Controllable and Document-Level Text Simplification

Liam Cripwell

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Controllable and Document-Level Text Simplification

THÈSE

présentée et soutenue publiquement le 10 Novembre 2023

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Liam Cripwell

Composition du jury

Président :	Benoît Sagot	Directeur de recherche, Inria, France
Rapporteurs :	Benoît Sagot	
	Benoit Favre	Professeur, Aix-Marseille Université, France
Examinateurs :	Wei Xu	Assistant Professor, Georgia Institute of Technology, USA
	Liana Ermakova	Maître de conférences, Université de Bretagne Occidentale, France
Directeur de thèse :	Claire Gardent	Directrice de recherche, CNRS, LORIA, France
Co-directeur de thèse :	Joël Legrand	Maître de conférences, CentraleSupélec, LORIA, France

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Acknowledgments

The work presented in this thesis was made possible by the supporting project, Chaire IA XNLG "Multi-lingual, Multi-Source Text Generation, funded by the French National Research Agency (Gardent; award ANR-20-CHIA-0003"), the Région Grand-Est and Meta.

I would also like to thank the members of the jury for taking the time to participate in my examination. I appreciate the interest shown in my work and the opportunity for valuable feedback and criticism.

To my supervisors, Claire and Joël, thank you for pointing me in the right direction and guiding me to build confidence in my research abilities. Claire, I am truly grateful for the amount of time you were willing to put into discussions and answering my questions even with so many other students and responsibilities to take care of. Working with you has been a wonderful experience.

To the other students and members of Synalp, thank you for the little discussions in between lunch each day and our prolonged tea breaks. Having so many others going through the same experience over these years has helped to stay on track and keep things in perspective. A special thanks to Juliette for proofreading an early version of this thesis.

To those who helped to mentor and guide me during my previous studies and early career: Jim, Sazzad, Bevan, Guido, Dan, Iain, Jaco, and others. Thank you for taking the time to impart some of your knowledge onto me and providing the opportunities and guidance to build my skills and confidence.

To my friends back home, your continued friendship despite our distance means a lot to me. You boys keep me laughing even on the hard days.

To my parents, thank you for all you have done for me over the years and the sacrifices you have made to provide me with more opportunities. I would have never made it this far without the values you instilled in me and the guidance you continue to provide.

A Gilbert et Marcelle, je vous remercie infiniment pour votre gentillesse et votre hospitalité tout au long de mon séjour en France. Vous avez rendu ces dernières années d'adaptation à la vie en France beaucoup plus confortables. Votre volonté d'accueillir cet étrange étranger dans votre famille signifie beaucoup pour moi et c'est quelque chose que je n'oublierai pas.

Et bien sûr, à Maud, pour t'avoir à mes côtés chaque jour afin de m'aider à rester sain d'esprit et à prendre le temps d'apprécier les petites choses. Votre amour et votre soutien font que tout cela en vaut la peine.

"Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte."
— Blaise Pascal

Abstract

Text simplification is a task that involves rewriting a text to make it easier to read and understand for a wider audience, while still expressing the same core meaning. This has potential benefits for disadvantaged end-users (e.g. non-native speakers, children, the reading impaired), while also showing promise as a preprocessing step for downstream NLP tasks. Recent advancement in neural generative models have led to the development of systems that are capable of producing highly fluent outputs. However, these end-to-end systems often rely on training corpora to implicitly learn how to perform the necessary rewrite operations. In the case of simplification, these datasets are lacking in both quantity and quality, with most corpora either being very small, automatically constructed, or subject to strict licensing agreements. As a result, many systems tend to be overly conservative, often making no changes to the original text or being limited to the paraphrasing of short word sequences without substantial structural modifications. Furthermore, most existing work on text simplification is limited to sentence-level inputs, with attempts to iteratively apply these approaches to document-level simplification failing to coherently preserve the discourse structure of the document. This is problematic, as most real-world applications of text simplification concern document-level texts.

In this thesis, we investigate strategies for mitigating the conservativity of simplification systems while promoting a more diverse range of transformation types. This involves the creation of new datasets containing instances of under-represented operations and the implementation of controllable systems capable of being tailored towards specific transformations and simplicity levels. We later extend these strategies to document-level simplification, proposing systems that are able to consider surrounding document context and use similar controllability techniques to plan which sentence-level operations to perform ahead of time, allowing for both high performance and scalability. Finally, we analyze current evaluation processes and propose new strategies that can be used to better evaluate both controllable and document-level simplification systems.

Keywords: natural language generation, text simplification, text-to-text generation, machine learning, evaluation.

Résumé

La simplification de texte est une tâche qui consiste à réécrire un texte pour le rendre plus facile à lire et à comprendre pour un public plus large, tout en exprimant toujours le même sens fondamental. Cela présente des avantages potentiels pour certains utilisateurs (par exemple, les locuteurs non natifs, les enfants, les personnes ayant des difficultés de lecture), tout en étant prometteur en tant qu'étape de prétraitement pour les tâches de Traitement Automatique des Langues (TAL) en aval. Les progrès récents dans les modèles génératifs neuronaux ont conduit au développement de systèmes capables de produire des sorties très fluides. Cependant, étant donné la nature de "boîte noire" (black box) de ces systèmes de bout en bout, l'utilisation de corpus d'entraînement pour apprendre implicitement comment effectuer les opérations de réécriture nécessaires. Dans le cas de la simplification, ces ensembles de données comportent des limitation en termes à la fois de quantité et de qualité, la plupart des corpus étant soit très petits, soit construits automatiquement, soit soumis à des licences d'utilisation strictes. En conséquence, de nombreux systèmes ont tendance à être trop conservateurs, n'apportant souvent aucune modification au texte original ou se limitant à la paraphrase de courtes séquences de mots sans modifications structurelles substantielles. En outre, la plupart des travaux existants sur la simplification du texte se limitent aux entrées au niveau de la phrase, les tentatives d'application itérative de ces approches à la simplification au niveau du document ne parviennent en effet souvent pas à préserver de manière cohérente la structure du discours du document. Ceci est problématique, car la plupart des applications réelles de simplification de texte concernent des documents entiers.

Dans cette thèse, nous étudions des stratégies pour atténuer la conservativité des systèmes de simplification tout en favorisant une gamme plus diversifiée de types de transformation. Cela implique la création de nouveaux ensembles de données contenant des instances d'opérations sous-représentées et la mise en œuvre de systèmes contrôlables capables d'être adaptés à des transformations spécifiques et à différents niveaux de simplicité. Nous étendons ensuite ces stratégies à la simplification au niveau du document, en proposant des systèmes capables de prendre en compte le contexte du document environnant. Nous développons également des techniques de contrôlabilité permettant de planifier les opérations à effectuer, à l'avance et au niveau de la phrase. Nous montrons que ces techniques permettent à la fois des performances élevées et une évolutivité des modèles de simplification.

Mots-clés: génération de langage naturel, simplification du texte, génération de texte à texte, apprentissage automatique, évaluation.

Contents

List of Figures

List of Tables

Simplification de texte contrôlable et au niveau du document

La simplification de texte consiste à reformuler un morceau de texte afin qu'il soit plus facilement compréhensible pour un public plus large, tout en continuant à transmettre les mêmes informations de base. L'application sociétale principale de ceci est d'améliorer l'accessibilité de l'information pour les groupes défavorisés qui pourraient autrement avoir du mal à lire ou à comprendre.

Contrairement au domaine voisin du résumé de texte, où le but premier est de réduire la longueur d'un texte tout en ne véhiculant que les informations les plus importantes, la simplification vise plutôt à rendre les mêmes informations plus faciles à comprendre, sans nécessairement réduire la longueur du texte. Dans le cas de la simplification au niveau du document, les documents simplifiés résultants contiennent souvent plus de phrases que dans l'original (bien que le nombre de mots soit normalement réduit dans une certaine mesure) (Xu et al., 2015).

Afin d'effectuer avec succès la simplification du texte, une gamme de différents types de transformation doivent généralement être effectuées, qui peuvent être vaguement classées en variétés lexicales ou syntaxiques (Shardlow, 2014; Alva-Manchego et al., 2020b; Gooding, 2022). Les opérations lexicales comprennent le remplacement de mots et de phrases compliqués ou rares par des alternatives plus simples ou couramment utilisées, ou la suppression d'informations inutiles. Les opérations syntaxiques concernent la modification de la structure grammaticale d'un texte en remplaçant des constructions telles que les clauses relatives, les conjonctions et les appositions (généralement en divisant les phrases longues en plusieurs phrases plus courtes), en passant de la voix active à la voix passive, etc. (Siddharthan, 2006). Cela nécessite souvent aussi des modifications de préservation du discours comme la résolution d'anaphores afin de maintenir la cohérence du texte résultant. D'autres types d'édition peuvent également se produire, comme la simplification des concepts via des explications élaborées (Srikanth and Li, 2021; Gooding, 2022).

Pour illustrer plus clairement le processus de simplification de texte, considérons l'exemple suivant en anglais[1] :

A. Owls are birds from the order of Strigiformes, comprising over 200 species of mostly solitary and nocturnal birds of prey typified by an upright stance, binocular vision, binaural hearing, and sharp talons. Owls hunt mostly small mammals, insects, and other birds, although a few species specialize in hunting fish.

B. Owls are birds. There are over 200 species and are all animals of prey. Most of them are solitary and nocturnal. Owls' prey may be birds, large insects (such as crickets), small

[1]Cet exemple a été adapté des articles de Wikipédia en anglais et en anglais simple sur les hiboux et de la présentation de 2011 du Dr Mirella Lapata http://videolectures.net/esslli2011_lapata_simplification/

reptiles (such as lizards) or small mammals (such as mice, rats, and rabbits).

Ici, nous pouvons voir qu'une série d'opérations ont été effectuées afin de transformer le texte complexe A en la forme simplifiée B. Celles-ci incluent la substitution lexicale (par exemple *comprising* → *There are*), suppression du contenu inutile (par exemple *from the order of Strigiformes*), division des phrases en unités plus basiques (par exemple la première phrase de A est divisée en trois phrases dans B), ainsi que l'inclusion d'explications supplémentaires de certains concepts (par exemple, les exemples entre parenthèses dans la dernière phrase de B).

Les systèmes de simplification automatique des textes (ATS, en anglais) visent à réaliser cette tâche automatiquement grâce à l'apprentissage automatique ou à d'autres méthodes algorithmiques. Les systèmes ATS ont été proposés pour aider des publics tels que les locuteurs non natifs (Petersen and Ostendorf, 2007; Paetzold, 2016), les personnes ayant des capacités de lecture et d'écriture limitées (Watanabe et al., 2009), les enfants (De Belder and Moens, 2010; Gala et al., 2020), et les personnes souffrant de troubles cognitifs tels que l'aphasie (Carroll et al., 1998) ou la dyslexie (Rello et al., 2013). De plus, l'ATS peut être une étape de prétraitement utile pour améliorer les performances d'autres tâches de TAL (traitement automatique des langues) en aval comme la traduction automatique (Chandrasekar et al., 1996; Mishra et al., 2014; Li and Nenkova, 2015; Štajner and Popovic, 2016) et l'extraction de relations (Miwa et al., 2010; Niklaus et al., 2016).

Cette thèse vise à faire progresser le domaine de l'ATS en étudiant et en proposant des solutions potentielles aux limitations courantes des systèmes actuels et en explorant de nouvelles façons d'évaluer automatiquement leurs performances.

Conservatisme et manque de diversité. Dans le passé, les systèmes ATS ont souvent été constitués de composants multiples, chacun conçu avec l'intention d'effectuer un certain type d'opération de réécriture (par exemple, substitutions lexicales, division de phrases) (Siddharthan and Mandya, 2014; Narayan and Gardent, 2014; Ferrés et al., 2016). Ces dernières années, cependant, la majorité des systèmes ATS nouvellement proposés utilisent des modèles statistiques ou neuronaux guidés par les données qui sont capables d'apprendre les différentes opérations de réécriture implicitement à partir de grands corpus d'exemples de simplification (Nisioi et al., 2017; Martin et al., 2022). Cependant, dans la pratique, ces systèmes ont tendance à être plutôt conservateurs dans la quantité d'édition qu'ils effectuent et exécutent rarement des transformations structurelles compliquées telles que la division de phrases composées en unités plus basiques (Alva-Manchego et al., 2017). Ceci est souvent attribué en partie au chevauchement élevé entre les phrases d'entrée et de sortie et au manque de diversité des opérations dans les corpus de formation populaires, qui contiennent principalement des cas de substitution lexicale et de paraphrase de base (Xu et al., 2015). En conséquence, la plupart des systèmes présentent également une capacité limitée à générer des variantes alternatives d'un texte simplifié. Certains développements récents de systèmes de simplification contrôlables tentent d'atténuer ces problèmes en permettant la préconfiguration des attributs souhaités du texte de sortie (Martin et al., 2020) ou des opérations explicites à effectuer avant la génération (Scarton and Specia, 2018). Toutefois, le potentiel de ces techniques n'a pas encore été pleinement exploré.

Évolutivité au niveau des documents. La plupart des travaux antérieurs sur la simplification automatique n'envisagent que la simplification de phrases isolées, sans tenir compte du contexte des documents dont elles peuvent faire partie. Cela s'explique principalement par la complexité réduite de la tâche, qui facilite la curation des ressources de formation et l'applicabilité des systèmes issus de la littérature plus large sur la génération de textes (Alva-Manchego et al., 2020b). La transition de la simplification au niveau de la phrase à la simplification au niveau

du document est cependant la direction naturelle du domaine, car les documents sont le support le plus susceptible de nécessiter une simplification dans les applications du monde réel. Avant d'y parvenir, certaines limitations des systèmes actuellement proposés doivent être surmontées. L'application itérative de modèles de simplification de phrases ne tient pas compte de la possibilité d'opérations sur plusieurs phrases et ne permet pas de garantir que la cohérence du discours sera préservée dans le document simplifié qui en résulte (Siddharthan, 2003; Alva-Manchego et al., 2019b). En outre, les modèles neuronaux de bout en bout généralement utilisés pour la simplification des phrases ne s'adaptent pas facilement à de longues séquences d'entrée, ce qui signifie que leur application naïve à des documents complets en une seule fois ne permettra probablement pas d'atteindre le même niveau de performance que celui observé pour les phrases. Par conséquent, des systèmes capables de prendre en compte les aspects nécessaires du contexte et de la structure du document doivent être développés avant que la simplification automatique de texte ne puisse s'étendre au niveau de documents complets.

Limites de l'évaluation. L'une des principales limites de l'évaluation automatique des performances des systèmes de simplification réside dans le fait que la plupart des mesures courantes nécessitent des références de haute qualité, qui sont rares et coûteuses à produire. Cette situation est encore compliquée par le fait que de multiples références sont nécessaires pour garantir que les différents types de transformations viables sont correctement pris en compte. Dans le contexte des systèmes de simplification contrôlables, qui visent à simplifier sous des contraintes spécifiques ou à satisfaire les exigences de groupes d'utilisateurs spécifiques, cela pose un problème particulier. La création de simplifications de référence pour de longs documents sera également beaucoup plus difficile et prohibitive qu'elle ne l'est déjà pour des phrases individuelles. En outre, étant donné que la simplification des documents porte sur des textes beaucoup plus longs, les problèmes liés à l'incapacité des références à représenter toutes les transformations valides seront exacerbés par l'explosion du nombre de possibilités de simplification. En outre, les mesures existantes tentent généralement de quantifier la qualité de la simplification en tenant compte de plusieurs attributs (fluidité, adéquation sémantique et simplicité) à la fois, ce qui rend difficile de déterminer exactement où les systèmes réussissent. Certains travaux observent même une corrélation inverse entre l'adéquation et la simplicité (Schwarzer and Kauchak, 2018; Vu et al., 2018), ce qui implique qu'un certain degré de dégradation sémantique est nécessaire pour effectuer une simplification. Ainsi, le développement de meilleures stratégies d'évaluation et de mesures plus faciles à interpréter et moins dépendantes des références est crucial pour l'avancement du domaine.

Objectifs et contributions

Compte tenu des questions abordées ci-dessus, nous envisageons donc les questions de recherche suivantes :

- Comment pouvons-nous permettre aux systèmes de simplification d'effectuer une gamme plus diversifiée d'opérations, malgré une représentation limitée dans les données d'apprentissage existantes ?
- Pouvons-nous construire des systèmes de simplification de documents capables de s'adapter à des entrées longues tout en préservant la cohérence du discours dans le document résultant ?
- Compte tenu de la dépendance actuelle à l'égard des références, comment pouvons-nous rendre l'évaluation des systèmes de simplification plus adaptable à des textes plus volu-

mineux et à de nouveaux ensembles de données, tout en étant robuste à l'égard d'une diversité de simplifications ?

C'est pourquoi, dans cette thèse, nous visons d'abord à explorer les stratégies possibles pour atténuer la conservativité des systèmes ATS et leur permettre d'effectuer une plus large gamme de types de transformation. Cela implique la création de nouveaux ensembles de données contenant des instances d'opérations sous-représentées et la mise en œuvre de systèmes contrôlables capables d'être adaptés à des transformations et à des niveaux de simplicité spécifiques. Nous étendons ensuite ces stratégies à la simplification au niveau du document, en proposant des systèmes capables de prendre en compte le contexte du document environnant et d'utiliser ces techniques de contrôlabilité pour planifier à l'avance les opérations à effectuer au niveau de la phrase, ce qui permet d'obtenir des performances élevées et une grande évolutivité. Enfin, nous analysons les processus d'évaluation actuels et proposons de nouvelles stratégies qui peuvent être utilisées pour mieux évaluer les systèmes de simplification contrôlable et au niveau du document.

Concrètement, nos contributions sont les suivantes :

1. Une étude sur le cas particulier du découpage des phrases basé sur le discours et le développement d'un nouvel ensemble de données contenant de nombreuses instances de ces découpages.

2. Un système de simplification contrôlable formé à l'aide d'une combinaison d'ensembles de données garantissant une représentation diversifiée des opérations d'édition.

3. Un cadre pour la simplification de documents qui décompose la tâche en deux phases distinctes de planification puis de génération. Nous proposons également un système qui utilise un planificateur pour prédire une séquence d'opérations au niveau de la phrase basée sur le contexte du document, qui est utilisée pour guider un modèle de simplification, atteignant une performance de pointe.

4. Le développement de modèles de simplification capables de prendre en compte le contexte du document pendant la génération. Nous étudions également l'utilité de différents types de contexte, ce qui nous permet d'améliorer les résultats de notre système précédent.

