisara, C. (2021). Automated audio captioning by fine-tuning bart with audioset tags. In *DCASE 2021 - 6th Workshop on Detection and Classification of Acoustic Scenes and Events*. Virtual, Spain.
URL https://hal.inria.fr/hal-03522488 [Cited on page 46.]

Gorman, K. (2016). Pynini : A Python library for weighted finite-state grammar compilation. In *Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata*, (pp. 75–80). Berlin, Germany : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/W16-2409 [Cited on pages 62 and 77.]

Grabar, N., & Hamon, T. (2014). Automatic extraction of layman names for technical medical terms. In *2014 IEEE International Conference on Healthcare Informatics, ICHI 2014, Verona, Italy, September 15-17, 2014*, (pp. 310–319). IEEE Computer Society.
URL https://doi.org/10.1109/ICHI.2014.49 [Cited on pages 15 and 23.]

Graesser, A. C., McNamara, D. S., Louwerse, M. M., & Cai, Z. (2004). Coh-metrix : Analysis of text on cohesion and language. *Behavior research methods, instruments,*

& computers, *36*(2), 193–202.
URL https://doi.org/10.3758/BF03195564 [Cited on page 30.]

Graff, D., Kong, J., Chen, K., & Maeda, K. (2003). English gigaword. *Linguistic Data Consortium, Philadelphia*, *4*(1), 34. [Cited on page 27.]

Gunning, R., et al. (1952). Technique of clear writing. [Cited on page 29.]

Guzman, F., Sajjad, H., Vogel, S., & Abdelali, A. (2013). The AMARA corpus : building resources for translating the web's educational content. In *Proceedings of the 10th International Workshop on Spoken Language Translation : Papers*. Heidelberg, Germany.
URL https://aclanthology.org/2013.iwslt-papers.2 [Cited on page 47.]

Haeusler, L., DE LAVAL, T., Millot, C., & VON LENNEP, F. (2014). Etude quantitative sur le handicap auditif à partir de l'enquête handicap-santé. *SERIE ETUDES-DOCUMENT DE TRAVAIL-DREES*, (131). [Cited on page 35.]

Hamm, M. (2008). L'apprentissage de la lecture chez les enfants sourds. *Education & Formation*, (p. e 288).
URL https://hal.archives-ouvertes.fr/hal-00443706 [Cited on page 15.]

Hearst, M. A. (1997). Text tiling : Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, *23*(1), 33–64.
URL https://aclanthology.org/J97-1003 [Cited on page 117.]

Horn, C., Manduca, C., & Kauchak, D. (2014). Learning a lexical simplifier using Wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, (pp. 458–463). Baltimore, Maryland : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/P14-2075 [Cited on page 23.]

Hu, J. E., Rudinger, R., Post, M., & Durme, B. V. (2019). Parabank : Monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. *CoRR*, *abs/1901.03644*.
URL http://arxiv.org/abs/1901.03644 [Cited on page 28.]

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., & Xing, E. P. (2017). Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, (pp. 1587–1596). JMLR.org.
URL http://dl.acm.org/citation.cfm?id=3305381.3305545 [Cited on pages 26 and 70.]

Jansen, D., Alcala, A., & Guzmán, F. (2014). Amara : A sustainable, global solution for accessibility, powered by communities of volunteers. In C. Stephanidis, & M. Antona (Eds.) *Universal Access in Human-Computer Interaction. Design for All and Accessibility Practice - 8th International Conference, UAHCI 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014, Proceedings, Part IV*, vol. 8516 of *Lecture Notes in Computer Science*, (pp. 401–411). Springer.
URL https://doi.org/10.1007/978-3-319-07509-9_38 [Cited on page 47.]

Jensema, C. (1998). Viewer reaction to different television captioning speeds. *American annals of the deaf*, (pp. 318–324). [Cited on page 41.]

Jia, Y., Johnson, M., Macherey, W., Weiss, R. J., Cao, Y., Chiu, C.-C., Ari, N., Laurenzo, S., & Wu, Y. (2019). Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (pp. 7180–7184). [Cited on page 45.]

Jiang, C., Maddela, M., Lan, W., Zhong, Y., & Xu, W. (2020). Neural CRF model for

sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (pp. 7943–7960). Online : Association for Computational Linguistics.
URL https://aclanthology.org/2020.acl-main.709 [Cited on page 27.]

John, V., Mou, L., Bahuleyan, H., & Vechtomova, O. (2019). Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (pp. 424–434). Florence, Italy : Association for Computational Linguistics.
URL https://aclanthology.org/P19-1041 [Cited on page 70.]

Kalchbrenner, N., & Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, (pp. 1700–1709). Seattle, Washington, USA : Association for Computational Linguistics.
URL https://aclanthology.org/D13-1176 [Cited on pages 24 and 54.]

Kandel, L., & Moles, A. (1958). Application de l'indice de Flesch à la langue française. *Cahiers Etudes de Radio-Télévision*, *19*(1958), 253–274. [Cited on page 30.]

Karakanta, A., Buet, F., Cettolo, M., & Yvon, F. (2022). Evaluating subtitle segmentation for end-to-end generation systems. In *Proceedings of the Language Resources and Evaluation Conference*, (pp. 3069–3078). Marseille, France : European Language Resources Association.
URL https://aclanthology.org/2022.lrec-1.328 [Cited on pages 19 and 115.]

Karakanta, A., Negri, M., & Turchi, M. (2020a). Is 42 the answer to everything in subtitling-oriented speech translation ? In *Proceedings of the 17th International Conference on Spoken Language Translation*, (pp. 209–219). Online : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/2020.iwslt-1.26 [Cited on pages 45, 48, 49, 73, 92, 97, 115, 119, 120, and 122.]

Karakanta, A., Negri, M., & Turchi, M. (2020b). MuST-cinema : a speech-to-subtitles corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, (pp. 3727–3734). Marseille, France : European Language Resources Association.
URL https://www.aclweb.org/anthology/2020.lrec-1.460 [Cited on pages 42, 48, 73, 78, 97, 116, and 123.]

Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., & Okumura, M. (2016). Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (pp. 1328–1338). Association for Computational Linguistics.
URL http://aclweb.org/anthology/D16-1140 [Cited on pages 25, 31, 52, 54, and 68.]

Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L., & Chissom, B. S. (1975). Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. *Research Branch Report*, (pp. 8–75). [Cited on page 52.]

Kingma, D. P., & Ba, J. (2015). Adam : A method for stochastic optimization. In Y. Bengio, & Y. LeCun (Eds.) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
URL http://arxiv.org/abs/1412.6980 [Cited on pages 62 and 93.]

Knight, K., & Marcu, D. (2000). Statistics-based summarization - step one : Sentence compression. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*,

(pp. 703–710). AAAI Press.
URL http://dl.acm.org/citation.cfm?id=647288.721086 [Cited on page 21.]

Kobus, C., Crego, J., & Senellart, J. (2017). Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, (pp. 372–378). Varna, Bulgaria.
URL https://doi.org/10.26615/978-954-452-049-6_049 [Cited on pages 90 and 94.]

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., & Herbst, E. (2007). Moses : Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, (pp. 177–180). Prague, Czech Republic : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/P07-2045 [Cited on page 24.]

Kudo, T., & Richardson, J. (2018). SentencePiece : A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, (pp. 66–71). Brussels, Belgium : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/D18-2012 [Cited on page 93.]

Kushalnagar, P., Smith, S., Hopper, M., Ryan, C., Rinkevich, M., & Kushalnagar, R. (2018). Making cancer health text on the internet easier to read for deaf people who use american sign language. *Journal of Cancer Education*, *33*(1), 134–140.
URL https://doi.org/10.1007/s13187-016-1059-5 [Cited on page 40.]

Kwolek, W. F. (1973). A readability survey of technical and popular literature. *Journalism Quarterly*, *50*(2), 255–264.
URL https://doi.org/10.1177/107769907305000206 [Cited on page 29.]

Lakew, S. M., Gangi, M. D., & Federico, M. (2019). Controlling the output length of neural machine translation. In *Proceedings of IWSLT'2019*. [Cited on pages 56, 68, and 115.]

Laks, S. (1957). *Le sous-titrage de films : sa technique, son esthétique*. Propriété de l'Auteur. [Cited on page 36.]

Lample, G., Subramanian, S., Smith, E. M., Denoyer, L., Ranzato, M., & Boureau, Y. (2019). Multiple-attribute text rewriting. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
URL https://openreview.net/forum?id=H1g2NhC5KQ [Cited on page 27.]

Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., & Ranzato, M. (2017). Fader networks : Manipulating images by sliding attributes. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, & R. Garnett (Eds.) *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, (pp. 5969–5978).
URL http://papers.nips.cc/paper/7178-fader-networksmanipulating-images-by-sliding-attributes [Cited on page 26.]

Lin, C.-Y. (2004). ROUGE : A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, (pp. 74–81). Barcelona, Spain : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/W04-1013 [Cited on page 31.]