5. Une nouvelle mesure d'évaluation apprise pour la simplification des phrases qui ne nécessite aucune référence. Elle présente une corrélation plus élevée avec les jugements humains sur la simplicité par rapport à toutes les mesures sans référence existantes et à la plupart des mesures basées sur des références.

6. Une étude préliminaire sur l'évaluation automatique de la fidélité sémantique dans la simplification des documents et la proposition de variations de métriques issues de la littérature sur les résumés.

7. Une étude sur les performances hors domaine des systèmes de simplification des documents et sur l'efficacité de l'évaluation automatisée dans un tel environnement.

Vue d'ensemble de la thèse

Le chapitre 1 (Background) présente la littérature existante sur la simplification de texte, y compris une discussion sur les systèmes antérieurs, les ressources de données et les techniques d'évaluation. Nous fournissons également une revue de la littérature relative aux deux axes principaux de la thèse – la simplification contrôlable et la simplification au niveau du document – en identifiant plus en détail les domaines que nous visons à explorer plus avant.

Diversité et contrôlabilité

Le chapitre 2 (Discourse-Based Sentence Splitting) identifie la variante de découpage de phrases sous-représentée du découpage de phrases basé sur le discours et décrit comment elle diffère du cas plus standard basé sur la syntaxe. Afin d'améliorer les performances de cette variante de découpage sous-représentée, nous proposons un nouvel ensemble de données contenant de nombreux exemples de découpage basé sur le discours. Nous extrayons des exemples d'Internet et de corpus de découpage/paraphrase existants qui couvrent un large éventail de relations discursives et de connecteurs. Nous menons des expériences avec des modèles de bout en bout et des systèmes de pipeline qui apprennent d'abord à reconnaître la relation discursive sous-jacente avant d'effectuer le découpage. L'évaluation expérimentale a révélé que les modèles de pipeline basés sur le discours ont de meilleures capacités de préservation des relations de discours que les modèles de bout en bout, et que les données synthétiques sont essentielles pour apprendre des modèles qui peuvent bien se généraliser (générer de multiples variantes valides de la même relation de discours).

Dataset	**# Instances**	**Discourse Relation**						
		Temporal		**Contingency**	**Comparison**	**Expansion**		
		Async	Sync	Cause	Contrast	Conj	Inst	Alt
D-MUSS	31,417	10,382	3,744	4,294	7,468	3,534	1,526	236
D-WikiSplit	371,117	192,798	21,076	36,086	59,729	44,739	10,346	6,343
Total Organic	402,534	203,183	24,820	40,380	67,197	48,273	11,872	6,579
D-CCNews-C	817,316	262,466	55,270	116,341	288,123	63,599	25,349	6,168
D-CCNews-S	999,437	113,298	150,105	102,956	69,864	345,189	137,178	80,847
Total	**2,219,287**	**578,947**	**230,195**	**259,677**	**425,184**	**457,061**	**174,399**	**93,594**

Table 1: Données d'entraînement à la "discourse splitting" (# Instances : Nombre de paires $\langle C, T \rangle$ pour D-CCNews-C, nombre de triplets $\langle C, T, S \rangle$ pour tous les autres ensembles de données, Conj:Conjonction, Inst:Instantiation, Alt:Alternative). Le premier niveau décrit les données du discours organique extraites de MUSS et WikiSplit, le deuxième niveau les données synthétiques dérivées de CC-News.

Le chapitre 3 (Controllable Sentence Simplification via Operation Classification) déplace notre attention du découpage des phrases vers le cas plus général de la simplification des phrases. Après avoir identifié les problèmes liés au fait que les modèles existants sont trop conservateurs et évitent certains types d'opérations, nous proposons une stratégie de simplification contrôlable qui permet l'exécution d'une gamme plus diversifiée de transformations possibles. Plus précisément, nous considérons quatre opérations globales (reformuler ou copier et diviser en fonction de la structure syntaxique ou discursive) et créons un nouvel ensemble de données qui peut être utilisé pour entraîner des systèmes de classification très précis pour identifier chacune d'entre elles. Nous proposons ensuite un modèle de simplification contrôlable qui adapte les simplifications à l'opération prédite et nous montrons qu'il est plus performant que les approches non contrôlables de bout en bout et que les approches contrôlables précédentes.

Simplification au niveau des documents

Le chapitre 4 (Document-Level Planning for Text Simplification) examine la simplification au niveau du document et traite de la manière dont elle diffère de la variante plus limitée au niveau de la phrase qui fait l'objet de la plupart des publications existantes. En s'appuyant sur les stratégies de simplification contrôlables discutées dans le chapitre précédent, nous proposons

un cadre qui se concentre d'abord sur la génération d'un plan de simplification au niveau du document qui est ensuite utilisé pour guider un modèle génératif en aval. Dans ce cadre, nous définissons un plan comme étant une séquence d'opérations au niveau de la phrase (par exemple, copier, supprimer, diviser, reformuler) qui devraient être effectuées sur chaque phrase dans le document complexe.

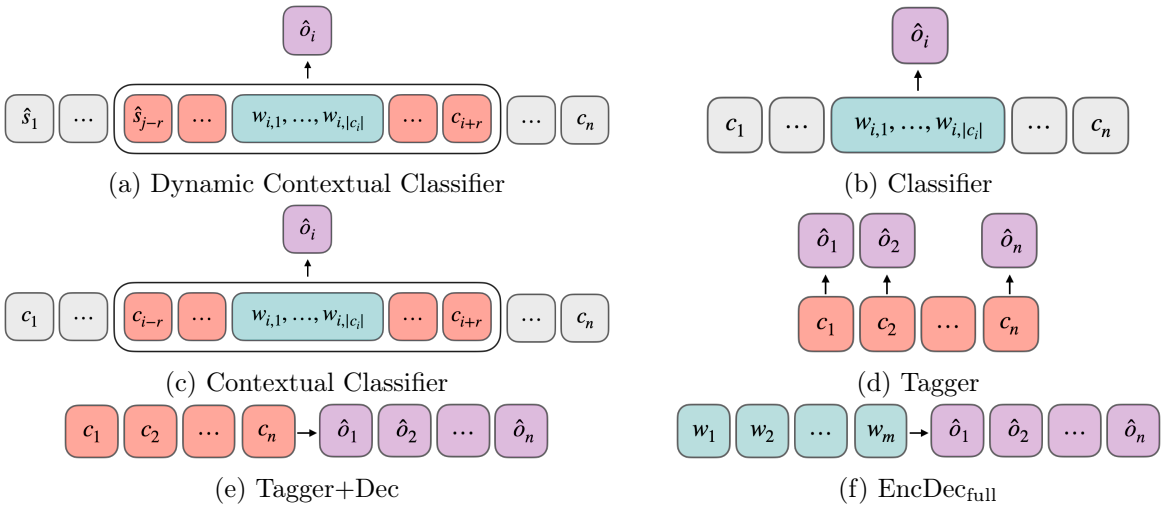


Figure 1: Visualisation des entrées/sorties des différents modèles, où $w_{i,t}$ est le $t$ème token dans c_i, n est le nombre de phrases dans C et m est le nombre de tokens dans C. Les représentations au niveau des phrases sont représentées en rouge, les représentations au niveau des jetons en sarcelle, les étiquettes des opérations en rose et les parties inutilisées de C en gris.

Plusieurs modèles potentiels sont comparés pour réaliser cette phase de planification, chacun considérant une représentation différente du contexte du document (Figure 1). Par exemple, nous expérimentons un classificateur de phrases sans contexte, un marqueur de séquences qui considère le document comme une séquence d'enchâssements de phrases, et un classificateur contextuel plus nuancé qui considère à la fois une représentation de la phrase au niveau du jeton et une représentation de haut niveau du contexte du document environnant. Nous montrons ensuite que cette approche conduit à des améliorations universelles des performances par rapport aux systèmes de bout en bout.

Le chapitre 5 (Context-Aware Document Simplification) étend le travail présenté dans le chapitre précédent en examinant comment le contexte du document peut potentiellement être utilisé dans la composante générative du système de simplification dans une tentative d'obtenir des gains de performance supplémentaires (ou d'ignorer complètement la planification). Nous proposons plusieurs systèmes capables de prendre en compte des contextes de différentes tailles (par exemple notre modèle ConBART, Figure 2) et résolutions (c'est-à-dire au niveau de la phrase, du paragraphe ou du document) et nous les comparons en termes de performance et d'efficacité.

Nos résultats montrent qu'une représentation de haut niveau du document peut être utile pour la réalisation de surfaces de bas niveau ainsi que pour la planification globale. De plus, les modèles de simplification ayant accès au contexte local du document, soit en travaillant au niveau du paragraphe, soit en traitant une représentation d'entrée supplémentaire, conduisent à une meilleure préservation du sens que ceux qui opèrent sur des phrases individuelles (résultats dans le tableau 2). En particulier, nos modèles obtiennent de bien meilleurs résultats que les sys-

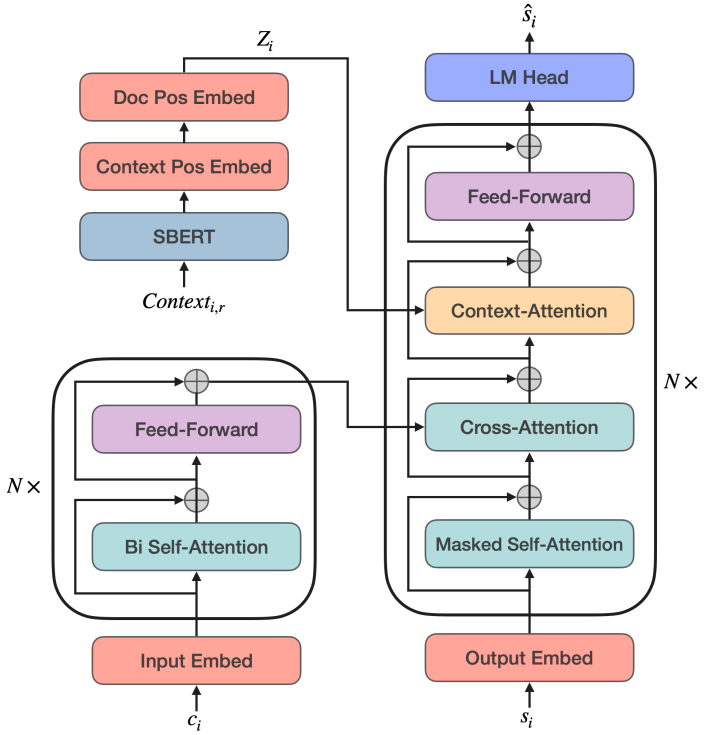


Figure 2: **Architecture du modèle ConBART.** La couche d'attention contextuelle ajoutée est représentée en jaune, ce qui permet une attention croisée sur le contenu de haut niveau du document, Z_i.

tèmes existants pour les cas nécessitant une édition importante (c'est-à-dire une grande différence entre le texte d'entrée et le texte de référence). Cela montre que la fourniture d'informations contextuelles au moment de la génération permet aux modèles d'être moins conservateurs.

System	**Fluency**			**Adequacy**			**Simplicity**			**Mean**
	Minor	Major	All	Minor	Major	All	Minor	Major	All	
Reference	90.9*	**96.0**	93.4	80.8	70.7*	75.8	83.8*	82.8*	83.3	84.2
PG_{Dyn}	91.9*	94.9	93.4	**83.8**	73.7	78.8	88.9	85.9	87.4	86.5
LED_{para}	**98.0**	92.9	**95.5**	81.8	80.8	81.3	**92.9**	85.9	**89.4**	**88.7**
$\hat{O} \rightarrow LED_{para}$	90.9*	**96.0**	93.4	80.8	**82.8**	**81.8**	83.8*	90.9	87.4	87.5
$\hat{O} \rightarrow$ ConBART	89.9**	**96.0**	92.9	81.8	79.8	80.8	86.9	**91.9**	**89.4**	87.7

Table 2: **Résultats de l'évaluation humaine de certains systèmes de simplification.** Le groupe *minor* comprend les exemples avec une transition de niveau de lecture de 2 niveaux (par exemple 0-2, 1-3, etc.), tandis que la classe *major* comprend ceux de 3-4 niveaux. Chacun de ces groupes représente la moitié de l'ensemble. Les notes significativement différentes de la note la plus élevée dans chaque colonne sont signalées par $*$ ($p < 0,05$) et $**$ ($p < 0,01$). La signification a été déterminée à l'aide de tests Z à deux proportions.

Metric	Human-Likert	Simplicity-DA✓
LENS	**0.531****	**0.429****
SARI	0.395**	0.109
BERTScore	0.389**	0.142
BLEU	0.333**	0.084
ΔSLE	**0.516****	**0.381****
$\Delta\text{SLE}_{\mathbb{Z}}$	0.479**	0.328**
FKGL	0.354**	0.260*
QUESTEVAL	0.134	0.090

Table 3: Corrélations absolues de Pearson avec l'appréciation humaine de la simplicité. Le niveau supérieur contient des mesures basées sur des références et le niveau inférieur des mesures sans référence. * indique une signification avec une valeur $p < 0{,}01$ et ** $< 0{,}001$.

Améliorer l'évaluation

Chapitre 6 (A Learned Reference-Less Metric for Sentence Simplification) propose une nouvelle mesure d'évaluation de la simplification qui ne nécessite aucune simplification de référence (SLE). Ceci est motivé par le fait que presque toutes les mesures populaires nécessitent de multiples références pour fonctionner de manière optimale, ce qui rend difficile l'évaluation de systèmes sur de nouveaux domaines sans passer par le processus laborieux d'écriture manuelle de références. En outre, nous montrons que les mesures existantes confondent également la simplicité avec d'autres attributs corrélés (par exemple, la préservation du sens), ce qui limite en fin de compte l'interprétabilité des scores.

Notre mesure, SLE, se concentre uniquement sur la simplicité sans tenir compte des autres caractéristiques du texte. Elle est également optimisée pour maintenir des prédictions moyennes précises au niveau du document, ce qui lui permet d'être plus précise en moyenne et de pouvoir être utilisée comme mesure agrégée au niveau du document. Les résultats indiquent que l'ELS est plus fortement corrélé avec les évaluations humaines que toutes les mesures sans référence existantes et que toutes les mesures basées sur des références sauf une (Tableau 3).

Le chapitre 7 (Semantic Faithfulness and Out-of-Domain Performance in Document Simplification) étudie les performances des systèmes de simplification de documents proposés dans les chapitres 4 et 5, en mettant l'accent sur la fidélité sémantique aux données d'entrée et sur les performances hors domaine. Nous tentons d'adapter les mesures de fidélité de la littérature sur les résumés à la simplification des documents et discutons de leur applicabilité à la tâche. Nous examinons également les performances des systèmes sur des domaines inédits et les comparons aux performances d'un grand modèle de langage (LLM).

En fin de compte, nous constatons que les systèmes guidés par un plan semblent mieux à même de s'adapter à des domaines inédits tout en équilibrant le compromis entre fidélité et simplicité et en limitant la conservativité. Le LLM a tendance à supprimer trop de contenu et à utiliser un langage moins simple que les systèmes dédiés (selon l'évaluation humaine et les mesures automatiques, y compris une variante de SLE). Toutefois, les mesures automatiques de fidélité ne correspondent pas aux préférences humaines et mettent en évidence la difficulté d'évaluer une fidélité suffisante lorsqu'il n'existe pas de référence à utiliser comme point de repère.

Introduction

Text simplification is the task of reformulating a piece of text so that it is more easily understandable to a wider audience, while continuing to convey the same core information. The primary societal application of this is to improve the accessibility of information for specific groups who might otherwise find it difficult to read or understand.

Different from the neighboring field of text summarization, where the primary aim is to reduce the length of a text while conveying only the most important information, simplification rather aims to make the same information easier to understand, without necessarily reducing the length of the text. In the case of document-level simplification, the resulting simplified documents often contain more sentences than were in the original (albeit the number of words is normally reduced to some extent) (Xu et al., 2015).

In order to successfully perform text simplification, a range of different transformation types generally need to be performed, which can loosely be categorized into lexical or syntactic varieties (Shardlow, 2014; Alva-Manchego et al., 2020b; Gooding, 2022). Lexical operations include substituting complicated or rare words and phrases for more simple or commonly used alternatives, or removing unnecessary information. Syntactic operations concern the modification of a text's grammatical structure by replacing constructs such as relative clauses, conjunctions and apposition (generally by splitting long sentences into multiple shorter sentences), switching from active to passive voice, etc. (Siddharthan, 2006). This often also requires discourse preserving modifications like anaphora resolution in order to maintain the coherence of the resulting text. Other types of editing can also occur, such as simplifying concepts via elaborated explanations (Srikanth and Li, 2021; Gooding, 2022).

To illustrate the process of text simplification more clearly, consider the following example:[2]

A. Owls are birds from the order of Strigiformes, comprising over 200 species of mostly solitary and nocturnal birds of prey typified by an upright stance, binocular vision, binaural hearing, and sharp talons. Owls hunt mostly small mammals, insects, and other birds, although a few species specialize in hunting fish.

B. Owls are birds. There are over 200 species and are all animals of prey. Most of them are solitary and nocturnal. Owls' prey may be birds, large insects (such as crickets), small reptiles (such as lizards) or small mammals (such as mice, rats, and rabbits).

Here, we can see that a range of operations have been performed in order to transform the complex text A into the simplified form B. These include lexical substitution (e.g. *comprising* → *There are*), deletion of unnecessary detail (e.g. *from the order of Strigiformes*), splitting of sentences into more basic units (e.g. the first sentence of A is split into three sentences in B), as

[2]This example was adapted from the English and Simple English Wikipedia articles on owls and Dr. Mirella Lapata's 2011 presentation: http://videolectures.net/esslli2011_lapata_simplification/

well as the inclusion of further explanation of certain concepts (e.g. the parenthesized examples in the last sentence of B).

Automatic text simplification (ATS) systems aim to perform this task automatically via machine learning or other algorithmic methods. ATS systems have been proposed to aid audiences such as non-native speakers (Petersen and Ostendorf, 2007; Paetzold, 2016), people with limited literacy skills (Watanabe et al., 2009), children (De Belder and Moens, 2010; Gala et al., 2020), and those suffering from cognitive impairments such as aphasia (Carroll et al., 1998) or dyslexia (Rello et al., 2013). Moreover, ATS can be a useful preprocessing step for improving performance on other downstream NLP tasks like machine translation (Chandrasekar et al., 1996; Mishra et al., 2014; Li and Nenkova, 2015; Štajner and Popovic, 2016) and relation extraction Miwa et al. (2010); Niklaus et al. (2016). However, this latter point might be less applicable in the case of neural systems.