Lison, P., & Tiedemann, J. (2016). OpenSubtitles2016 : Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, (pp. 923–929). Portorož, Slovenia : European Language Resources Association (ELRA).
URL https://aclanthology.org/L16-1147 [Cited on page 46.]

Liu, D., Fu, J., Zhang, Y., Pal, C., & Lv, J. (2019). Revision in continuous space : Fine-grained control of text style transfer. *CoRR*, *abs/1905.12304*.
URL http://arxiv.org/abs/1905.12304 [Cited on page 26.]

Liu, D., Niehues, J., & Spanakis, G. (2020). Adapting end-to-end speech recognition for readable subtitles. In *Proceedings of the 17th International Conference on Spoken Language Translation*, (pp. 247–256). Online : Association for Computational Linguistics.
URL https://aclanthology.org/2020.iwslt-1.30 [Cited on pages 22, 45, and 115.]

Luong, M.-T., & Manning, C. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation : Evaluation Campaign*, (pp. 76–79). Da Nang, Vietnam.
URL https://aclanthology.org/2015.iwslt-evaluation.11 [Cited on page 94.]

Macaire, C., Ormaechea-Grijalba, L., & Pupier, A. (2022). Une chaîne de traitements pour la simplification automatique de la parole et sa traduction automatique vers des pictogrammes (simplification and automatic translation of speech into pictograms). In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 2 : 24e Rencontres Etudiants Chercheurs en Informatique pour le TAL (RECITAL)*, (pp. 111–123). Avignon, France : ATALA.
URL https://aclanthology.org/2022.jeptalnrecital-recital.9 [Cited on page 22.]

Mallinson, J., Sennrich, R., & Lapata, M. (2018). Sentence compression for arbitrary languages via multilingual pivoting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (pp. 2453–2464). Association for Computational Linguistics.
URL http://aclweb.org/anthology/D18-1267 [Cited on page 28.]

Marleau, L. (1982). Les sous-titres... un mal nécessaire. *Meta : journal des traducteurs/Meta : translators' Journal*, *27*(3), 271–285. [Cited on pages 34 and 39.]

Martin, L. (2021). *Automatic sentence simplification using controllable and unsupervised methods*. Theses, Sorbonne Université.
URL https://tel.archives-ouvertes.fr/tel-03543971 [Cited on pages 22 and 29.]

Martin, L., de la Clergerie, É., Sagot, B., & Bordes, A. (2020). Controllable sentence simplification. In *Proceedings of the 12th Language Resources and Evaluation Conference*, (pp. 4689–4698). Marseille, France : European Language Resources Association.
URL https://www.aclweb.org/anthology/2020.lrec-1.577 [Cited on page 25.]

Matusov, E., Leusch, G., Bender, O., & Ney, H. (2005). Evaluating machine translation output with automatic sentence segmentation. In *Proceedings of the Second International Workshop on Spoken Language Translation*. Pittsburgh, Pennsylvania, USA.
URL https://aclanthology.org/2005.iwslt-1.19 [Cited on page 122.]

Matusov, E., Mauser, A., & Ney, H. (2006). Automatic sentence segmentation and punctuation prediction for spoken language translation. In *Proceedings of the Third International Workshop on Spoken Language Translation : Papers*. Kyoto, Japan.
URL https://aclanthology.org/2006.iwslt-papers.1 [Cited on page 45.]

Matusov, E., Wilken, P., & Georgakopoulou, Y. (2019). Customizing neural machine translation for subtitling. In *Proceedings of the Fourth Conference on Machine Trans-*

lation (Volume 1 : Research Papers), (pp. 82–93). Florence, Italy.
URL https://www.aclweb.org/anthology/W19-5209 [Cited on pages 43, 49, 73, 92, and 97.]

Mc Laughlin, G. H. (1969). Smog grading : A new readability formula. *Journal of reading*, *12*(8), 639–646. [Cited on pages 29 and 52.]

McDonald, R. (2006). Discriminative sentence compression with soft syntactic evidence. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, EACL 06, (pp. 297–304). Trento, Italy.
URL http://www.aclweb.org/anthology/E06-1038 [Cited on page 21.]

Milde, B., Geislinger, R., Lindt, I., & Baumann, T. (2021). Open source automatic lecture subtitling. In *Proceedings of Elektronische Sprachsignalverarbeitung (ESSV), Tagungsband der 32. Konferenz, Berlin, 3.-5. März 2021*, (pp. 128 – 135).
URL https://www.essv.de/pdf/2021_128_135.pdf?id=1109 [Cited on page 43.]

Mohri, M. (2002). Semiring frameworks and algorithms for shortest-distance problems. *J. Autom. Lang. Comb.*, *7*(3), 321–350.
URL https://doi.org/10.25596/jalc-2002-321 [Cited on pages 60 and 77.]

Napoles, C., Gormley, M., & Van Durme, B. (2012). Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction*, AKBC-WEKEX '12, (p. 95–100). USA : Association for Computational Linguistics. [Cited on page 27.]

Napoles, C., Van Durme, B., & Callison-Burch, C. (2011). Evaluating sentence compression : Pitfalls and suggested remedies. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, (pp. 91–97). Stroudsburg, PA, USA : Association for Computational Linguistics.
URL http://dl.acm.org/citation.cfm?id=2107679.2107690 [Cited on page 29.]

Narayan, S., & Gardent, C. (2014). Hybrid simplification using deep semantics and machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, (pp. 435–445). Association for Computational Linguistics.
URL http://aclweb.org/anthology/P14-1041 [Cited on page 24.]

Netflix (2021). Timed text style guide : General requirements. Consultable en ligne : https://partnerhelp.netflixstudios.com/hc/en-us/articles/215758617-Timed-Text-Style-Guide-General-Requirements. Dernier accès : 04/07/2022. [Cited on page 116.]

Ng, J.-P., & Abrecht, V. (2015). Better summarization evaluation with word embeddings for ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (pp. 1925–1930). Lisbon, Portugal : Association for Computational Linguistics.
URL https://aclanthology.org/D15-1222 [Cited on page 31.]

Okabe, S., Yvon, F., & Besacier, L. (2021). Segmentation en mots faiblement supervisée pour la documentation automatique des langues.
URL https://hal.archives-ouvertes.fr/hal-03477475 [Cited on page 117.]

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU : a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, (pp. 311–318). Stroudsburg, PA, USA : Association for Computational Linguistics.
URL http://dx.doi.org/10.3115/1073083.1073135 [Cited on page 30.]

Pavlick, E., & Callison-Burch, C. (2016). Simple PPDB : A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, (pp. 143–148). Association for Computational Linguistics.
URL http://aclweb.org/anthology/P16-2024 [Cited on page 28.]

Pavlick, E., Rastogi, P., Ganitkevitch, J., Van Durme, B., & Callison-Burch, C. (2015). PPDB 2.0 : Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers)*, (pp. 425–430). Beijing, China : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/P15-2070 [Cited on page 28.]

Perego, E. (2008). Subtitles and line-breaks : Towards improved readability. *Between Text and Image : Updating research in screen translation*, *78*(1), 211–223.
URL https://doi.org/10.1075/btl.78.21per [Cited on page 115.]

Petersen, S. E., & Ostendorf, M. (2007). Text simplification for language learners : a corpus analysis. In *Workshop on Speech and Language Technology in Education, SLaTE 2007, Farmington, PA, USA, October 1-3, 2007*, (pp. 69–72). Carnegie Mellon University / ISCA.
URL http://www.isca-speech.org/archive/slate_2007/sle7_069.html [Cited on page 15.]

Pevzner, L., & Hearst, M. A. (2002). A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, *28*(1), 19–36.
URL https://aclanthology.org/J02-1002 [Cited on pages 119 and 126.]

Pham, M. Q., Crego, J., & Yvon, F. (2021). Revisiting multi-domain machine translation. *Transactions of the Association for Computational Linguistics*, *9*(0), 17–35.
URL https://transacl.org/index.php/tacl/article/view/2327 [Cited on pages 94 and 101.]

Piperidis, S., Demiros, I., Prokopidis, P., Vanroose, P., Hoethker, A., Daelemans, W., Sklavounou, E., Konstantinou, M., & Karavidas, Y. (2004). Multimodal, multilingual resources in the subtitling process. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal : European Language Resources Association (ELRA).
URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/680.pdf [Cited on page 43.]

Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation : Research Papers*, (pp. 186–191). Belgium, Brussels : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/W18-6319 [Cited on pages 97 and 123.]

Prokopidis, P., Karra, V., Papagianopoulou, A., & Piperidis, S. (2008). Condensing sentences for subtitle generation. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association.
URL http://www.lrec-conf.org/proceedings/lrec2008/summaries/328.html [Cited on page 43.]

Rajendran, D. J., Duchowski, A. T., Orero, P., Martínez, J., & Romero-Fresco, P. (2013). Effects of text chunking on subtitling : A quantitative and qualitative examination. *Perspectives*, *21*(1), 5–21.
URL https://doi.org/10.1080/0907676X.2012.722651 [Cited on page 115.]