This thesis aims to advance the field of ATS by investigating and proposing potential solutions to common limitations of current systems and exploring new ways to automatically evaluate their performance.

Conservativity and lack of diversity. In the past, ATS systems have often been comprised of multiple components each designed with the intent to perform a certain kind of rewriting operation (e.g. lexical substitutions, sentence splitting) (Siddharthan and Mandya, 2014; Narayan and Gardent, 2014; Ferrés et al., 2016). In recent years, however, the majority of newly proposed ATS systems use data-driven statistical or neural models which are able to learn the various rewriting operations implicitly from large corpora of simplification examples (Nisioi et al., 2017; Martin et al., 2022). However, in practice these systems tend to be rather conservative in the amount of editing they perform and rarely execute complicated structural transformations such as splitting compound sentences into more basic units (Alva-Manchego et al., 2017). This is often attributed in part to the high overlap between input and output sentences and lack of operation diversity in popular training corpora, which predominately contain instances of lexical substitution and basic paraphrasing (Xu et al., 2015). As a result, most systems also exhibit a limited ability to generate alternative variants of a simplified text. Some recent developments in controllable simplification systems attempt to mitigate these issues by enabling the pre-configuration of desired attributes of the output text (Martin et al., 2020) or explicit operations to perform prior to generation (Scarton and Specia, 2018). However, these methods generally require manual configuration to perform optimally, without being able to accurately determine desirable characteristics and operations automatically.

Document-level scalability. Most previous work on ATS only consider the simplification of isolated sentences, without regard for the context of the documents they might be a part of. This is primarily because of the reduced task complexity allowing for easier curation of training resources and applicability of systems from the wider text generation literature (Alva-Manchego et al., 2020b). Transitioning from sentence- to document-level simplification, however, is the natural direction of the field, as documents are the medium most likely to require simplification in real-world applications. Before this can be achieved, certain limitations of currently proposed systems needs to be overcome. Iteratively applying sentence simplification models fails to take into account the possibility of multi-sentence operations and has no way of ensuring discourse coherence will be preserved in the resulting simplified document (Siddharthan, 2003; Alva-Manchego et al., 2019b). In addition, the end-to-end neural models typically used for sentence simplification do not easily scale to long input sequences, meaning that naively applying them to full documents at once will likely fail to achieve the same level of performance seen on sentences. Therefore, systems that are capable of considering the necessary aspects of document

context and structure need to be developed before automatic text simplification can properly scale to the level of full documents. We address this in Chapters 4 and 5 by proposing systems that are able to plan and generate document-level simplifications by considering both the content of individual sentences and high-level representations of document content.

Evaluation limitations. A major limitation in automatically evaluating the performance of simplification systems is that most popular metrics require high-quality references, which are rare and expensive to produce. This is further complicated by the fact that multiple references are required to ensure that different types of viable transformations are properly considered. In the context of controllable simplification systems, which aim to simplify under specific constraints or to satisfy the requirements of specific user groups, this is a particular concern. Creating reference simplifications for long documents will also be significantly more difficult and prohibitive than it already is for individual sentences. Furthermore, because document simplification considers much longer texts, the issues arising from references being unable to represent all valid transformations will be exacerbated as the number of simplification possibilities explodes. Moreover, existing metrics generally try to quantify simplification quality by considering multiple attributes (fluency, semantic adequacy, and simplicity) at once, making it difficult to determine where exactly systems are succeeding. Some works even observe an inverse correlation between adequacy and simplicity (Schwarzer and Kauchak, 2018; Vu et al., 2018), implying that a certain amount of semantic degradation is a requirement for performing simplification. As such, the development of better evaluation strategies and metrics that are more interpretable and not so reliant on references is crucial for the advancement of the field.

0.1 Objectives and Contributions

Given the issues discussed above, we therefore consider the following research questions:

- How can we enable simplification systems to carry out a more diverse range of operations, despite limited representation in existing training data?
- Can we build document-level simplification systems that are capable of scaling to long inputs while also preserving discourse coherence in the resulting document?
- Given the existing reliance on references, how can we make the evaluation of simplification systems more scalable to larger texts and new datasets while being robust towards a diversity of simplifications?

As such, in this thesis we aim to first explore possible strategies for mitigating the conservativity of ATS systems and allowing them to perform a wider range of transformation types. This involves the creation of new datasets containing instances of under-represented operations and the implementation of controllable systems capable of being tailored towards specific transformations and simplicity levels. We then go on to extend these strategies to document-level simplification, proposing systems that are able to consider surrounding document context and use the aforementioned controllability techniques to plan which sentence-level operations to perform ahead of time, allowing for both high performance and scalability. Finally, we analyze current evaluation processes and propose new strategies that can be used to better evaluate controllable and document-level simplification systems.

Concretely, our contributions are as follows:

1. An investigation into the special case of discourse-based sentence splitting and development of a novel dataset containing many instances of these splits.

2. A controllable simplification system that is trained with a combination of datasets that ensure a diverse representation of edit operations.

3. A framework for document simplification that decomposes the task into separate planning and generation phases. We also propose a system that uses a planner to first predict a sequence of sentence-level operations based on document context, which is used to guide a simplification model, achieving state-of-the-art performance.

4. The development of simplification models that are able to consider document context during generation. We also investigate the utility of various types of context, allowing us to further improve upon the results of our previous system.

5. A novel learned evaluation metric for sentence simplification that does not require any references. It yields higher correlation with human judgements of simplicity compared to all existing reference-less metrics and most reference-based metrics.

6. A preliminary study on the automatic evaluation of semantic faithfulness in document simplification and proposal of variations of metrics from the summarization literature.

7. An investigation into the out-of-domain performance of document simplification systems and the efficacy of automated evaluation in such an environment.

0.2 Thesis Outline

Chapter 1 (Background) introduces the existing literature covering text simplification, including a discussion of past systems, data resources, and evaluation techniques. We also provide a review of literature pertaining to the two main focuses of the thesis – controllable simplification and document-level simplification – more thoroughly identifying the areas we aim to explore further.

Chapter 2 (Discourse-Based Sentence Splitting) identifies the under-represented sentence splitting variant of discourse-based sentence splitting and outlines how this differs from the more standard syntax-based case. In an attempt to improve the performance of this under-represented splitting variant, we propose a novel dataset containing many examples of discourse-based splitting and experiment with pipeline systems that first learns to recognize the underlying discourse relation before performing the split.

Chapter 3 (Controllable Sentence Simplification via Operation Classification) shifts our focus from sentence splitting to more general case of sentence simplification. After identifying the issues of existing models being overly conservative and avoiding certain types of operations, we propose a strategy for controllable simplification that allows for the execution of a more diverse range of possible transformations.

Chapter 4 (Document-Level Planning for Text Simplification) examines document-level simplification and addresses how this differs from the more limited sentence-level variant that is the focus of most existing literature. Building upon controllable simplification strategies discussed in the previous chapter, we propose a framework that first focuses on generating a document-level simplification plan which is subsequently used to guide a generative model downstream. Several potential models are compared for performing this planning phase, each considering a different representation of document context. We then go on to show that this approach leads to universal performance improvements over end-to-end systems.

Chapter 5 (Context-Aware Document Simplification) extends the work presented in Chapter 4 by investigating how document context can potentially be used within the generative component

of the simplification system in order to achieve further performance gains. We propose several systems that are capable of considering context of varying sizes and resolutions and compare them in terms of both their performance and efficiency.

Chapter 6 (A Learned Reference-Less Metric for Sentence Simplification) proposes a novel simplification evaluation metric that does not require any reference simplifications. This is motivated by the fact that almost all popular metrics require multiple references to work optimally, which makes it difficult to evaluate systems on new domains without going through the laborious process of manual reference writing. Moreover, we show that existing metrics also conflate simplicity with other correlated attributes (e.g. meaning preservation), ultimately limiting the interpretability of scores. Our metric focuses purely on simplicity without regard for other characteristics of the text.

Chapter 7 (Semantic Faithfulness and Out-of-Domain Performance in Document Simplification) investigates the performance of the document simplification systems proposed in Chapters 4 and 5, with particular focus on semantic faithfulness to inputs and out-of-domain performance. We attempt to adapt faithfulness metrics from the summarization literature to document simplification and discuss their applicability to the task. Ultimately, we show that plan-guided systems seem better able to adapt to unseen domains while limiting conservativity and balancing the trade-off between faithfulness and simplicity.

0.3 List of Publications

Some of the content presented in this thesis has previously appeared in the following publications:

Conference Publications:

- Liam Cripwell, Joël Legrand, and Claire Gardent. 2021. Discourse-based sentence splitting. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 261–273, Punta Cana, Dominican Republic. Association for Computational Linguistics
- Liam Cripwell, Joël Legrand, and Claire Gardent. 2022. Controllable sentence simplification via operation classification. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2091–2103, Seattle, United States. Association for Computational Linguistics
- Liam Cripwell, Joël Legrand, and Claire Gardent. 2023b. Document-level planning for text simplification. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 993–1006, Dubrovnik, Croatia. Association for Computational Linguistics
- Liam Cripwell, Joël Legrand, and Claire Gardent. 2023a. Context-aware document simplification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13190–13206, Toronto, Canada. Association for Computational Linguistics

Other Submitted Works:

- Liam Cripwell, Joël Legrand, and Claire Gardent. 2023d. On semantic faithfulness and out-of-domain performance in document simplification
- Liam Cripwell, Joël Legrand, and Claire Gardent. 2023e. Simplicity level estimate (sle): A learned reference-less metric for sentence simplification

Chapter 1

Background

Contents

Throughout this chapter we will discuss existing work from the text simplification literature as it pertains to the central topics of the thesis (i.e. controllable and document-level approaches). In the first section, we will describe popular datasets that have been used to train systems and consider the strengths and weaknesses of each. We will then move on to review evaluation strategies and provide a typological discussion of systems previously used to perform text simplification. Finally, we will provide additional context and motivation for both controllable and document-level approaches to ATS.

1.1 Data

Most modern ATS systems are data-driven, meaning that they consist of statistical and neural models that are trained to perform simplification by implicitly identifying transformation patterns within datasets of simplification examples. In this section, we outline the most popular corpora used to train such systems.

1.1.1 Wikipedia-derived Corpora

Many of the publicly available datasets that have been used for text simplification are derived from alignments made between English Wikipedia (EW) articles and their corresponding Simple English Wikipedia (SEW) version. Although there are many works that propose EW/SEW corpora constructed with varying alignment methods, we describe the most popular and thematically relevant of these here.

PWKP/WikiSmall Zhu et al. (2010) produced the first compilation of alignments between EW and SEW for use in training simplification systems (PWKP). They used TF*IDF to align sentences between the complex and simple article pairings. In total, they aligned 108K sentence pairs across a pool of 65K articles. In a later work, this dataset was further refined by removing duplicate sentences, resulting in a corpus of 89,042 training pairs (Zhang and Lapata, 2017). This later version is generally referred to as WikiSmall.

C&K Coster and Kauchak (2011) introduced another dataset of EW/SEW alignments using a different strategy (C&K-1). They first perform alignments at the paragraph-level using TF*IDF cosine similarities before applying a dynamic programming algorithm to find optimal sentence alignments while considering document context (Barzilay and Elhadad, 2003). In a follow-up work, Kauchak (2013) release a revised version of this dataset containing many more examples (C&K-2).[3]

AlignedWL/RevisionWL Woodsend and Lapata (2011a) leverage the same dynamic programming algorithm of Barzilay and Elhadad (2003) to compile yet another aligned dataset (AlignedWL), which was larger than the C&K-1 version released at the time. They also compile a more novel corpus (RevisionWL) from SEW article revision histories. They search for specific keywords in revision comments (e.g. *simple*, *clarification*, *grammar*) and then perform text difference checking to extract 15K complex/simple sentence pair examples.

WikiLarge Zhang and Lapata (2017) combined several of the existing EW/SEW datasets to create the then largest EW/SEW corpus. It includes the WikiSmall, C&K-2, AlignedWL, and RevisionWL corpora. In total, WikiLarge contains 296,402 training sentence pairs.

Turk Corpus Xu et al. (2016) propose a corpus containing several reference simplifications for each complex sentence, with the purpose being for use to evaluate system performances. They select 2,359 complex sentences from PWKP and have 8 crowdsourced Mechanical Turk workers write simplifications for each. However, they limit these simplifications to only include lexical changes, therefore making the resulting dataset suboptimal for evaluating general simplification models that are expected to perform syntactic/structural transformations as well.

ASSET The ASSET corpus (Alva-Manchego et al., 2020a) aims to resolve the limitations of TurkCorpus by incorporating examples that exhibit a wider range of simplification operations. As such, they take the same complex sentences as used in TurkCorpus and also have a group of crowdsourced workers manually write simplifications, while explicitly instructing them to perform

[3] https://cs.pomona.edu/~dkauchak/simplification/

multiple types of operations. The final dataset consists of 23,590 simplifications (10 per complex sentence).

Wiki-auto Jiang et al. (2020) exploit more modern techniques to produce another EW/SEW dataset purporting to contain more reliable sentence alignments. Specifically, they use a neural conditional random field (CRF) trained on manual sentence alignment annotations to compare semantic similarities between sentences throughout document pairs and predict alignments. They apply this method to 138K article pairs, resulting in a dataset of 488K sentence-level alignments.

The main appeal of EW/SEW corpora is that they provide a large pool of training examples which makes it very convenient to train neural models. However, because SEW articles are not necessarily written to be direct simplifications of their EW counterpart, coupled with the fact that sentence-level alignments are created using imperfect automatic methods, means that there is a lot of noise and poor-quality alignments within these datasets. Xu et al. (2015) explored this by manually annotating a random sample of 200 sentence pairs from PWKP and found that 50% of them are either incorrectly aligned (i.e. have different meanings) or contain a simple sentence that is not actually simpler than its complex variant. This problem is echoed in other parts of the literature, with many works finding that EW/SEW trained models regularly perform no editing whatsoever (Woodsend and Lapata, 2011b; Yasseri et al., 2012; Xu et al., 2015; Alva-Manchego et al., 2020b). When considering article pairings at the document level, the problem becomes more apparent, with almost all SEW articles containing fewer than 20% the amount of characters as their aligned complex article (Xu et al., 2015). This clearly shows that the SEW articles are not intended to contain the same content as the original and largely act more as a summary of the key information.

1.1.2 Newsela

In response to the many shortcomings of the Wikipedia-derived datasets, Xu et al. (2015) proposed a new simplification corpus (Newsela) that contains news articles manually written by professional editors to meet official reading standards for students at different grade levels. Specifically, they rewrite news articles 4 times such that the dataset contains five versions of each article at gradually increasing levels of simplicity. Because articles are written with the explicit goal of being simplified versions of the original, aligning sentences is naturally a much easier task. The fact that they were written by professional writers also provides a higher assurance of quality.

According to the analysis of Xu et al. (2015), Newsela exhibits a more diverse set of simplification operations than EW/SEW. There also appears to be a more extensive degree of simplification in Newsela, with the vocabulary size dropping by 50.8% between the original and highest simplicity level compared to the 18% drop observed in EW/SEW. Upon examination of discourse connectives present in the simplified texts, SEW also appears to contain a higher ratio of complex conjunction connectives (e.g. *even though*, *if ... then*, etc.) compared to Newsela. The same can be observed with regards to complex syntactic patterns.

The main limitation of the Newsela corpus is that it is not fully open source. Researchers must first obtain express permission from Newsela Inc. before they can conduct research using the data. As such, there is still reason to use Wikipedia in cases where it is necessary to publicly show system outputs vs inputs/references (doing so violates the terms of the Newsela license) and it has therefore remained the more popular dataset.

1.1.3 Other Datasets

WebSplit Narayan et al. (2017) introduce the simplification subtask, split-and-rephrase, which

aims to split a complex sentence into a meaning preserving sequence of shorter sentences. They derive instances from the WebNLG corpus (Gardent et al., 2017) which consists of Resource Description Framework (RDF) triples paired with plain English lexicalizations of their meanings. The resulting dataset, WebSplit, consists of examples containing (i) the RDF representation, (ii) the original lexicalization, and (iii) a set of sentences constituting valid splits of the original lexicalization.

WikiSplit Botha et al. (2018) offer a critique on the quality of WebSplit and present a new splitting corpus, WikiSplit, which consists of 1M examples extracted from Wikipedia revision histories. Here, Botha et al. (2018) apply string-match heuristics to identify cases where existing sentences have been transformed into a pair of sentences with approximately equivalent content to the original. One potential limitation of WikiSplit is that every example contains only one split, even when the resulting simple sentences could realistically be further split into additional sentences.

OneStopEnglish Vajjala and Lučić (2018) presented this corpus which contains 189 news articles manually rewritten to target three different levels of English language learner competency (elementary, intermediate, advanced). They include automatic sentence alignments between each level combination by using cosine similarity, which resulted in 6,994 individual alignment pairs (elementary-intermediate: 1674, elementary-advanced: 2166, intermediate-advanced: 3154). However, its relatively small size makes it difficult to exclusively rely on for training neural models.

MUSS The multilingual unsupervised sentence simplification dataset (MUSS) (Martin et al., 2022) contains several million pairs of text sequences mined from Common Crawl web data. They consider sequences of lengths up to 300 characters with limited punctuation and generate LASER embedding representations for each. Aligned paraphrase examples are then assigned according to L2 distance between embeddings. The resulting dataset contains 1.1M English pairs, 1.3M French pairs, and 1M Spanish pairs. The authors show that by using MUSS paraphrase examples for pretraining, with either a controllable generation training strategy or dedicated simplification fine-tuning, they are able to achieve state-of-the-art results on a range of simplification datasets. However, models trained on MUSS very rarely perform sentence splitting (3.45%), presumably as a result of the LASER embeddings being optimized for individual sentences and therefore failing to consistently identify splitting examples.

Non-English Data The vast majority of publicly available corpora for text simplification are specifically in and for English. This has meant that most cutting each work exclusively considers English, with progress and evaluation on other languages lagging behind. However, there are a range of resources available for other languages, such as French (Grabar and Cardon, 2018; Cardon and Grabar, 2020), Japanese (Maruyama and Yamamoto, 2018), Russian (Dmitrieva and Tiedemann, 2021; Sakhovskiy et al., 2021), Spanish (Xu et al., 2015; Saggion et al., 2015), Italian (Brunato et al., 2016), German (Säuberli et al., 2020), Danish (Klerke and Søgaard, 2012), amongst others. However, most of these are quite small in size or contain many instances of automatically translated text.

1.1.4 Discussion

Although many attempts to create and compile aligned simplification datasets have been made, available resources are still fairly limited compared to other NLP tasks. As most of the data is extracted from Wikipedia using automatic alignment methods, there is a lot of noise

Dataset	Sentence Pairs	Alignment Types
PWKP (Zhu et al., 2010)	108K	1-to-1, 1-to-N
WikiSmall (Zhang and Lapata, 2017)	89K	1-to-1, 1-to-N
C&K-1 (Coster and Kauchak, 2011)	137K	1-to-1, 1-to-N, N-to-1
AlignedWL (Woodsend and Lapata, 2011a)	142K	1-to-1, 1-to-N
RevisionWL (Woodsend and Lapata, 2011a)	15K	1-to-1, 1-to-N, N-to-1
C&K-2 (Kauchak, 2013)	167K	1-to-1, 1-to-N, N-to-1
Turk Corpus (Xu et al., 2016)	2,359	1-to-1, 1-to-N
WikiLarge (Zhang and Lapata, 2017)	292K	1-to-1, 1-to-N, N-to-1
WebSplit (Narayan et al., 2017)	1.3M	1-to-N
WikiSplit (Botha et al., 2018)	1.0M	1-to-N
ASSET (Alva-Manchego et al., 2020a)	2,359	1-to-1, 1-to-N
Wiki-auto (Jiang et al., 2020)	488K	1-to-1
Newsela-auto (Jiang et al., 2020)	666K	1-to-1, 1-to-N
OneStopEnglish (Vajjala and Lučić, 2018)	6,994	1-to-1
MUSS (Martin et al., 2022)	3.4M	1-to-1, 1-to-N

Table 1.1: Breakdown of the various aligned datasets used within the sentence simplification literature.

and imbalanced operation representation within these datasets, likely contributing to the conservativity problem observed with many neural simplification systems. Further, the possibility of misaligned references also means it is difficult to rely on in-domain evaluation results without also conducting human evaluation or using a smaller curated test set. Although manually constructed corpora like Newsela and OneStopEnglish show promise by ensuring higher quality data, they are either restricted by licensing constraints or are too small to be used as a basis for training large and robust models.

1.2 Evaluation

In general, there are three main output criteria that are considered important for the simplification task: (1) meaning preservation or semantic adequacy, which is the extent to which a simplification conveys the same meaning as the input text; (2) how fluent the generated text is; and (3) the overall simplicity of the output, which could include both lexical and structural simplicity. While the ideal would be to have automatic methods of reliably evaluating generated simplifications according to each of these qualities, the development of sound evaluation metrics is far from a solved problem, with human evaluation usually being carried out as a means to validate system performances. In this section we describe most evaluation metrics and strategies that have been used throughout mainstream simplification literature to date.

1.2.1 Automatic Evaluation

As a low-cost means of estimating the performance of simplification systems, researchers have made use of various automatic metrics that aim to approximate human judgements of simplification quality. Some of these are borrowed from other more common sequence-to-sequence tasks (e.g. machine translation), while others have been created specifically for simplification.

String Matching Metrics

BLEU (Papineni et al., 2002) is a popular metric originating from the machine translation literature. In simplification, it has been used to measure the similarity of generated outputs

to reference simplifications. Some studies claim that BLEU is highly correlated with meaning preservation and grammaticality (Wubben et al., 2012; Xu et al., 2016), but its suitability for evaluating simplification outside of these two dimensions is often questioned (Sulem et al., 2018a; Reiter, 2018; Alva-Manchego et al., 2020b).

The cornerstone of the metric is a modified n-gram precision, which is computed by counting the number of times a generated n-gram occurs in one of the references, clipping these counts by that n-gram's maximum frequency in any reference, summing all clipped counts, and finally dividing by the total number of generated n-grams. As this is a precision-oriented metric, it can favour very short outputs; so, they also impose a brevity penalty (BP).

The final BLEU metric can then be calculated as follows:

$$\text{BP} = \begin{cases} 1 & \text{if } c \geq r \\ e^{1-r/c} & \text{if } c \leq r \end{cases}$$

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{1.1}$$

where c is the output length, r is the reference length, p_n is a modified n-gram precision, and w_n is a weighting. Typically, $N = 4$ and $w_n = 1/N$.

TER (**T**ranslation **E**dit **R**ate) (Snover et al., 2006) is a translation metric that measures the amount of editing a human would have to perform in order to transform the input into the output text. This is done by counting the number of word insertions, deletions, substitutions, and shifts, each of which are considered as an individual edit. This count is subsequently normalized by the reference length.

$$\text{TER} = \frac{\#\text{ of edits}}{\text{average }\#\text{ of reference words}} \tag{1.2}$$

Although TER is not generally used to directly evaluate the quality of simplifications, it has been used as an indicator of how much editing is performed by models (Zhang and Lapata, 2017).

iBLEU (Sun and Zhou, 2012) is a modification of the BLEU score that aims to measure meaning preservation as well as the diversity of generated paraphrases. It does this by applying a penalty based on the similarity (BLEU) between the generated text and the input.

$$\text{iBLEU}(s, r_s, c) = \alpha\text{BLEU}(c, r_s) - (1-\alpha)\text{BLEU}(c, s) \tag{1.3}$$

where c is a candidate paraphrase, s is an input text, and r is a set of references. The authors recommend using $\alpha \in [0.7, 0.9]$ for the best balance between sentence quality and variety.

Readability Metrics

FKGL The Flesch-Kincaid grade level (FKGL) (Kincaid et al., 1975) is a document-level metric used to measure text readability. It has seen extensive use in the education field as a means to judge the readability of books and other text material for students. It has also been found to achieve the highest correlation with simplicity measures of human-written simplifications (Scialom et al., 2021b). The FKGL formula is as follows:

$$FKGL = 0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59 \tag{1.4}$$

The coefficients were derived from multiple regression procedures applied to Navy personnel reading tests. A lower number result implies that a text is more easily read than those with higher scores. Because it only looks at surface-level statistical features of a text, it does not rely on any references and can be computed from the output text alone. However, the fact that it only considers features like syllable counts and sentence length means it is somewhat limited in its ability to identify semantic simplicity in all cases (e.g. rare or complex words that happen to be short in length will still yield high readability scores).

FKBLEU Xu et al. (2016) combine FKGL and iBLEU in an attempt to produce a metric capable of measuring both simplicity/readability and meaning preservation.

$$FKdiff = sigmoid\left(FKGL(I) - FKGL(O)\right) \tag{1.5}$$

$$\text{FKBLEU} = \text{iBLEU}(I, R, O)^{1/2} \times FKdiff(I, O)^{1/2} \tag{1.6}$$

where I is an input, O is an output simplification, and R is a reference. The $FKdiff(I, O)$ function allows the metric to consider the relative change in readability resulting from the simplification. Unlike FKGL, a higher FKBLEU score suggests a better simplification with higher readability.

Dedicated Simplification Metrics

SARI (Xu et al., 2016) was introduced specifically for use in simplification. It compares system outputs with both their input and references, measuring the goodness of words added, deleted, and kept by the system. SARI is used widely in the simplification literature, being one of the most popular automatic metrics for measuring simplification quality.

Given an input I, output O, and references R, n-gram precision and recall scores for addition are defined so as to reward the inclusion of n-grams present in the references, but not in the input (i.e. $O \cap \bar{I} \cap R$):

$$\begin{aligned} p_{add} &= \frac{\sum_{g \in O} \min\left(\#_g(O \cap \bar{I}), \#_g(R)\right)}{\sum_{g \in O} \#_g(O \cap \bar{I})} \\ r_{add} &= \frac{\sum_{g \in O} \min\left(\#_g(O \cap \bar{I}), \#_g(R)\right)}{\sum_{g \in O} \#_g(R \cap \bar{I})} \end{aligned} \tag{1.7}$$

where $\#_g(\cdot)$ is a function giving a binary indication of occurrence of n-grams g in a given set.

$$\begin{aligned} \#_g(O \cap \bar{I}) &= \max\left(\#_g(O) - \#_g(I), 0\right) \\ \#_g(R \cap \bar{I}) &= \max\left(\#_g(R) - \#_g(I), 0\right) \end{aligned}$$

For retention, it rewards cases where n-grams are kept in both the output and the references:

$$\begin{aligned} p_{keep} &= \frac{\sum_{g \in I} \min\left(\#_g(I \cap O), \#_g(I \cap R')\right)}{\sum_{g \in O} \#_g(I \cap O)} \\ r_{keep} &= \frac{\sum_{g \in I} \min\left(\#_g(I \cap O), \#_g(I \cap R')\right)}{\sum_{g \in O} \#_g(I \cap R')} \end{aligned} \tag{1.8}$$

where R' is used to indicate scaling applied to n-gram counts if $r > 1$ references are used and

$$\#_g(I \cap O) = \min\left(\#_g(I), \#_g(O)\right)$$
$$\#_g(I \cap R') = \min\left(\#_g(I), \#_g(R)/r\right)$$

For deletion, only precision is considered, as the authors deem over-deletion to be significantly more important to simplification quality:

$$p_{keep} = \frac{\sum_{g \in I} \min\left(\#_g(I \cap \bar{O}), \#_g(I \cap \bar{R}')\right)}{\sum_{g \in O} \#_g(I \cap \bar{O})} \tag{1.9}$$

where

$$\#_g(I \cap \bar{O}) = \max\left(\#_g(I) - \#_g(O), 0\right)$$
$$\#_g(I \cap \bar{R}') = \max\left(\#_g(I) - \#_g(R)/r, 0\right)$$

Finally, the overall SARI uses the arithmetic average of n-gram precisions and recalls for these three operations:

$$SARI = d_1 F_{add} + d_2 F_{keep} + d_3 P_{delete} \tag{1.10}$$

where $d_1 = d_2 = d_3 = 1/3$, and

$$P_{operation} = \frac{1}{k} \sum_{n=1}^{k} p_{operation}(n)$$
$$R_{operation} = \frac{1}{k} \sum_{n=1}^{k} r_{operation}(n)$$
$$F_{operation} = \frac{2 \times P_{operation} \times R_{operation}}{P_{operation} + R_{operation}}$$
$$operation \in [add, keep, delete]$$

where k is the highest n-gram size (generally $k = 4$).

Despite its widespread usage, SARI has some known limitations. The small set of operations it considers means that it is much more oriented towards lexical simplifications, with Sulem et al. (2018b) showing that SARI actually obtains very low correlations with human ratings in cases where structural changes (e.g. sentence splitting) have been performed. As it is token-based it also cannot account for cases where synonyms of reference words are used in the generated output. Further, any potentially valid transformations that are not also expressed in the reference will be penalized; so, multiple, diverse references are a requirement for accurate appraisals.

SAMSA (Sulem et al., 2018b) is a metric that puts more emphasis on the structural aspects of a text. More specifically, SAMSA works under the assumption that simplified texts should contain separate sentences for each individual semantic scene (according to the UCCA semantic structure scheme (Abend and Rappoport, 2013)). Because of this, SAMSA does not require reference simplifications like SARI, BLEU, etc. SAMSA leverages a semantic parser (Hershcovich et al., 2017) to observe these changes made to predicate-argument structures, and thus for the sentence *"John got home and gave Mary a call."*, a higher score will be given to *"John got home. John gave Mary a call."* than for *"John got home and gave. Mary called."*.

While commonly used specifically to evaluate sentence splitting, it has also seen some use in simplification due to it being one of the only metrics to explicitly evaluate structural transformations. However, it is limited by its inability to consider lexical transformations and paraphrasing, meaning that is is poorly correlated with human judgements of general simplicity (Alva-Manchego et al., 2021). The assumption that all scenes should be within their own sentences to ensure simplicity is also not necessarily well founded in many cases. As such, it is ideally used as an estimator of structural simplicity alongside another metric which focuses on lexical simplicity (e.g. SARI) (Alva-Manchego et al., 2020b).

Learned Metrics

BERTScore (Zhang et al., 2019) is a learned metric that overcomes some of the limitations of n-gram matching metrics like SAMSA and BLEU via its use of token embeddings. The score is computed using the cosine similarities between token representations of the output and those of reference simplifications. BERTScore has been found to correlate highly with human ratings of simplicity (Alva-Manchego et al., 2021), leading to its recent adoption as a mainstream evaluation metric. There are three variants of the score (recall, precision, F1), with the precision-based variant seemingly being the most effective for simplification. However, BERTScore is still somewhat limited by the fact that it relies on reference simplifications. Furthermore, it is reportedly worse than SARI at differentiating conservative edits (Maddela et al., 2022) and its high correlation with simplicity ratings may be spurious (Scialom et al., 2021b). This makes sense intuitively, as it is only trained to estimate semantic similarity, without regard to the inherent simplicity of the text.

Quality Estimation (QE) Some works aim to predict the quality of generated simplifications without needing to rely on references. Štajner et al. (2014) attempt to predict the quality of simplifications in terms of their grammaticality and meaning preservation. Specifically, they train classifiers of two (good, bad) or three (good, medium, bad) classes using existing evaluation metrics (e.g. BLEU, METEOR, TER) as features. The results were somewhat promising, but with a lot of remaining room for improvement. Later, in the Shared Task on Quality Assessment for Text Simplification (QATS) (Štajner et al., 2016) the simplicity dimension was also considered; however, this proved much more difficult to predict than the other dimensions, with the best systems barely exceeding 50% accuracy. Performance was even worse for systems attempting to classify according to all 3 dimensions. Later, Martin et al. (2018) conducted an investigation into the effect of a wide range of features on quality estimation, examining their correlations with human ratings on the QATS dataset. By combining metrics and using them to train a range of models, they were able to slightly outperform QATS shared task submissions on meaning preservation and simplicity classifications, but note that maximizing performance on one dimension often leads to lower results on others (particularly meaning preservation vs. simplicity). However, despite somewhat promising results, QE-based systems have so far not been used to actually perform evaluation in simplification studies.

Pair-Wise Ranking A similar line of work has explored the ranking of multiple sentences according to their predicted simplicity levels. This was first considered as a possible solution in Vajjala and Meurers (2014), where it was shown that individual sentences in the EW/SEW corpora could not be easily separated into two groups, despite EW sentences generally being simpler than their aligned SEW counterpart (their binary classifier only achieved accuracy of 66%). Later, in Vajjala and Meurers (2016), they train a pair-wise ranking model (RankSVM (Herbrich, 2000)) using various lexical, syntactic and other linguistic features, which was able to correctly

identify the simple sentence in a given aligned pair from EW/SEW with approximately 80% accuracy. This approach does not estimate the absolute simplicity level of a given sentence, but it instead offers a relative ranking of two sentences. Some following works showed the possibility of further performance gains by incorporating different syntactic and psycholinguistic features (Ambati et al., 2016; Howcroft and Demberg, 2017) or leveraging neural model architectures (Lee and Vajjala, 2022).

1.2.2 Human Evaluation

Because of automatic evaluation metrics generally having some sort of limitation making it difficult to confidently rely on them, most simplification works also conduct human evaluation to ensure the consistency of their results. Generally, this involves human annotators being instructed to rate simplification outputs on three dimensions: fluency/grammaticality, adequacy/meaning preservation, and simplicity (Štajner et al., 2016) — with some works also including an *overall* criterion (Štajner et al., 2016; Sun et al., 2021). In most cases, ratings are given on a 0-100, or 1-5 Likert scale, with higher a number indicating better compliance with the quality dimension.

However, some studies stray from this standard approach by proposing alternative methodolgies. For instance, Xu et al. (2016) ask annotators to rate the *simplicity gain* of outputs by explicitly counting the number of individual lexical or syntactic paraphrases taking place during the simplification process. This was done to counter the potential effects arising from the simplicity variance amongst the input texts, and also to reduce over-punishment of errors that are already accounted for in the fluency or adequacy ratings. Surya et al. (2019) only use a binary 0-1 score for the simplicity dimension, asking raters to simply give an indication of whether or not a simplification output is a valid simplified version of the input. Sulem et al. (2018b) address the meaning preservation dimension by posing two separate questions: one asking if the simplification has added new information, and the other asking if it resulted in the removal of any important information. Further, they specifically ask annotators to give simplicity ratings by ignoring the complexity of the words – this was done to obtain a *structural complexity* rating.

Given the difficulty of weighing the importance of the various aspects of simplification quality, some works reduce human evaluation to a single dimension of *overall* simplification quality. However, this can result in the problem of not fully capturing the nuances of model performance, which can be important in diagnosing failures. Furthermore, some studies have suggested there is often a trade-off between the simplicity and adequacy dimensions (Schwarzer and Kauchak, 2018; Vu et al., 2018; Martin et al., 2018), the dynamics of which would not be detected via a single quality dimension. However, related issues still exist when using the standard evaluation procedure, with Scialom et al. (2021b) suggesting that human ratings of the various quality dimensions are often highly-correlated, meaning that many simplicity ratings could be spurious.

1.3 Existing Approaches

The ATS literature contains a vast array of different system types that have grown and faded in popularity throughout the years. The early approaches started by using rule-based systems to carry out common transformation types. Later, attention started to shift towards statistical and hybrid approaches which were largely influenced by machine translation work. Currently, almost all newly proposed systems are based on neural encoder-decoder architectures that, for the most part, operate in an end-to-end manner. In this section we will review influential works from each of these categories, which should provide enough background to enable the discussion of controllable and document-level simplification techniques to come later.

1.3.1 Rule-based Systems

The earliest work on text simplification mainly focused on developing handcrafted rules for the syntactic simplification of individual sentences. Chandrasekar et al. (1996) proposed identifying *articulation-points* where a given sentence could feasibly be split (e.g. phrase endings, subordinating and coordinating conjunctions, relative pronouns). Siddharthan (2002) presented a pipeline system that extracts clausal components from an input sentence and constructs individual stand-alone sentences for each, using a set of hand-written syntactic rules. Heilman and Smith (2010) incorporated rules for extracting simplifications from complex sentences via the removal of adjunct modifiers, discourse connectives, and conjunctions.