Rello, L., Baeza-Yates, R., Bott, S., & Saggion, H. (2013). Simplify or help ? text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, W4A '13. New York, NY, USA : Association for Computing Machinery.
URL https://doi.org/10.1145/2461121.2461126 [Cited on page 15.]

Romero-Fresco, P. (2009). More haste less speed : Edited versus verbatim respoken subtitles. *Vigo International Journal of Applied Linguistics*, (6), 109–133. [Cited on page 40.]

Rousseau, A., Deléglise, P., & Estève, Y. (2012). TED-LIUM : an automatic speech recognition dedicated corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, (pp. 125–129). Istanbul, Turkey : European Language Resources Association (ELRA).
URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/698_Paper.pdf [Cited on page 47.]

Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (pp. 379–389). Association for Computational Linguistics.
URL http://aclweb.org/anthology/D15-1044 [Cited on pages 24 and 27.]

Sanabria, R., Caglayan, O., Palaskar, S., Elliott, D., Barrault, L., Specia, L., & Metze, F. (2018). How2 : A Large-scale Dataset for Multimodal Language Understanding. In *NeurIPS*. Montréal, Canada.
URL https://hal.archives-ouvertes.fr/hal-02431947 [Cited on page 47.]

Saunders, D. (2021). Domain adaptation and multi-domain adaptation for neural machine translation : A survey. *CoRR*, *abs/2104.06951*.
URL https://arxiv.org/abs/2104.06951 [Cited on page 94.]

Schlippe, T., Alessai, S., El-Taweel, G., Wölfel, M., & Zaghouani, W. (2020). Visualizing voice characteristics with type design in closed captions for arabic. In A. Sourin, C. Charrier, C. Rosenberger, & O. Sourina (Eds.) *International Conference on Cyberworlds, CW 2020, Caen, France, September 29 - October 1, 2020*, (pp. 196–203). IEEE.
URL https://doi.org/10.1109/CW49994.2020.00039 [Cited on page 39.]

Schmidhuber, J., & Heil, S. (1996). Sequential neural text compression. *IEEE Trans. Neural Networks*, *7*(1), 142–146.
URL https://doi.org/10.1109/72.478398 [Cited on page 57.]

Sennrich, R., Haddow, B., & Birch, A. (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, (pp. 86–96). Berlin, Germany : Association for Computational Linguistics.
URL http://www.aclweb.org/anthology/P16-1009 [Cited on page 79.]

Shardlow, M. (2014). A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, *4*(1), 58–70. [Cited on page 29.]

Shi, X., Knight, K., & Yuret, D. (2016). Why neural translations are the right length. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (pp. 2278–2282). Austin, Texas : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/D16-1248 [Cited on pages 53 and 61.]

Siddharthan, A. (2014). A survey of research on text simplification. *International Journal of Applied Linguistics*, *165(2)*, 259–298.
URL http://homepages.abdn.ac.uk/advaith/pages/survey.pdf [Cited on page 15.]

Siddharthan, A., & Katsos, N. (2010). Reformulating discourse connectives for non-expert readers. In *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, (p. 1002–1010). USA : Association for Computational Linguistics. [Cited on page 15.]

Siddharthan, A., Nenkova, A., & McKeown, K. (2004). Syntactic simplification for improving content selection in multi-document summarization. In *COLING 2004 : Proceedings of the 20th International Conference on Computational Linguistics*, (pp. 896–902). Geneva, Switzerland : COLING.
URL https://www.aclweb.org/anthology/C04-1129 [Cited on page 23.]

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., & Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the seventh conference of the Association for Machine Translation in the Americas (AMTA)*, vol. 200, (pp. 223–231). Boston, Massachusetts, USA : Cambridge, MA.
URL https://www.mt-archive.net/AMTA-2006-Snover.pdf [Cited on pages 49 and 119.]

Specia, L. (2010). Translating from complex to simplified sentences. *Lecture Notes in Computer Science*, *6001*, 30–39. [Cited on page 24.]

Specia, L., Frank, S., Sima'an, K., & Elliott, D. (2016). A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation : Volume 2, Shared Task Papers*, (pp. 543–553). Association for Computational Linguistics.
URL http://aclweb.org/anthology/W16-2346 [Cited on page 46.]

Specia, L., Jauhar, S. K., & Mihalcea, R. (2012). SemEval-2012 task 1 : English lexical simplification. In **SEM 2012 : The First Joint Conference on Lexical and Computational Semantics – Volume 1 : Proceedings of the main conference and the shared task, and Volume 2 : Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, (pp. 347–355). Montréal, Canada : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/S12-1046 [Cited on page 23.]