Ultimately, these techniques appear insufficient to perform robust and complete text simplification. They generally have a scope limited to specific syntactic patterns, cannot account for lexical substitutions and paraphrasing, and in many cases only consider individual sentences in isolation from their greater context (Siddharthan, 2003). Additionally, some of these systems have the main purpose of preprocessing text for further downstream NLP tasks, which often leads to outputs containing a large number of split sentences representing minimal semantic units, which do not read as smoothly as naturally written text (Heilman and Smith, 2010; Niklaus et al., 2019b,a,c). As such, they are not necessarily suited for user-facing applications of text simplification.

In response to some of these limitations, Siddharthan and Mandya (2014) proposed a unification between a rule-based style approach and more data-driven, lexically-focused techniques by introducing a system that uses a large set of hand-crafted syntactic rules in combination with lexical rules extracted from aligned EW/SEW corpora. Ferrés et al. (2016) took a similar approach using a set of syntactic rules in combination with a vector space model to perform lexical substitutions. However, in the current landscape of human-facing text simplification, rule-based methods have largely been abandoned in favour of data-driven approaches.

1.3.2 Statistical Machine Translation Approaches

Many early approaches to text simplification took inspiration from work in statistical machine translation (SMT) by treating the simplification task as a form of monolingual translation (i.e. English → Simple English). Generally, these are based on the noisy-channel model, using Bayes Theorem to model the following:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

where x is an input string and y is the translated output string. In this formulation, $p(x|y)$ is considered the translation model and $p(y)$ the language model. An optimal translation can then be found by maximizing this probability:

$$y^\star = \arg\max_{y \in Y} p(y|x)$$

Phrase-Based MT The first class of SMT approaches used in text simplification is phrase-based machine translation (PBMT), which considers word sequences as the primary unit of translation. Specia (2010) were the first to explore this, using the existing SMT toolkit, Moses (Koehn et al., 2007), trained on a parallel corpus of complex and simple Portuguese texts. They found that the model produced promising results, but was prone to being overly cautious in its transformations. Edits mostly consist of lexical simplifications and occasional structural rewritings, with more complicated transformations like sentence splitting not being supported.

Coster and Kauchak (2011) took a similar approach for English simplification, but added the possibility of deletion as a transformation, by allowing a complex phrase to be aligned to NULL. Wubben et al. (2012) too leverage the Moses toolkit, but include a novel post-processing step with the aim of mitigating transformation conservativity. In particular, they generate several simplifications for a given input, rank them according to edit distance with the input, then select the most dissimilar as the selected output. This achieved higher BLEU than many other approaches at the time.

Ultimately, PBMT approaches are capable of performing well when it comes to lexically-oriented simplification, where word substitutions and paraphrasings are sufficient. However, they are unable to accurately perform complex structural transformations like sentence splitting.

Syntax-Based MT An alternative class of models are those that use syntax-based machine translation (SBMT). Here, word sequences are replaced by syntactic components of the text as the primary unit of translation. For example, Zhu et al. (2010) uses syntactic information parsed from the input text to predict the suitability of four rewrite operations: splitting, deletion, reordering, and substitution. This method is based on the original SBMT approach of Yamada and Knight (2001, 2002) and uses the PWKP corpus to train the model. During the simplification process, the system traverses the input parse tree and determines likely edits to be made at each node. The system edits the tree to complete the predicted transformation before looking for candidate edits of the next operation type.

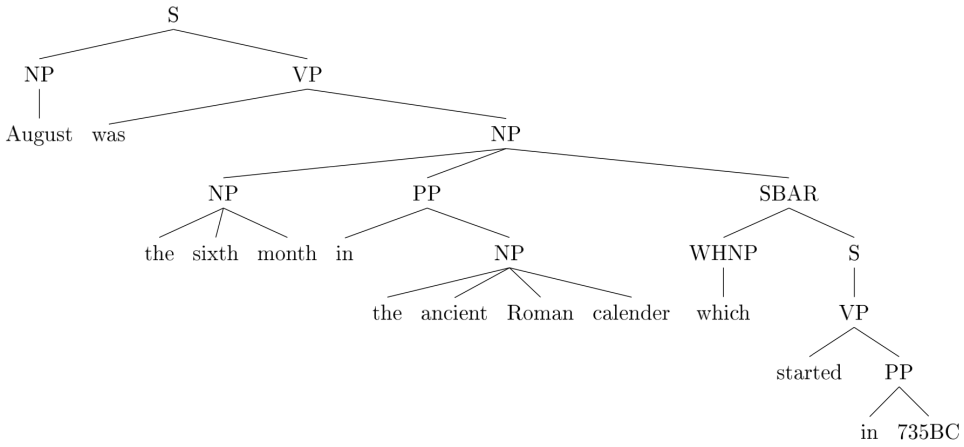
Xu et al. (2016) propose a system that uses an SBMT model in combination with rule-based features (word length, syllable counts, etc.) and tuning metrics specifically for lexical transformations (BLEU, FKBLEU, SARI). This system is able to perform quite well on the lexical metric it is optimized for, however is unable to carry out the same range of structural operations as Zhu et al. (2010).

1.3.3 Grammar-Induction Techniques

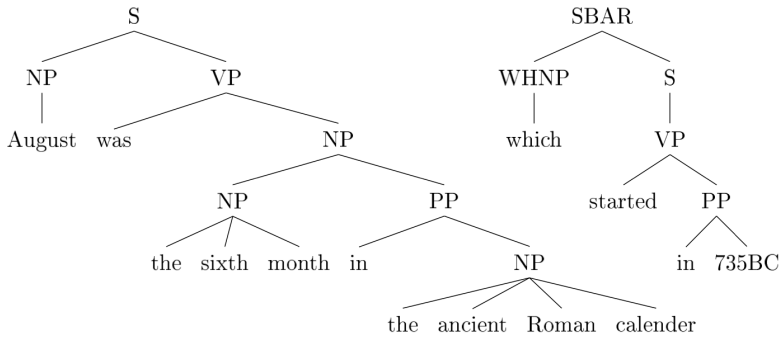
Some proposed strategies attempt to simplify sentences by approaching the task as a tree-to-tree rewriting problem. Generally, systems take the form of a pipeline, with a corpus being used to extract tree transformation rules before a trained model is used to select and carry out the execution of these rules. As such, this style of approach has some similarities with SBMT techniques (Alva-Manchego et al., 2020b).

Woodsend and Lapata (2011a) leverage a quasi-synchronous grammar (QG) induced from aligned Wikipedia data parse trees, in combination with an integer linear programming (ILP) model to select appropriate simplifications from the space of possible rewrites according to the grammar. Similar to Zhu et al. (2010), simplification is performed by the system traversing the input parse tree, applying appropriate transformation rules where a match is determined (multiple candidates are retained where more than one rule matches). The ILP model then determines the best simplification according to an objective function, which favours more common operations and the reduction of words and syllables in the output.

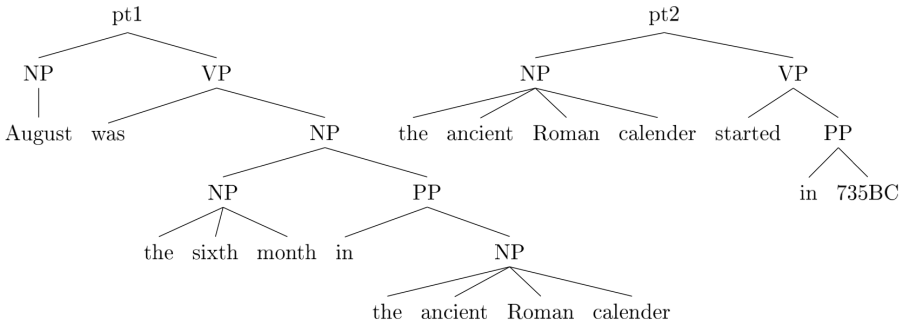
Paetzold and Specia (2013) extract simplification rules using the existing Tree Transducer Toolkit (T3) (Cohn and Lapata, 2009). Following this, they perform rule selection and ranking of sentence modifications in order to produce simplified texts. The ranking is determined by their perplexity in a language model trained on SEW. Although performance is promising in terms of lexical simplicity, the system struggles with structural simplicity, with modifications often resulting in loss of grammaticality and semantic adequacy. This is believed in part to be a result of T3 producing a very large number of, often spurious, syntactic rules.



(a) Original tree



(b) Segmentation



(c) Split resolution

Figure 1.1: **Example parse tree transformation for a sentence splitting operation.** Reprinted from Zhu et al. (2010).

Feblowitz and Kauchak (2013) propose an approach that aims to reduce this large ruleset produced by T3. During the decoding process, their system attempts to match the more specific rules with one of the more general ones, where possible. They then generate the 10K most probable simplifications and select the best according to a combination of various features. Although this approach achieves strong results in terms of both simplicity and grammatically, it yields worse meaning preservation compared to other models tested against.

Grammar-induction-based techniques have been viewed as offering more flexibility compared to end-to-end systems, due to the possible control over the rules that are learned and the way in which they are applied (Alva-Manchego et al., 2020b). For example, rules for additional operations could be added to a system later on. Despite this, they are still generally outperformed by many of the more modern text simplification methods.

1.3.4 Semantics-Assisted Systems

Narayan and Gardent (2014) take a rather different approach to sentence simplification by considering a deep semantic representation of the input rather than the raw text or its syntactic parse tree. They propose a system that includes a dedicated model for performing splitting and deletion in combination with a PBMT model for lexical edits (phrase substitution, reordering). Specifically, they use Boxer (Bos, 2015) to obtain a semantic representation of the input, from which candidate splitting pairs are selected from events connected via specific semantic roles. Deletions are also predicted according to the semantic relations of these events. For lexical simplification they employ the Moses SMT toolkit (Koehn et al., 2007) to train a model on the PWKP corpus, like in Specia (2010). Their system is able to achieve higher BLEU scores than existing systems, while also performing splits at a rate much closer to that exhibited in reference simplifications. According to human evaluators, they achieved the best results in terms of simplicity while being a close second to the best performing system on the grammaticality and adequacy dimensions (Wubben et al., 2012).

Later, Narayan and Gardent (2016) proposed another semantics-assisted system that learns to perform simplification in an unsupervised manner, without relying on aligned complex-simple sentence pairs. Essentially, they used the same semantic representation as in Narayan and Gardent (2014), but determine splitting points according to the maximum likelihood of sequences of thematic role sets resulting from the split. To do this, they train a probabilistic model on the deep semantic representations of sentences in SEW. They also include a context-aware lexical simplification component (Biran et al., 2011) and a dedicated ILP model for phrasal deletion (Filippova and Strube, 2008). The system was able to achieve results competitive with many of the supervised models at the time, according to human evaluations of simplicity and fluency. However, it was outperformed by most other systems in terms of adequacy.

Štajner and Glavaš (2017) also proposed a simplification system that is applied at the discourse level. They leverage an event extraction system and aim to delete irrelevant information from the texts by removing content that is not a part of factual event mentions. Hand-crafted rules are also applied to ensure each individual event mention is contained within its own sentence. Further, an existing unsupervised model based on word embeddings is used to perform lexical substitutions (Glavaš and Štajner, 2015). Like other systems of this kind, it achieves quite high human evaluation scores with respect to grammaticality and simplicity.

Compared to other system types that have been discussed so far, semantics-assisted simplification systems generally put much more emphasis on the sentence splitting operation. However, they are often prone to producing simplifications with poor meaning preservation, likely a result of over-deletion and disruption of content in the pursuit of sentences representing minimal

semantic units (Niklaus, 2022).

1.3.5 Split-and-Rephrase

Narayan et al. (2017) took the consideration of sentence splitting a step further by explicitly proposing a new simplification-adjacent task, *split-and-rephrase*. This task has the express goal of splitting sentences and rephrasing them where appropriate to ensure grammaticality, but without performing any deletions or lexical simplification, so as to preserve full semantic adequacy. They compile the WebSplit corpus and train a range of models using this data. Their best performing system, Split-Seq2Seq, is a pipeline that first uses a component trained to perform splitting given a complex sentence and its RDF meaning representation, then a separate sequence-to-sequence model trained to perform rephrasing given the RDF triple of a split sentence.

Aharoni and Goldberg (2018) attempt the split-and-rephrase task in a purely text-to-text fashion. They used a vanilla sequence-to-sequence model with attention (Bahdanau et al., 2016) which is able to perform competitively with Split-Seq2Seq. By introducing a copy mechanism within their models and reformulating the train-dev-test split of the WebSplit corpus, they are able to further improve the performance of their model dramatically. Botha et al. (2018) proposed a new split-and-rephrase dataset derived from Wikipedia edit history, WikiSplit, and show that using it to train the same end-to-end system as Aharoni and Goldberg (2018) results in significant improvements to performance over WebSplit.

Niklaus et al. (2019b,a,c) presented DisSim, which uses a set of 35 hand-crafted transformation rules to recursively decompose sentences into a hierarchical structure relating core sentences linked via rhetorical relations. They are able to produce accurate sentence splits while preserving the full information of the input. However, because the system is aimed at serving downstream NLP tasks, the resulting sentences are often very short and contain a lot of re-iteration. As such, they do not flow as naturally as standard human-written text, likely reducing readability in many cases.

1.3.6 Lexical Simplification

A literature focusing specifically on the lexical component of ATS has developed in parallel to the more mainstream approaches of holistic simplification (Glavaš and Štajner, 2015; Qiang et al., 2020; Saggion et al., 2023). Here, the focus is entirely on substituting words for simpler alternatives, rather than generating an entire simplified text from scratch. The strategies taken to perform lexical simplification (LS) more often take the form of a pipeline system (see Figure 1.2) with individual components for identifying complex words, sourcing and selecting substitutes, and ranking which is best given the context and simplicity of the words (Paetzold and Specia, 2017). As such, LS is often treated more like an information retrieval (IR) task with respect to evaluation, with metrics such as precision, recall, and MAP (mean average precision) being commonplace. In recent years, like for most NLP tasks, neural systems have also become the standard for LS (Qiang et al., 2020; North et al., 2023). Although LS systems may be sufficient for certain applications, it is obviously more restrictive compared to holistic ATS approaches, as it does not account for potential structural changes to the original text nor the substitution and deletion of larger phrases. Accordingly, this thesis will not put as much focus on these strategies going forward.

1.3.7 Neural Systems

Most recent text simplification systems take a neural sequence-to-sequence approach, often making use of attention-based encoder-decoder architectures, like in Bahdanau et al. (2016). As

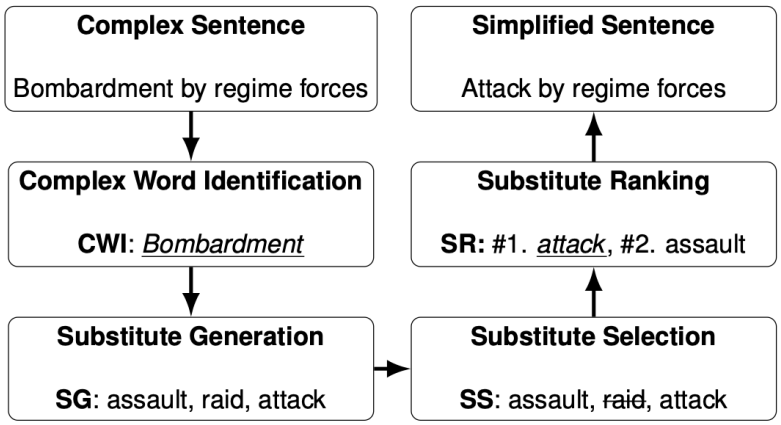


Figure 1.2: An example showing various steps of the typical lexical simplification pipeline. Reprinted from North et al. (2023).

these systems are trained in an end-to-end manner they are able to generate outputs with very high fluency, while performing both lexical and syntactic operations in combination, without the need for explicit feature extraction.

The first attempt at using sequence-to-sequence neural networks to perform text simplification was carried out by Nisioi et al. (2017). They propose their system, NTS, consisting of a network architecture containing two LSTM layers (Hochreiter and Schmidhuber, 1997) with global attention applied in the decoder, which is trained on aligned EW/SEW sentences. During generation, they sample multiple candidate hypotheses and use BLEU and SARI to select the best output. They were able to outperform existing state-of-the-art MT systems on BLEU, but were slightly outperformed in terms of SARI. According to human evaluations, their system outperformed all other systems across three criteria (grammaticality, meaning preservation, simplicity). An interesting observation is that all configurations of their system perform far fewer edits compared to existing methods, despite having a much higher rate of correct edits. This suggests a relative conservativity compared to other system types, despite simplifications generally being considered to be of a higher quality.

Alva-Manchego et al. (2017) explore the conservativity problem further, comparing NTS and another baseline neural model with the Moses PBMT system (Koehn et al., 2007). They find that although both Moses and NTS achieve better BLEU and TER than the baseline system, they are both very conservative, with generated simplifications having BLEU >90 with respect to the input. Further, sentence splits are very rarely performed (<2% frequency) relative to their frequency in the reference data (13.5%). For NTS, they are able to mitigate conservativity to some extent by excluding identical and 1-to-many alignments from the training set, but Moses remains extremely conservative. The human evaluation confirms these observations with all systems receiving simplicity ratings much lower than those of the reference simplifications.

Vu et al. (2018) avoid the standard LSTM-based model in favour of of a memory-augmented RNN architecture known as Neural Semantic Encoders (NSE) (Munkhdalai and Yu, 2017). Unlike LSTMs, which only have access to the previous hidden state, NSEs have unrestricted access to the entire input sequence stored in the memory. They presented two model variants, each tuned on a different evaluation metric: NseLstm-B (BLEU) and NseLstm-S (SARI). NseLstm-B was able to outperform every existing system in terms of BLEU, while NseLstm-S outperformed state-of-the-art neural and SMT systems with respect to SARI on Newsela and WikiSmall (not

so on WikiLarge). According to human evaluation, the NseLstm models generally outperformed all other systems on average. In particular, NseLstm-B achieved the highest fluency across all datasets, while NseLstm-S achieved the highest simplicity on WikiSmall and WikiLarge.

Zhang and Lapata (2017) proposed Dress, in which they also use a standard attention-based encoder-decoder architecture, but adopt a reinforcement learning fine-tuning process. A central motivation for this approach is to mitigate the impact of the inductive bias towards copying (i.e. conservativity) present in simplification datasets by maximizing a task-specific reward. Specifically, they make use of the REINFORCE algorithm (Williams, 1992) — a policy-gradient method which aims to maximize a reward with respect to the model parameters θ, approximated via Monte Carlo sampling. The gradient of the revised loss function for a given sequence is:

$$\nabla_\theta \mathcal{L}(\theta) = \sum_{t=1}^{|\hat{Y}|} \nabla_\theta \log \pi_\theta(\hat{y}_t | \hat{y}_{1:t-1}, X) r(\hat{y}_{1:|\hat{Y}|}) \tag{1.11}$$

where π_θ is the policy (i.e. the encoder-decoder model distribution) and $r(\cdot)$ is the reward function for a generated simplification, $\hat{Y}$. They follow steps taken in Ranzato et al. (2016) to improve learning stability, such as subtracting a token-level baseline from the reward in Equation 1.11 and adopting a curriculum learning strategy.

For their reward function, they include three weighted components, each focusing on a specific evaluation criteria: fluency, adequacy, or simplicity. For these, they respectively use: a language model's perplexity for the generated sequence; the cosine similarity between vector representations of the input and generated sequences; and a modified SARI score. To encourage better lexical simplification, they also train a separate encoder-decoder model without the RL fine-tuning and integrate it into the final policy via linear interpolation to form a revised variant of their system (Dress-Ls). However, it is worth noting that the authors do not make any similar attempt to explicitly improve the performance of sentence splitting.

Compared to existing systems, Dress and Dress-Ls achieved the two best FKGL scores on WikiSmall. On Newsela, Dress-Ls achieves the highest BLEU score. In general, the Dress models outperform a baseline neural encoder-decoder, with Dress-Ls being the better of the two. According to human evaluation, Dress-Ls outperforms all other systems on average, with all tested neural systems showing clear dominance over existing methods with respect to fluency.

Guo et al. (2018) proposed a system that learns to perform sentence simplification as part of a multi-task learning (MTL) framework. They use a standard encoder-decoder architecture with an added pointer-copy mechanism (See et al., 2017). They train it to perform sentence simplification as the main task, with the auxiliary tasks of paraphrase and entailment generation. They update all network weights with respect to simplification performance, but only update weights of lower- or higher-level layers for each of the auxiliary tasks (paraphrasing and entailment, respectively). This is based on the idea that lower layers contribute more to syntactic-level features, whereas the deeper layers are better for higher-level semantic tasks (Belinkov et al., 2017).

In an attempt to overcome the problems posed by training data scarcity and imbalanced operation representation, Zhao et al. (2018) proposed a system that incorporates external knowledge into an encoder-decoder. Specifically, they include human-curated paraphrasing rules from the Simple Paraphrase Database (SPPDB) (Pavlick and Callison-Burch, 2016).

In addition to the obvious concerns over conservativity, an important takeaway from examining the performance of neural systems on text simplification is their limited ability to perform structural transformations like sentence splitting. This is commonly believed to be in large part due to the high overlap between input and output texts and the relative scarcity of sentence splits in the training corpora when compared with other rewrite operations (Sulem et al., 2018c; Jiang

et al., 2020). Large-scale datasets with a diverse representation of rewrite operations simply do not currently exist for text simplification, and so, better training techniques and system architectures must be proposed in order to achieve improved performance on the task. In Section 1.4 we will discuss further neural systems that are specifically designed to overcome some of these limitations.

1.4 Controllable Simplification

So far, the most successful text simplification studies train statistical or neural models on pairs of complex and simplified sentences and assume that they will learn to perform the necessary operations implicitly from the inductive bias of the training data (Zhang and Lapata, 2017; Nisioi et al., 2017; Jiang et al., 2020). However, as discussed earlier, much of this data is obtained using distant supervision techniques and is often imbalanced in terms of the simplification operations contained within (Xu et al., 2016; Jiang et al., 2020). This has contributed to systems that generate very conservative outputs, often making little to no changes to the input or being limited to the paraphrasing of short word sequences. Additionally, these end-to-end systems provide limited capacity for controllability (unable to express alternative variants of the simplified text) (Alva-Manchego et al., 2017) which limits their ability to satisfy the requirements of different target audiences.

In order for ATS systems to successfully achieve their goal of improving accessibility to disadvantaged readers, it is important to understand and be able to address what exactly constitutes simplicity for different groups (Gooding et al., 2021). However, given the aforementioned lack of high-quality data resources and the difficulty and expense involved in evaluating for diverse audience groups, much of the ATS literature has converged to a generalized approach. As such, most text simplification systems of recent years are designed and evaluated in isolation from the actual downstream audiences they are ultimately intended for (Gooding, 2022). The development of systems that can be easily configured at inference-time to cater to the differing requirements of end-user groups is therefore of great value and importance. Some recent works have begun proposing controllable simplification systems, aiming to either constrain attributes of the output (length, amount of paraphrasing, lexical and syntactic complexity, etc.) or explicitly specify target reader groups and/or which transformation operations to perform. In this section, we will discuss some important works of this kind.

1.4.1 Mainstream Approaches

The first controllable ATS system was proposed by Scarton and Specia (2018), who modified input sequences with prepended tokens indicating target Newsela reading levels or high-level edit operations (e.g elaboration, splitting, copying), following a similar strategy to that taken by Johnson et al. (2017) to specify target languages in a multilingual MT system. By controlling for the target reading levels in this way, they achieved notable performance increases, but were unable to improve over a baseline when predicting which edit operations to specify (only using gold operation labels led to improvements). In a follow-up work, they were able to improve upon the operation prediction component of their system (from 0.51 to 0.71 accuracy) by using a more sophisticated neural classifier and data preparation steps (Scarton et al., 2020). This resulted in modest performance gains according automatic simplification metrics, but the variation in the classification accuracies for the different operation types was quite high (between 0.48 and 1.0). Nishihara et al. (2019) took a similar approach in controlling for target reading levels, but additionally modified the training loss according to a pre-defined vocabulary of words likely to occur at the given level.

Within their model, ACCESS, Martin et al. (2020) allow for the control of a wider range of surface-level attributes such as compression level, amount of paraphrasing, and lexical and syntactic complexity. During training, they use proxy statistics derivable from source and reference texts for each attribute, which are also integrated via control tokens. Statistics used include the character length ratio between the source and target sentence, Levenshtein similarity (Levenshtein et al., 1966) between the source and target sentence, word frequency ratios, and dependency tree depth ratio between the source and target. At inference time, they tuned each of these values to maximize SARI on the validation set, managing to significantly outperform SARI results of previous state-of-the-art systems on WikiLarge.

Most recently, Maddela et al. (2021) proposed a system that first uses a rule-based component (Niklaus et al., 2019b) to generate candidate output sequences that have undergone splitting and deletion, before ranking them and sending the top n to a neural paraphasing model. Tunable settings in both components provide control over how much of the input is changed and whether to favour deletion or splitting. Their system received higher simplicity ratings from human annotators compared to an end-to-end Transformer system and selected existing works. More interestingly, their system performs splitting far more often than the stand-alone Transformer (97% vs 53% of cases where reference simplifications contain a split) while also doing so more accurately (90% vs 49%), according to human annotators. This showed that the combination of neural and rule-based systems could potentially allow for exploiting the strengths while mitigating the weaknesses of each, allowing for a diversity of transformation operations despite imbalanced representation in training corpora.

1.4.2 Edit-Based Systems

One line of recent work has explored the possibility of using edit-based generative models to more efficiently perform simplification. Most of these strategies approach the task as a revision problem, rather than one of full generation, by identifying individual operations to perform (often at the token level) before using an additional mechanism to realize these, resulting in an edited version of the input.

The first of these used a bi-directional RNN to predict an edit operation for each input token (delete, replace, add, or copy), which are then realized downstream using naive substitution strategies (Alva-Manchego et al., 2017). It was able to achieve higher simplicity ratings from human evaluators than existing models, but achieved poorer results in terms of meaning preservation and grammaticality. Although quite simplistic in its design and limited in terms of the operations it can perform (e.g. unable to perform sentence splitting), this system showed potential as an alternative architecture to the standard neural approaches that allows for more interpretability and control over the specific transformations that are performed.

Dong et al. (2019) proposed a more sophisticated implementation of such a system that does not rely on external tools to perform lexical substitutions like in Alva-Manchego et al. (2017). Their model, EditNTS, uses a neural programmer-interpreter (NPI) to learn edit operations explicitly, unlike conventional seq2seq models. The programmer component predicts an operation z_t for a given input token x_t, given previously predicted operation labels $z_{1:t-1}$, a vector representation of other tokens in the input c_t, and tokens already generated by the interpreter component $y_{1:t-1}$. The interpreter then generates y_t by executing z_t. They are able to achieve results competitive with or better than existing neural systems, while managing to produce less conservative outputs. They also show that their system can be controlled to prioritize certain operations (e.g. amount of copying, insertion of novel words) by modifying the loss weights of certain labels.

Independently, Malmi et al. (2019) proposed a system with similar motivations, but instead of using an interpreter use a 1-layer decoder on top of a Transformer-based tagging model to autoregressively realize predicted edit operations, showing major inference speed improvements over seq2seq systems. By replacing the decoder layer with a non-autoregressive feed-forward layer, they achieve 100x faster inference speed with only minor performance degradation. Other works further explore the potential of edit-based models for ATS, in terms of the ability to perform fast non-autoregressive inference (Omelianchuk et al., 2021) and the possibility of iterative refinement (Kumar et al., 2020; Agrawal et al., 2021).

Although edit-based systems show potential for providing more interpretable alternatives to the commonly used MT-inspired seq2seq systems, they do lose some of the major benefits that make end-to-end neural systems so attractive. For instance, most edit-based systems require explicit modelling of individual low-level edit operations, rather than being able to implicitly learn them on-the-fly like end-to-end systems (which can also learn more complex structural transformations). The speed-up gains offered by non-autoregressive tagging and editing are also marred by the relative inability to consistently produce fluent and grammatical outputs that continue to preserve the input's meaning (Agrawal et al., 2021; Martin et al., 2021).

1.5 Document-Level Simplification

Although sentence-level simplification has been a popular NLP topic throughout recent years, there has been very limited existing work considering the simplification of entire documents. Alva-Manchego et al. (2019b) conducted an analysis of parallel Newsela documents, finding that many contain transformations operating across multiple sentences, meaning that iterative application of sentence-level systems would be insufficient to execute them. These include basic multi-sentence operations such as sentence reordering, sentence fusion and anaphora resolution, as well as more nuanced cases where context from other sentences is necessary to properly perform content deletion and addition (i.e. to handle redundancies and over-deletion). How to properly incorporate document context into systems in order to overcome these limitations remains relatively unexplored. It is also not clear whether the same evaluation metrics will remain as effective when used to assess entire documents. In this section we will outline existing work and available resources relating to the simplification of full documents.

1.5.1 Data and Evaluation

Lack of Quality Datasets

A major challenge for document-level simplification is the lack of high-quality datasets with which to train systems. Existing aligned corpora used for sentence simplification (such as EW/SEW and Newsela) do contain document pairs, but each have certain limitations. EW/SEW article pairs are aligned only by the article titles, with the simple articles themselves being written independently of the original (i.e. they are not necessarily direct simplifications) and are often of very different lengths (Sun et al., 2021). Figure 1.3 shows the difference in compression ratios between aligned documents in EW/SEW articles vs. Newsela. This is further illustrated by the noisiness of and difficulty in generating sentence-level alignments (Xu et al., 2015; Jiang et al., 2020).

Sun et al. (2021) recently proposed D-Wikipedia as an EW/SEW variant specifically for document-level use, but it also appears to be of rather low quality (Blinova et al., 2023). In particular, all text is lower-cased and pretokenized in a way that makes it difficult to accurately parse sentences. We also find regular formatting issues at points where citations exist in the

source articles. On the other hand, Newsela contains articles of much higher quality, with several versions of each that are manually written with the express purpose of being direct simplifications of each other. However, as previously discussed, Newsela is only available under a restrictive license, which makes it difficult to publish and share results with other researchers. OneStopEnglish (Vajjala and Lučić, 2018) is an open source dataset similar to Newsela, but is too small (less than 7000 aligned sentences) to be used to train neural models on its own.

For non-English languages, data resources are even more limited. There are some document-level corpora available for French (Gala et al., 2020), Spanish (Xu et al., 2015; Saggion et al., 2015), Italian (Brunato et al., 2015), Portuguese (Aluísio and Gasperin, 2010), and German (Säuberli et al., 2020; Rios et al., 2021; Aumiller and Gertz, 2022), but, as is the case for sentence-level corpora, these are generally much smaller (either in terms of corpus or document size) than those available for English. For the remainder of this thesis, we will primarily focus our attention towards English simplification.

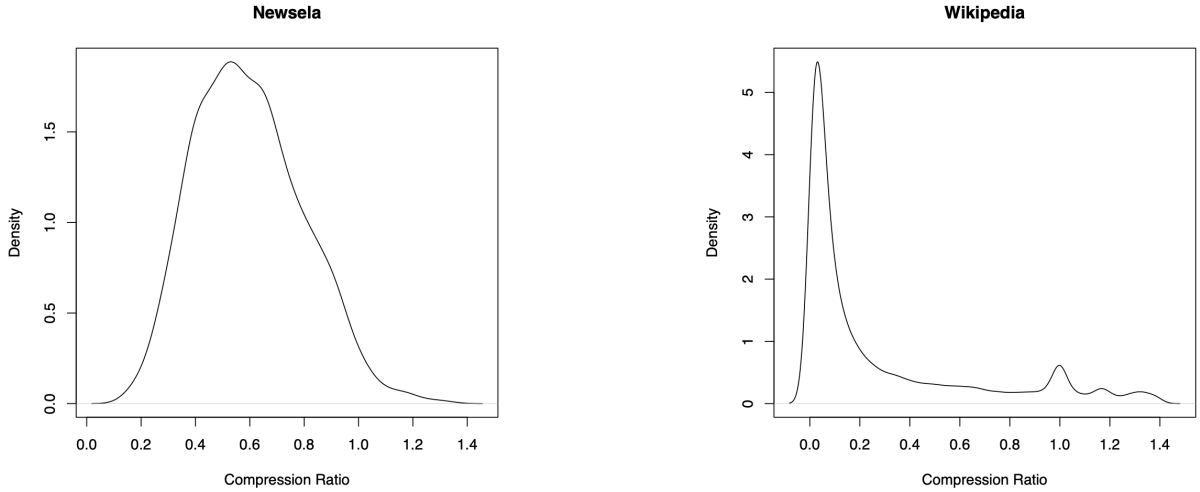


Figure 1.3: The document-level compression ratio of Newsela vs EW/SEW. This shows that Newsela has more consistent rate of compression, whereas Wikipedia is much more varied, often performing significantly higher amounts of compression. Reprinted from Xu et al. (2015).

Document-Level Evaluation Metrics

As document simplification is still a relatively unexplored task, there are yet to be many proposals of dedicated evaluation metrics. There has also been limited investigation into the applicability of existing metrics intended for sentence-level simplification or other text generation tasks.

FKGL is one metric regularly used in the sentence simplification literature that is actually intended to operate at the document level. As such, its applicability should naturally translate (and presumably improve) to document simplification. However, it cannot be relied upon for universal evaluation of simplification, as it focuses on surface-level readability and does not consider other aspects of simplification quality like fluency or semantic adequacy with respect to the input or references.

Sun et al. (2021) propose D-SARI, a modification of the original SARI metric that applies extra penalities according to the length difference between the generated document and reference simplification. Although the intuition behind these modification may be sound, no empirical

evaluation of how well D-SARI correlates with human judgements (the norm when proposing novel metrics in the sentence simplification literature) has been carried out.

Many model-based text generation metrics of recent popularity in the NLP literature, such as BERTScore (Zhang et al., 2019) and QuestEval (Scialom et al., 2021a), cannot easily be applied to document simplification, given they do not support sequence lengths exceeding 512 tokens. However, there are some alternatives that do support longer sequences, such as BARTScore (Yuan et al., 2021) (a state-of-the-art summarization metric), which could be more applicable. Although models of this kind could be effective for evaluating meaning preservation and fluency, the fact that they are not designed with simplification in mind means that they will likely not be able to evaluate simplicity itself. As such, the development of new metrics specifically targeting simplification at the document level should still be explored.

SMART (Amplayo et al., 2022) is a recently proposed text generation metric that considers sentences as the primary units of comparison and has been shown to be highly effective at evaluating document summarization. Furthermore, it can be used without relying on a neural model which makes it much faster to compute than other modern metrics for long documents (the inference-time of common neural networks often scale quadratically with input length).

1.5.2 Existing Techniques

Early Works

The first work to consider the discourse-level implications of simplification transformations was Siddharthan (2003). Specifically, they analyze possible issues that can arise when performing syntactic operations (e.g. sentence splitting) on individual sentences. However, their analysis does not fully extend to document simplification as they only consider sentence-level inputs and their potentially multi-sentence outputs.

Woodsend and Lapata (2011b) was the first work to actually attempt full document-level simplification. Given an EW article, their model selects salient phrases and sentences to include in the simple article, each of which is simplified through quasi-synchronous grammar (QG) rewrite rules. However, transformations are performed at the level of individual phrases or sentences, with the model having no notion of discourse-level document structure.

Angrosh et al. (2014) propose a two-part hybrid system using both hand written rules for syntactic simplification and automatically acquired rules for lexical constructs, combined with a new method for sentence compression. As with Woodsend and Lapata (2011b), simplification is performed on individual sentence inputs, meaning the system does not consider inter-sentential relationships throughout the larger document.

Alva-Manchego et al. (2019b) perform an analysis of parallel Newsela documents and find that there are many transformations that operate across multiple sentences. They go on to evaluate a typical sentence-level system when iteratively simplifying documents to act as a baseline for the document simplification task, ultimately concluding the future work needs to produce systems that can consider document-level features if the task is to be performed effectively.

Addressing Sub-Problems

While not directly aiming to perform document simplification, a number of works have addressed specific sub-problems which involve a more restricted set of operation types. As many of these focus on document-level transformations such as sentence fusions, deletion, reordering, etc. they use novel architectures and training setups in order to incorporate important features and information pertaining to structure and contents of the greater document, which could likely translate to application on general document simplification.

Zhong et al. (2020) attempt to predict whether or not to delete each sentence in a complex document, using both sentence- and document-level features within a feed-forward neural network (FNN) classifier. They consider the average GloVe word embeddings (Pennington et al., 2014) as a means to capture sentence-level semantics, features of the document such as number of sentences, as well as discourse-related features of the document's RST (Mann and Thompson, 1987) tree. Zhang et al. (2022) also performed sentence deletion prediction, but used a network architecture consisting of two document-level BiLSTM (Hochreiter and Schmidhuber, 1997) layers with intermediate self-attention and a final binary classification head. As input, it takes pretrained BERT (Devlin et al., 2019) representations of both the input sentence and max-pooled representations of the sentences surrounding it on either side (radius of 2) to capture local document context. Their best system is trained to jointly perform the deletion prediction as well as the classification of discourse content types (following the News discourse profiling categorization of Choubey et al. (2020)).