Sperber, M., Niehues, J., & Waibel, A. (2017). Toward robust neural machine translation for noisy input sequences. In *Proceedings of the 14th International Conference on Spoken Language Translation*, (pp. 90–96). Tokyo, Japan : International Workshop on Spoken Language Translation.
URL https://aclanthology.org/2017.iwslt-1.13 [Cited on page 44.]

Sperber, M., & Paulik, M. (2020). Speech translation and the end-to-end promise : Taking stock of where we are. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (pp. 7409–7421). Online : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/2020.acl-main.661 [Cited on pages 42, 43, and 45.]

Sulem, E., Abend, O., & Rappoport, A. (2018a). BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing"*, (pp. 738–744). Brussels, Belgium : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/D18-1081 [Cited on page 31.]

Sulem, E., Abend, O., & Rappoport, A. (2018b). Semantic structural evaluation for text simplification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, (pp. 685–696). New Orleans, Louisiana : Association for Computational Linguistics.

URL http://www.aclweb.org/anthology/N18-1063 [Cited on page 32.]

Sulubacak, U., Caglayan, O., Grönroos, S., Rouhe, A., Elliott, D., Specia, L., & Tiedemann, J. (2020). Multimodal machine translation through visuals and speech. *Mach. Transl.*, *34*(2-3), 97–147.
URL https://doi.org/10.1007/s10590-020-09250-0 [Cited on pages 42 and 45.]

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.) *Advances in Neural Information Processing Systems 27*, (pp. 3104–3112). Curran Associates, Inc.
URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf [Cited on pages 24 and 54.]

Takase, S., & Okazaki, N. (2019). Positional encoding to control output sequence length. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, (pp. 3999–4004). Minneapolis, Minnesota : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/N19-1401 [Cited on pages 25, 31, 52, 56, and 68.]

Tapu, R., Mocanu, B., & Zaharia, T. B. (2019). Dynamic subtitles : A multimodal video accessibility enhancement dedicated to deaf and hearing impaired users. In *2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019*, (pp. 2558–2566). IEEE.
URL https://doi.org/10.1109/ICCVW.2019.00313 [Cited on page 46.]

TED (2021). Subtitling tips. Consultable en ligne : https://www.ted.com/participate/translate/subtitling-tips. [Cited on pages 37 and 116.]

Telles, S. V. (2018). *Readability and understandability of notes to the financial statements*. Ph.D. thesis, Universidade de São Paulo. [Cited on pages 20 and 29.]

Torres Monreal, S., & Santana Hernández, R. (2005). Reading levels of spanish deaf students. *American Annals of the Deaf*, *150*(4), 379–387. [Cited on pages 15 and 40.]

Traxler, C. B. (2000). The Stanford Achievement Test, 9th Edition : National Norming and Performance Standards for Deaf and Hard-of-Hearing Students. *The Journal of Deaf Studies and Deaf Education*, *5*(4), 337–348.
URL https://doi.org/10.1093/deafed/5.4.337 [Cited on page 15.]

Turner, J., & Charniak, E. (2005). Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, (pp. 290–297). Ann Arbor, Michigan : Association for Computational Linguistics.
URL http://www.aclweb.org/anthology/P05-1036 [Cited on page 21.]

UE 2016/2102 (2016). Directive (UE) 2016/2102 du Parlement et du conseil du 26 octobre 2016 relative à l'accessibilité des sites internet et des applications mobiles des organismes du secteur public. Consultable en ligne : http://eur-lex.europa.eu/legal-content/FR/TXT/?uri=celex:32016L2102. Dernier accès : 04/07/2022. [Cited on page 35.]

UNESCO-UIL (2017). 3e rapport mondial sur l'apprentissage et l'éducation des adultes : l'impact de l'apprentissage et l'éducation des adultes sur la santé et le bien-être, l'emploi et le marché du travail, et la vie sociale, civique et communautaire (grale iii). Consultable en ligne : https://unesdoc.unesco.org/ark:/48223/pf0000246943. [Cited on page 20.]

Vandeghinste, V., & Pan, Y. (2004). Sentence compression for automated subtitling : A hybrid approach. In *Text Summarization Branches Out*, (pp. 89–95). Barcelona, Spain : Association for Computational Linguistics.
URL https://aclanthology.org/W04-1015 [Cited on page 43.]

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, & R. Garnett (Eds.) *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, (pp. 6000–6010).
URL http://papers.nips.cc/paper/7181-attention-is-all-you-need [Cited on pages 25, 55, 90, and 93.]

Vickrey, D., & Koller, D. (2008). Sentence simplification for semantic role labeling. In *Proceedings of ACL-08 : HLT*, (pp. 344–352). Columbus, Ohio : Association for Computational Linguistics.
URL http://www.aclweb.org/anthology/P/P08/P08-1040 [Cited on page 23.]

Wieczorek, A., Kłyszejko, Z., Szarkowska, A., & Krejtz, I. (2011). Accessibility of subtitling for the hearing-impaired. In *Interfejs użytkownika - Kansei w praktyce, Warszawa 2011*, (pp. 27–33). Warsaw : Wydawnictwo PJWSTK. [Cited on page 41.]

Wilken, P., Georgakopoulou, P., & Matusov, E. (2022). SubER - a metric for automatic evaluation of subtitle quality. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, (pp. 1–10). Dublin, Ireland (in-person and online) : Association for Computational Linguistics.
URL https://aclanthology.org/2022.iwslt-1.1 [Cited on pages 49 and 50.]

Williams, H., & Thorne, D. (2000). The value of teletext subtitling as a medium for language learning. *System*, *28*(2), 217–228.
URL https://www.sciencedirect.com/science/article/pii/S0346251X00000087 [Cited on page 40.]

Woodsend, K., & Lapata, M. (2011). Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, (pp. 409–420). Edinburgh, Scotland, UK. : Association for Computational Linguistics.
URL https://aclanthology.org/D11-1038 [Cited on pages 21, 22, and 27.]

Wu, Z., Ive, J., Wang, J., Madhyastha, P., & Specia, L. (2019). Predicting actions to help predict translations. *CoRR*, *abs/1908.01665*.
URL http://arxiv.org/abs/1908.01665 [Cited on page 46.]

Wubben, S., van den Bosch, A., & Krahmer, E. (2012). Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, (pp. 1015–1024). Association for Computational Linguistics.
URL http://aclweb.org/anthology/P12-1107 [Cited on page 24.]

Wüest, J. (2009). La notion de diamésie est-elle nécessaire ? *Travaux de linguistique*, (2), 147–162. [Cited on page 38.]

Xu, J., Buet, F., Crego, J., Bertin-Lemée, E., & yvon, F. (2022). Joint Generation of Captions and Subtitles with Dual Decoding. In *19th International Conference on Spoken Language Translation (IWSLT 2022)*. Dublin, Ireland.
URL https://hal.archives-ouvertes.fr/hal-03666567 [Cited on page 19.]

Xu, J., & Yvon, F. (2021). One source, two targets : Challenges and rewards of dual

decoding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, (pp. 8533–8546). Online and Punta Cana, Dominican Republic : Association for Computational Linguistics.
URL https://aclanthology.org/2021.emnlp-main.671 [Cited on page 46.]

Xu, W., Callison-Burch, C., & Napoles, C. (2015). Problems in current text simplification research : New data can help. *Transactions of the Association for Computational Linguistics*, *3*, 283–297.
URL http://aclweb.org/anthology/Q15-1021 [Cited on page 27.]

Xu, W., Napoles, C., Pavlick, E., Chen, Q., & Callison-Burch, C. (2016). Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, *4*, 401–415.
URL http://aclweb.org/anthology/Q16-1029 [Cited on page 31.]

Yang, Z., Hirasawa, T., Komachi, M., & Okazaki, N. (2022). Why videos do not guide translations in video-guided machine translation ? an empirical evaluation of video-guided machine translation dataset. *J. Inf. Process.*, *30*, 388–396.
URL https://doi.org/10.2197/ipsjjip.30.388 [Cited on page 46.]

Zhang, X., & Lapata, M. (2017). Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, (pp. 584–594). Copenhagen, Denmark : Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/D17-1062 [Cited on pages 25, 27, 73, and 92.]

Zhang, Y., Ye, Z., Feng, Y., Zhao, D., & Yan, R. (2017). A constrained sequence-to-sequence neural model for sentence simplification. *CoRR*, *abs/1704.02312*.
URL http://arxiv.org/abs/1704.02312 [Cited on pages 25, 73, and 92.]

Zhu, Z., Bernhard, D., & Gurevych, I. (2010). A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, (pp. 1353–1361). Beijing, China : Coling 2010 Organizing Committee.
URL https://www.aclweb.org/anthology/C10-1152 [Cited on page 27.]