Srikanth and Li (2021) investigate the *elaboration* operation, which can be defined as the insertion of sentences containing information not already present in the complex document (e.g. to further explain certain concepts). They implement systems that generate elaborations for insertion at a specific point in the simple document, given the few previously simplified sentences. However, they approach elaboration as its own task without considering how it could interact within a more general simplification system. In particular, they do not consider the question of how to predict *when* elaboration is appropriate, which is something a general simplification system would need to be capable of.

Lin et al. (2021) propose a document-level paraphrasing model that uses an encoding of an inter-sentence coherence graph as well as sentence encodings as inputs to a decoder. The coherence-aware representations allow the system to re-organize the order of sentences within the document while also performing lexical paraphrasing. However, the system cannot perform important structural transformations such as sentence splitting or merging, and, because it is trained for paraphrase generation, outputs are not necessarily simpler than the inputs. The documents they paraphrase are also quite short, consisting of only 5 sentences each.

Holistic Approaches

A few recent works have approached document-level simplification with systems capable of implicitly performing multiple operations, much in the same way as standard neural sentence simplification systems. However, also like existing sentence-level systems, the models proposed here use a restricted window of document context, limited to either isolated paragraphs or a small number of adjacent sentences.

Laban et al. (2021) propose a system that performs simplification at the paragraph level. It is trained in an unsupervised manner using RL, with a reward made up of components measuring fluency, salience, and both lexical and structural simplicity. While the move to paragraph inputs expands the content window and allows for more complex structural transformations between sentences, it is still limited compared to document simplification. In particular, the paragraphs in Newsela (which is used to train their model) are quite short, being only around 2-3 sentences in most cases, meaning many of the same concerns of sentence-level approaches continue to apply if simplifying entire documents is the goal. They go on to use their system to simplify documents by iteratively processing one paragraph at a time, the results of which are then used in a human comprehension study. However, they do not evaluate the overall coherence between paragraphs and the documents they consider are only extracts of 3-4 paragraphs ($\sim$200 words), which are much shorter than standard Newsela articles. It remains unclear how much

better such an approach would scale to full-length documents compared to typical sentence-level models and whether or not document context wider than individual paragraphs is necessary to achieve consistent discourse coherence in generated outputs.

Devaraj et al. (2021) also approached simplification as a paragraph-level task, but focus on texts from the medical domain. The paragraphs they use are longer than what is considered in Laban et al. (2021), with inputs being of 14 sentences (or 521 tokens) and reference simplifications being of 10 sentences (or 274 tokens) in length on average. However, the fact that their data was sourced from *plain-language summaries* (PLS) of systematic reviews and the references are much shorter than the inputs suggests that this data might be more appropriate for summarization than simplification. Apart from some task-specific strategies to penalize the use of technical jargon, they leverage the standard encoder-decoder architectures that are commonly used for sentence simplification (texts are short enough to fit within maximum lengths supported by common transformer architectures). This same approach would not easily extend to longer documents like those in Newsela, which often exceed 1024 tokens in length (maximum length supported by BART (Lewis et al., 2020)) without revising the model architecture.

Sun et al. (2020) proposed a model for sentence-level simplification (SUC) which uses an encoding of surrounding sentences as contextual information to influence the simplification. They dedicate two additional encoders to build a representation of the two preceding and two following sentences of each input. These extra representations are then attended over in the main encoder-decoder generative model. In Sun et al. (2021), they apply SUC to document-level simplification by applying it to each sentence within an input document and concatenating results, ultimately finding that it was unable to outperform any of the baseline systems. Like with the paragraph-level models mentioned above, SUC has a relatively small window of context that may not be sufficient for capturing the long-range dependencies that are potentially necessary to properly produce coherent simplified documents.

Chapter 2

Discourse-Based Sentence Splitting

Contents

Sentence splitting is a task that involves the segmentation of one sentence into two or more shorter sentences. It is a key component of sentence simplification, has been shown to help human comprehension and is a useful preprocessing step for NLP tasks such as summarization and relation extraction. While several methods and datasets have been proposed for developing sentence splitting models, little attention has been paid to how sentence splitting interacts with discourse structure. In this chapter, we focus on cases where the input text contains a discourse connective, which we refer to as discourse-based sentence splitting. We create synthetic and organic datasets for discourse-based splitting and explore different ways of combining these datasets using different model architectures. We show that pipeline models that use discourse structure to mediate sentence splitting outperform end-to-end models in learning the various ways of expressing a discourse relation but generate text that is less grammatical; that large scale synthetic data provides a better basis for learning than smaller scale organic data; and that training on discourse-focused, rather than on general sentence splitting data provides a better basis to learn discourse splitting.

2.1 Introduction

Sentence splitting is a key component of text simplification involving the segmentation of one sentence into two or more shorter sentences. There is a large body of work where it has been

studied in the context of many text simplification systems (Siddharthan, 2006; Zhu et al., 2010; Woodsend and Lapata, 2011a; Siddharthan and Mandya, 2014; Narayan et al., 2017; Narayan and Gardent, 2016, 2014) as well as being the focus of so-called, split-and-rephrase models (Narayan et al., 2017; Aharoni and Goldberg, 2018; Botha et al., 2018; Niklaus et al., 2019b,a,c).

So far however, little attention has been paid to how discourse splitting interacts with discourse structure. As illustrated in Table 2.1, two main types of splitting can be distinguished depending on whether the split is licensed by a syntactic construct or by a discourse connective. Whereas syntax-based splitting is licensed by syntactic constructs such as relative clauses, verb phrase or sentence coordinations, gerund or appositive constructions, discourse-based splitting is licensed by the presence of a discourse relation between two discourse units.

Importantly, in the case of discourse-based splitting, the discourse relation which holds in the input must be preserved in the split output. This is illustrated in Table 2.1 where the temporal relation marked by ***and after this*** in the input (C1) is made explicit in the split output (S1) by the adverbial ***Afterwards***. In contrast, omitting this adverbial (S3) results in a semantic loss and makes the output more difficult to understand. As shown by the (S2) variant, a split can also use a discourse adverbial with an inverse meaning (***Before this***) which induces a corresponding inversion of the linear order of the text.

In this chapter, we focus on discourse-based sentence splitting and make the following contributions:

- We create synthetic and organic training data for discourse splitting and investigate various ways of leveraging this data for training discourse-based sentence splitting models.
- We compare a discourse-agnostic, end-to-end approach with a pipeline model that uses discourse structure to mediate the split.
- We show that training on discourse-focused rather than general sentence splitting data helps to improve performance.
- To assist future research on discourse-based sentence splitting, we have made our dataset and code publicly available.[4]

2.2 Related Work

Together with deletion, reordering and substitution, sentence splitting is one of the main operations used in text simplification. Early work on simplification used a rule based approach to splitting (Siddharthan, 2006; Siddharthan and Mandya, 2014). For instance, Siddharthan (2006) defines 26 handcrafted rules for simplifying apposition and/or relative clauses in dependency structures and 85 rules to handle subordination and coordination.

Further work focused on learning statistical simplification models from parallel datasets of complex-simplified sentences derived from English Wikipedia and Simple English Wikipedia. Zhu et al. (2010) introduces a syntax-based machine translation model where splitting probabilities are learned from syntactic structure. Woodsend and Lapata (2011a) induced a grammar from the parallel Wikipedia corpus annotated with syntactic trees and use an integer linear programming model for selecting the most appropriate simplification from the space of possible rewrites generated by the grammar. They report learning 438 rules for sentence splitting.

Probabilistic models have also been proposed. Narayan and Gardent (2014) determine splitting points using a dedicated probabilistic module trained on the Parallel Wikipedia corpus

[4]Our code and data is available at https://github.com/liamcripwell/disco_split.

C1.	The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell **and after this** Mindaugas crossed the Vistula river and captured the fortress of Jazdów.
S1. ✓	The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell. **Afterwards**, Mindaugas crossed the Vistula river and captured the fortress of Jazdów.
S2. ✓	Mindaugas crossed the Vistula river and captured the fortress of Jazdów. **Before this**, the Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell.
S3. ✗	Mindaugas crossed the Vistula river and captured the fortress of Jazdów. The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell.
T	<DR> TEMPORAL:ASYNCHRONOUS <ARG1> The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell <ARG2> Mindaugas crossed the Vistula river and captured the fortress of Jazdów <EOS>
C2.	He settled in London, devoting himself chiefly to practical teaching.
S4.	He settled in London. He devoted himself chiefly to practical teaching.
C3.	It was a time to go back to nature, and the plastic flamingo quickly became the prototype of bad taste and anti-nature.
S5.	It was a time to go back to nature. The plastic flamingo quickly became the prototype of bad taste and anti-nature.

Table 2.1: Discourse- (1) vs. Syntax-Based (2) Sentence Splitting

annotated with semantic structures while Narayan and Gardent (2016) extends this approach to an unsupervised setting where splitting points are determined based on the maximum likelihood of sequences of thematic role sets present in the simplified version of English Wikipedia.

More recent work has directly addressed the sentence splitting task. Narayan et al. (2017) introduce a dataset for training sentence splitting models called WebSplit and report results for various neural models trained on this data, comparing a vanilla sequence-to-sequence model with a multi-source and a semantically informed model. Aharoni and Goldberg (2018) present an alternative train/dev/test partition for WebSplit which better supports generalization and show that adding a copy mechanism helps improve results. One limitation of the WebSplit corpus is that it uses a small vocabulary. To remedy this shortcoming, Botha et al. (2018) create a new dataset called WikiSplit by mining Wikipedia's edit history. WikiSplit contains one million naturally occurring sentence splits. The authors show that incorporating WikiSplit as training data produces a model which outperforms prior results on the WebSplit test data by 32 BLEU points.

While these efforts are focused on syntax- or semantic-based sentence splitting, our work targets discourse-based sentence splitting. Closest to our work, Niklaus et al. (2019a,b,c) define a set of 35 hand-crafted transformation rules to recursively decompose sentences into a hierarchical structure relating core sentences linked via rhetorical relations. They do not generate a well-formed text and the proposed rule-based approach will fail too easily to generalize to other languages. Furthermore, because they focus on producing sentences representing minimal semantic units for downstream applications, their system outputs contain a large number of very short sentences which poses some readability issues for human audiences. In contrast, we present a dataset for training discourse splitting models and Transformer-based, encoder-decoder models for generating discourse splits. The included examples exhibit a single split per sentence and do not rely on a deep hierarchical representation of the discourse structure.

2.3 Tasks and Data

2.3.1 Tasks

We focus on cases of discourse-splitting such as illustrated in the top tier of Table 2.1, where the input text $C1$ includes a discourse connective ("after this") denoting a discourse relation between two discourse units and the split output includes a corresponding discourse adverbial ("Afterwards" in S2, "Before this" in S3).[5] We refer to the discourse tree representing the discourse structure of both C and S as T.

We consider two strategies: an end-to-end approach where the model directly splits the input text C into two shorter sentences S; and a pipeline approach where we first map C to a discourse tree T and then map this tree to the split output S.

2.3.2 Creating Data

We create $\langle C, S \rangle$ pairs using both synthetic and organic parallel data. We then extend these pairs to $\langle C, T, S \rangle$ triples, using rule-based and discourse parsing techniques, to create the associated linearized discourse tree T.

Creating C/S Pairs

Organic, Parallel Data. This is created by extracting discourse-split instances from two existing datasets, WikiSplit and MUSS. WikiSplit (Botha et al., 2018) is a sentence splitting dataset containing 1M single sentences alongside a two sentence variant which preserves the original meaning. This data is extracted from Wikipedia edit history, and therefore contains organic instances of C to S transformations. The multilingual unsupervised sentence simplification dataset (MUSS) (Martin et al., 2022) contains 2.7M pairs of text sequences mined from Common Crawl web data which were estimated to be paraphrases of each other using L2 distance on LASER embeddings. Filtering out only those pairs that represent a splitting operation yields a subset of 157K examples. Like WikiSplit, this dataset is organically human-authored.

To create a discourse splitting dataset, we then extract from these two datasets all instances where either the input contains a discourse connective or the output contains a discourse adverbial. We consider the discourse relations specified in the Penn Discourse Treebank (PDTB) (Prasad et al., 2008) and select a subset of these which we determined to be commonly represented via an adverbial connective between two sentences. We then compile a set of intra-sentential connective analogues for each. Table 2.2 shows the set of discourse connectives and adverbials used together with their corresponding discourse relations.[6] They cover 7 out of the 15 second order relations occurring in the PDTB.

Synthetic Data. The Common Crawl News corpus (CC-News) (Nagel, 2016) is a large collection of news articles that have been scraped from the internet. We use the *news-please* (Hamborg et al., 2017) python library to mine (i) a set of 1 million sentence pairs (D-CC-News-S) whose second sentence contains an adverbial and (ii) a set of 800K isolated sentences (D-CC-News-C) which contain a discourse connective. We then create the corresponding input text (C) and discourse tree (T) for each sentence in D-CC-News-S, and the corresponding discourse tree for each sentence in D-CC-News-C using rules and a discourse parser, as explained in the following section.

[5] We leave for future work cases where the input contains multiple or implicit discourse relations.

[6] We manually performed the mapping of discourse relations to a set of adverbials and equivalent connectives by studying the PDTB manual and examples from existing splitting datasets.

D-Reln	D-Con	D-Adv
TEMPORAL:Asynchronous	"and afterwards", "but afterwards", "after which", "then", "after that", "after this", "but, after that", "and after this", "after which", "eventually", "and eventually", "and in turn", "in turn", "which, in turn", "and then", "and so", "later", "and later", "but later", "next", "before", "followed by", "when", "thereafter", "and thereafter", "after which", "before that", "but before that", "although before that", "prior to this", "earlier", "and earlier", "formerly", "previously", "after", "and previously", "recently"	"afterward(s)", "after that", "eventually", "in turn", "later", "next", "thereafter", "before that", "earlier", "previously"
TEMPORAL:Synchrony	"in the meantime", "but in the meantime", "whilst", "meanwhile", "while in the meantime", "while", "simultaneously", "and simultaneously"	"in the meantime", "meanwhile", "simultaneously"
CONTINGENCY:Cause	"accordingly", "so", "as such", "and as such", "as a result", "and as a result", "however", "so that", "resulting in", "consequently", "and therefore", "and so", "with", "therefore", "which means", "which means that", "thus", "and thus", "thusly"	"accordingly", "as a result", "consequently", "therefore", "thus"
COMPARISON:Contrast	"by comparison", "in comparison", "while", "compared to", "whilst", "by contrast", "in contrast", "and in contrast", "while", "although", "conversely", "and conversely", "nevertheless", "but", "none the less", "yet", "however", "on the other hand", "and on the other hand", "but on the other hand", "but", "whereas"	"by/in comparison", "by/in contrast", "conversely", "nevertheless", "on the other hand"
TEXPANSION:Conjunction	"additionally", "and additionally", "and also", "and is also", "besides", "besides this", "aside from", "further", "furthermore", "and furthermore", "and further", "in addition to", "likewise", "and likewise", "moreover", "indeed", "similarly", "and similarly", "while"	"additionally", "also", "besides", "furthermore", "in addition", "likewise", "moreover", "similarly"
EXPANSION:Instantiation	"for example", "for instance", "such as", "in particular"	"for example", "for instance", "in particular"
EXPANSION:Alternative	"instead", "but instead", "though", "but rather", "rather"	"instead", "rather"

Table 2.2: Discourse Relations, Connectives and Adverbials

Creating $\langle C, T, S \rangle$ Triplets

We use discourse trees (i) to derive $\langle C, T, S \rangle$ triplets from the parallel data and (ii) to create matching C texts for the S texts in D-CC-News-S.

Creating Discourse Trees. For a given S, we employ the following rule-based method to derive a linearized tree, T, of the form shown in Table 2.1. The adverbial is removed from the sentence pair and mapped to the corresponding PDTB discourse relation while the two sentences are used as the tree's arguments and are rearranged into the linearized tree for the relation instantiated by the adverbial. The ordering of the arguments is determined according to a defined schema for each relation, as stipulated in the PDTB manual. In Table 2.1, T shows an example of this for the *Temporal.Asynchronous* relation, where the arguments are ordered chronologically.

To create a discourse tree for a complex sentence C occurring in D-CC-News-C, we use the end-to-end PDTB discourse parser of Lin et al. (2014). Although not the most recent discourse parser, it was selected because it is publicly available as a simple to use end-to-end system and specifically uses the PDTB schema. However, we noticed that the parser often fails to extract the arguments of the relation, so we also fall back to using a naive extraction strategy in such cases. This naive approach works by selecting the content on either side of the connective as the relation arguments.

In both cases (deriving a discourse tree from a complex sentence C, or a pair of sentences S), the created discourse tree is similar to a PDTB discourse tree in that it uses the PDTB inventory of discourse relations and order their arguments according to the PDTB annotation guidelines.

Deriving Complex Sentences from pairs of Simple Sentences. We derive a single sentence variant C from a sentence pair S in D-CC-News-S using a simple rule-based method which fuses the pair while maintaining the appropriate discourse relation and instantiating different possible argument orderings and connective alternatives. This process works by first randomly selecting a connective from the set of possibilities, given the adverbial in S, and then combining it along with the two arguments to form a single sentence. These combinations are of the form "arg1 connective arg2" or "connective arg1, arg2", depending on the selected connective.

This method only partially captures possible variations between C and S due to C being constructed from S using simple rules that do not take into account lexical variability (paraphrasings, etc.) that can exist for organic examples. However, as shall be shown in Section 2.6, because it permits creating multiple discourse variants of the same discourse split S using different connectives and orderings, this synthetic data helps to train discourse splitting models that are better able to generalize, such that they can generate different constructions for the same relation.

We do not attempt to automatically derive S from C for D-CC-News-C as this is a more complex task requiring many more alterations to reliably produce coherent samples. For instance, when there is a connective at the beginning of the sentence, it is difficult to identify which parts of the remaining sentence constitute the individual relation arguments. Additionally, rewriting and coreference resolution regularly need to be performed.