Annexe A

Calcul de $\text{BLEU}_{\text{br}}^{+}$

Nous détaillons dans cette annexe le calcul de $\text{BLEU}_{\text{br}}^{+}$, qui est partiellement expliqué à la section 7.3.2. Pour rappel, p_1, p_2, p_3, p_4 dénotent respectivement les précisions modifiées unigramme, bigramme, trigramme et quadrigramme calculées par BLEU_{nb}, et $p'_1, \ldots, p'_4$ désignent les précisions modifiées correspondantes pour BLEU_{br}. Notons de plus l le nombre de mots ordinaires dans la phrase prédite, L le nombre total d'unités (balises et mots), et α le rapport entre le nombre de balises et le nombre de mots ordinaires :

$$\alpha = \frac{L-l}{l} \quad \text{et} \quad L = l \times (1+\alpha)$$

Nous faisons les hypothèses suivantes :

1. le système produit un taux de balises α proche de celui de la référence (ce que nous avons pu vérifier en pratique avec nos modèles), de telle sorte que chaque frontière dans la prédiction pourrait être associée à une frontière dans la référence ;
2. les phrases sont suffisamment longues pour ignorer les effets de bords (c.-à-d. la diminution du nombre de n-grammes dans la phrase pour $n > 1$) ;
3. les n-grammes ne contiennent au plus qu'une balise chacun (le contraire signifierait que certaines lignes de sous-titres comportent moins de deux mots, ce qui est rare).

Dans ce qui va suivre, nous allons exprimer des majorants pour les précisions modifiées de BLEU_{br} ; cette majoration correspond à un scénario idéal dans lequel le contenu textuel engendré par le système n'est pas altéré, alors que la segmentation est réalisée de façon optimale.

1-gramme	**Nombre**	**Précision**
`X`	$l \times 1$	p_1
`B`	$l \times \alpha$	1

TABLE A.1 – Nombre et précision des types d'unigramme dans la prédiction segmentée. `X` représente un mot ordinaire quelconque, et `B` une balise.

Le tableau A.1 donne le nombre et la précision moyenne des types d'unigramme (mot

`X` ou balise `B`) dans la phrase segmentée prédite : du fait de notre première hypothèse la précision pour les balises est de 1, tandis que la précision moyenne des mots est p_1 (composante de BLEU_{nb}). En pondérant la précision des unigrammes par leur nombre, nous pouvons ainsi calculer facilement la précision unigramme de BLEU_{br} (qui tient compte des mot et des balises) :

$$p'_1 = p'^{+}_1 = \frac{p_1 + \alpha}{1 + \alpha} \tag{A.1}$$

2-gramme	**Nombre**	**Précision**
`X X`	$l \times (1-\alpha)$	p_2
`X B`	$l \times \alpha$	$\cdot < p_1$
`B X`	$l \times \alpha$	$\cdot < p_1$

TABLE A.2 – Nombre et précision des types de bigramme dans la prédiction segmentée. `X` représente un mot ordinaire quelconque, et `B` une balise.

Selon le même principe, le tableau A.2 donne le nombre et la précision moyenne des types de bigramme. `X B` ne pouvant être correct que si `X` l'est également, la précision pour ce type de bigramme est évidemment inférieure à la précision des unigrammes `X`, c'est-à-dire p_1. Le cas d'égalité correspond au scénario dans lequel les frontières sont disposées de façon optimale compte tenu du texte prédit. En combinant les précisions maximales et les nombres, il est alors possible de déterminer une borne supérieure pour la précision bigramme de BLEU_{br} :

$$p'_2 < p'^{+}_2 = \frac{(1-\alpha) \times p_2 + 2\alpha \times p_1}{1 + \alpha} \tag{A.2}$$

De manière parfaitement semblable, les équations (A.3) et (A.4) peuvent être déduites dans le cas des trigrammes et des quadrigrammes :

$$p'_3 < p'^{+}_3 = \frac{(1-2\alpha) \times p_3 + 3\alpha \times p_2}{1 + \alpha} \tag{A.3}$$

$$p'_4 < p'^{+}_4 = \frac{(1-3\alpha) \times p_4 + 4\alpha \times p_3}{1 + \alpha} \tag{A.4}$$

$\text{BLEU}_{\text{br}}{}^{+}$ peut alors être obtenu en remplaçant $p'_1, \ldots, p'_4$ par $p'^{+}_1, \ldots, p'^{+}_4$ dans le calcul de BLEU_{br} :

$$\text{BLEU}_{\text{br}}{}^{+} = \text{BP}' \times \exp\left(\sum_{n=1}^{N} w_n \log p'^{+}_n\right) \tag{A.5}$$