2.3.3 Training and Test Data

Table 2.3 summarizes the data used for training and development. For evaluation, we extracted a test set of 352 $\langle C, T, S \rangle$ triples from the organic datasets (184 triples from WikiSplit and 168 from MUSS), making sure to maintain an approximately even distribution over the supported connectives. To ensure a high level of quality, we then manually corrected the contents of T, C, and S, where necessary i.e., when C and S connectives did not match or when the wrong parts of the text had been flagged as relation arguments in T.

Dataset	# Instances	Discourse Relation						
		Temporal		Contingency	Comparison	Expansion		
		Async	Sync	Cause	Contrast	Conj	Inst	Alt
D-MUSS	31,417	10,382	3,744	4,294	7,468	3,534	1,526	236
D-WikiSplit	371,117	192,798	21,076	36,086	59,729	44,739	10,346	6,343
Total Organic	402,534	203,183	24,820	40,380	67,197	48,273	11,872	6,579
D-CCNews-C	817,316	262,466	55,270	116,341	288,123	63,599	25,349	6,168
D-CCNews-S	999,437	113,298	150,105	102,956	69,864	345,189	137,178	80,847
Total	**2,219,287**	**578,947**	**230,195**	**259,677**	**425,184**	**457,061**	**174,399**	**93,594**

Table 2.3: Discourse Split Training Data (# Instances: Number of $\langle C, T \rangle$ pairs for D-CCNews-C, number of $\langle C, T, S \rangle$ triples for all other datasets, Conj:Conjunction, Inst:Instantiation, Alt:Alternative). The top tier describes the organic discourse data extracted from MUSS and WikiSplit, the second tier the synthetic data derived from CC-News

2.4 Models Types

Given a complex sentence C with discourse tree T and split output S, we consider and compare two approaches: an end-to-end system $C2S$ where the split output S is directly generated from C; and a pipeline system PL which uses C's discourse tree to mediate the split i.e., first mapping C to its discourse tree T and second, mapping this tree to the split output S. We try both of these approaches to investigate how difficult it is for an end-to-end model to incorporate the discourse structure on its own and to what extent, if any, explicit mediation of this information aids the performance of discourse-based splitting.

For each of these two strategies, we explore different ways of combining the training data: using only the synthetic data (Synth), only the organic data (Organic) or both (Synth+Organic). We also investigate a pretraining and fine-tuning approach where we pretrain on the synthetic data and fine-tune on the organic data; and a multi-task learning approach where we multi-task on the intermediate mapping tasks (mapping C to T and mapping T to S) and the end-to-end task (mapping C to S).

2.5 Experimental Setup

All of our generative models use the BART architecture (Lewis et al., 2020) and were trained on a computing grid using 4 Nvidia RTX 2080 Ti GPUs. Each experiment starts by fine-tuning the `facebook/bart-base` model hosted by HuggingFace[7], which has 6 layers in each of the encoder and decoder, a hidden size of 768, and was pretrained to perform reconstruction of corrupted documents on a combination of books and Wikipedia data.

During training, we used a learning rate of $3e^{-5}$, a batch size of 16, and performed dropout with a rate of 0.1 and early stopping as regularization measures. For each experiment we set aside 5% of the training set for validation. During generation, we perform beam search with a beam size of 4.

We compare the following models:

Split Baseline (BL_{split}) pretrained BART fine-tuned on a 1M example dataset of both syntax- and discourse-based splits (WikiSplit). This baseline allows us to compare training with very large heterogeneous training data (BL_{split}) vs. learning from smaller, discourse-split data (BL_{dsplit}).

[7] https://huggingface.co/facebook/bart-base

Discourse-Split Baseline (BL_{dsplit}) pretrained BART fine-tuned on a discourse-focused subset of WikiSplit (D-WikiSplit). This baseline is to be directly compared with BL_{split}.

Parser Pipeline Baseline (PL_{Parse}) A pipeline of two models. The first uses the discourse parser process used to generate Ts from Cs in Section 4.4 (C2T) and the second is a pretrained BART fine-tuned on $\langle T, S \rangle$ data (T2S). We experimented training the T2S component on various datasets and found the best to be that trained purely on synthetic data. Thus, any pipeline mentioned in the remainder of this chapter refers to a specific C2T component connected to this same T2S component. This baseline allows us to compare pipeline models whose C2T component is learned on the split data vs. one where the C2T component uses an existing discourse parser.

End-to-End Model (E2E) pretrained BART fine-tuned on discourse-split data. We report results for variants trained on D-CC-News-S ($E2E_{Synth}$), D-Wikisplit and D-MUSS ($E2E_{Organic}$), and all three combined ($E2E_{both}$).

Pipeline Model (PL) A pipeline of two models. The first model is pretrained BART fine-tuned on $\langle C, T \rangle$ data and the second is a pretrained BART fine-tuned on $\langle T, S \rangle$ data from D-CCNews-S. We report results for pipelines with a C2T component trained on all D-CCNews data (PL_{Synth}), D-MUSS data ($PL_{Organic}$), and D-CCNews combined with D-Wikisplit and D-MUSS data (PL_{both}).

pretraining and Fine-tuning (PT+FT) pretrained BART fine-tuned on one data set before being further fine-tuned on another. We try training first on either synthetic or standard WikiSplit data and then fine-tuning on D-WikiSplit and D-MUSS data. Using WikiSplit for the first step was found to be the best performing configuration for the end-to-end system ($E2E_{ptft}$), while using D-CCNews proved better for the pipeline (PL_{ptft}).

Multi-Tasking (MTL) We prefix the training data with a control token indicating whether a training instance maps a complex input to a discourse tree (c2t), a discourse tree to a split text (t2s) or a complex input to a split output (c2s) and train pretrained BART on this data. We use training examples from D-CCNews, D-WikiSplit and D-MUSS. At inference time, we prefix the input with the c2s control token for the end-to-end model; and with the c2t and t2s control tokens for the two components of the pipeline model.

2.5.1 Evaluation Metrics

As illustrated in Table 2.4, variants of a discourse split may differ in terms of sentence order, discourse connective and rephrasing. To account for such variants while automatically assessing meaning preservation and discourse structure in the generated output, we use a combination of metrics.

Meaning Preservation. We measure meaning preservation using BLEU and SAMSA. We calculate BLEU scores (Papineni et al., 2002) between the ground-truth reference and the generated text using the SacreBLEU library (Post, 2018). We use the EASSE python library (Alva-Manchego et al., 2019a) to compute SAMSA scores. SAMSA (Sulem et al., 2018b) aims to put more focus on the structural aspects of the text, by leveraging a semantic parser. It observes changes made to predicate-argument structures, and thus for the sentence *"John got home and gave Mary a call."*, a higher score will be given to *"John got home. John gave Mary a call."* than for *"John got home and gave. Mary called."*. This indicates whether a model actually produces semantically coherent splits irrespective of whether a valid discourse connective and ordering is

used.[8]

Discourse Structure. To evaluate discourse structure we compute connective-, relation- and discourse-structure accuracy. Connective-accuracy (Conn-ACC) is the proportion of cases in which the generated text contains the same adverbial as the reference and relation-accuracy (Rel-ACC) the proportion of cases which maintain the discourse relation. The difference between Rel-ACC and Conn-ACC indicates how well the model is able to generalize amongst equivalent connectives of the same relation.

We also introduce a custom binary metric (D-ACC) which classifies an output as positive if (i) the correct discourse relation is maintained, (ii) the sentences are correctly ordered, and (iii) there is sufficient semantic similarity between the generated text and the ground-truth. A text will have a D-ACC score of 1 if it has a high BLEU (BLEU > 0.5) and either a low sentence BLEU (S-BLEU < 0.1) with a discourse adverbial which reverses the order of the argument (Table 2.4, Ex. 2) or a high sentence BLEU and a discourse adverbial which preserves the input discourse relation (Table 2.4, Ex. 2 and 3). Conversely, outputs with low BLEU and outputs with high BLEU, low S-BLEU and the same discourse connective as the reference (Table 2.4, Ex. 4) will be assigned a score of 0.[9]

We treat SAMSA and D-ACC as our primary metrics for comparing performance between models as, together, they provide an evaluation of both the meaning preservation and coherence of the split as well as the preservation of the discourse structure. Table 2.4 shows several example outputs and their corresponding scores.

2.5.2 Human Evaluation

In addition to using automated metrics, we performed human evaluation to compare our highest performing models and baseline systems. We considered a subset of 96 randomly selected examples from our test set (12 from each discourse relation type) and presented human annotators with the generated text for that example from our best performing pipeline system (PL_{Synth}) and asked them to compare it with (a) the result from BL_{split} (trained on generic split data), (b) the ground-truth result with adverbial removed, and (c) the result from our best performing end-to-end model ($E2E_{both}$). Each combination was presented to 10 different annotators who were asked to compare the two texts in terms of their grammaticality, as well as how similar in meaning they are to the C input.

We perform evaluation via the crowdsourcing platform, Amazon Mechanical Turk. We present a web form to evaluators, which includes some example texts and questions they must answer. These forms are referred to as *hits*. In our case, each hit contains three pieces of text (A, B, and X). These are the output from PL_{Synth} for a given test example, the output from one of our three comparators (BL_{split}, $E2E_{both}$, and the ground-truth with no adverbial) for the same example, and the input C, respectively.

Evaluators are then asked to answer the following questions:

- Which text (A or B) has more grammatical/fluent/well-formed English?
- Which text (A or B) is most similar in meaning to X?

[8] Despite SAMSA specifically targeting minimal units while our systems aim to only perform a single split, we believe it is sufficient here as all outputs should contain the same number of sentences and therefore would receive the same non-split penalty.

[9] We determined the above thresholds of 0.5 and 0.1 empirically via the manual examination of a number of test examples. The S-BLEU threshold is much lower because when the argument ordering is reversed we expect there to be little to no n-gram overlap with the ground-truths.

	Text	BLEU	S-BLEU	SAMSA	D-ACC
Ref.	The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell. **After this**, Mindaugas crossed the Vistula river and captured the fortress of Jazdów.				
✓1:C	The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell. **Afterwards**, Mindaugas crossed the Vistula river and captured the fortress of Jazdów.	89.11	92.80	66.66	1
✓2:O,C	Mindaugas crossed the Vistula river and captured the fortress of Jazdów. **Before this**, the Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell.	84.93	3.30	66.66	1
✓3:T	The Masovians were caught by surprise, since the capital, Płock, fell. **After this** had happened, Mindaugas then crossed the Vistula river and captured the fortress of Jazdów.	73.64	65.96	66.66	1
✗4:O	Mindaugas crossed the Vistula river and captured the fortress of Jazdów. **After this**, the Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell.	89.56	4.95	66.66	0

Table 2.4: Example illustrating how correct and incorrect variants of the reference impact the scores. O indicates that the order of the sentences has been reversed, C that the discourse adverbial differs from that used in the reference, and T that the text has changed. Only D-ACC distinguishes good from bad variants.

For each of the two questions they must answer with either *A*, *B*, or *Equal*. For each of the 96 selected test examples, we performed 3 model comparisons and sourced 10 separate evaluators for each, meaning we had 2,880 hits completed. Each of these hits gives us 2 judgements (one for grammaticality and one for meaning preservation), thus we received 5,760 individual judgements. We paid $0.06 USD for each hit, meaning we spent $172.8 USD in total. An example of how one of these hits looks to evaluators can be seen in Figure 2.1.

We do not compute inter-annotator agreement scores due to some of the complexities in using the crowdsourcing platform. Specifically, it would require having every annotator complete every comparison task, which is hard to manage at scale when posing each comparison as an individual task. To mitigate this issue, we opted to have a larger number of annotators complete each task, coupled with a larger number of unique tasks, in an attempt to smooth out individual differences.

2.6 Results and Discussion

Table 2.5 summarizes the results.

Pipeline vs. End-to-End. While no single configuration outperforms all others, PL_{Synth} ranks high for meaning preservation (SAMSA and BLEU) and for discourse structure (D-ACC and D-Rel). More generally, we see that *PL* models universally outperform their *E2E* variant in terms of discourse structure (Rel- and D-ACC). Conversely, the *E2E* models tend to show better results in terms of meaning preservation (SAMSA and BLEU). This suggests that while the *PL* models are good at producing valid connectives and the correct sentence order (high D-ACC), their generative capacity could be improved.

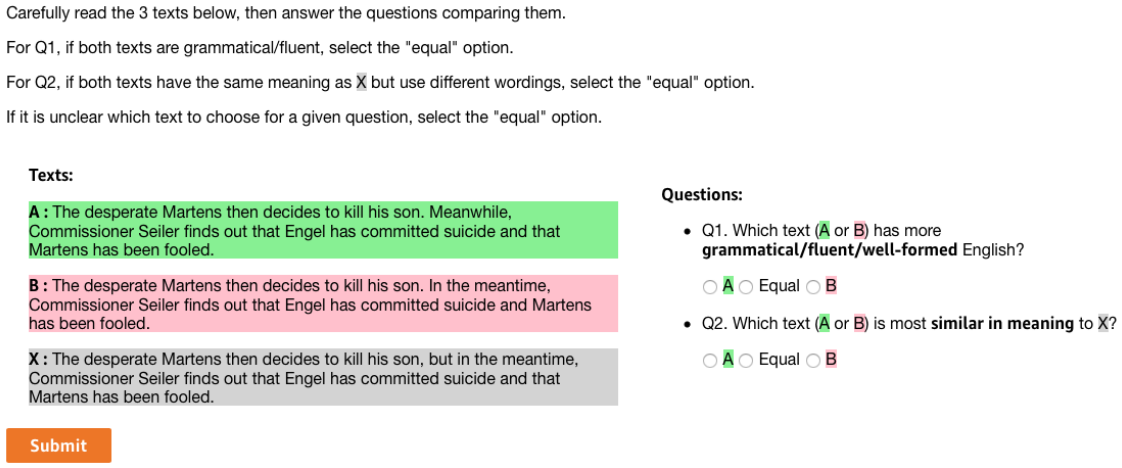


Figure 2.1: An example human evaluation hit for a single test example.

Model	Data	SAMSA		BLEU		Discourse Structure					
						Rel		Conn		D-ACC	
		E2E	*PL*	*E2E*	*PL*	*E2E*	*PL*	*E2E*	*PL*	*E2E*	*PL*
PL_{Parse}			46.37		67.25		0.73		0.31		0.65
BL_{dsplit}	D-WikiSplit	53.91	50.27	80.16	71.65	0.46	0.59	0.43	0.26	0.45	0.51
BL_{split}	WikiSplit	**54.15**		80.09		0.45		0.44		0.45	
	Synth	47.82	49.96	80.90	72.98	**0.57**	**0.69**	0.45	**0.31**	**0.55**	**0.64**
	Organic	53.26	48.15	80.00	68.90	0.47	0.61	0.43	0.27	0.46	0.55
	Both	52.96	50.40	81.31	72.87	0.55	0.63	0.44	0.27	0.54	0.60
PT+FT	Synth/Org	53.97	49.99	**81.64**	73.59	0.50	0.60	**0.47**	0.24	0.50	0.57
MTL	C/T,T/S,C/S	44.97	**52.93**	74.67	**75.55**	0.45	0.52	0.39	**0.31**	0.44	0.51

Table 2.5: A summary of results. Each row represents the results of the best E2E and *PL* model for the specified data category.

Synthetic vs Organic Data. Another clear trend is that models trained with synthetic data have significantly higher D-ACC than those trained with organic data. This confirms our hypothesis that, because it includes multiple variants of the same discourse split using different connectives and orderings, the synthetic data helps to train discourse splitting models that are better able to generalize, i.e. able to generate with different connectives for the same relation.

For both *E2E* and *PL* models, combining organic and synthetic data ($E2E_{both}$ and PL_{both}) appears to reduce the performance trade-off of using one data type in isolation. Alternative ways of combining organic and synthetic data using either fine-tuning and pretraining or multi-tasking did not yield improvements. For both regimes, we experimented with multiple hyper-parameters and data combinations. The details of these experiments are given in Appendix A.1.

Generic- vs. Discourse-Split Data In terms of meaning preservation (BLEU, SAMSA), BL_{dsplit} (trained on 371K instances) performs on par with BL_{split} (1M instances), showing that discourse-focused models can compete with standard splitting models when trained on much smaller, dedicated datasets. Moreover, in terms of discourse structure (D-ACC) and generalization (Rel-ACC, Conn-ACC), $E2E_{Organic}$ has significantly higher generalization capacity than

BL_{split} ($p = 0.046$). This improvement becomes more dramatic when also including the synthetic data ($E2E_{both}$) ($p = 8.72e^{-7}$).

Human Evaluation The results from the human evaluation (Table 2.6) essentially confirm those of the automatic evaluation. Human annotators find the output of PL_{Synth} less grammatical and meaning preserving than either of the end-to-end models ($E2E_{both}$ and BL_{split}). This corroborates the divergence seen between *E2E* and *PL* models for SAMSA and BLEU scores.

For meaning preservation, annotators more often selected PL_{Synth} over BL_{split} than they did PL_{Synth} over $E2E_{both}$ ($p = 0.138$), strengthening the observation that discourse-focused models perform this task better than generic splitting models.

PL_{Synth} produces texts that are equally grammatical yet significantly more meaning preserving ($p = 0.017$) than the adverbial-stripped ground-truths. This reinforces the importance of maintaining discourse coherence when performing sentence splitting.

Upon examination of human evaluations, we found that annotators often marked the less grammatical text as being less meaning preserving by default. When controlling for this and only considering cases where both texts were labelled as equally grammatical (bottom tier of Table 2.6), we see improved results for PL_{Synth} such that, in terms of meaning preservation, there is less difference between PL_{Synth} and the end-to-end models and an increased difference between PL_{Synth} and the ground-truth with adverbial removed.

Models	**Grammaticality**			**Meaning Pres.**		
	$>$	$=$	$<$	$>$	$=$	$<$
PL_{Synth} vs. $E2E_{both}$	0.21	0.40	0.39	0.15	0.47	0.38
PL_{Synth} vs. BL_{split}	0.24	0.34	0.42	0.18	0.44	0.39
PL_{Synth} vs. no adv.	0.36	0.29	0.36	0.35	0.35	0.30
PL_{Synth} vs. $E2E_{both}$				0.06	0.81	0.14
PL_{Synth} vs. BL_{split}		$=$		0.09	0.77	0.14
PL_{Synth} vs. no adv.				0.25	0.64	0.11

Table 2.6: Results for human evaluation. Cells show the proportion of cases where the pipeline was deemed better, equal or worse than a particular baseline.

Qualitative Analysis In addition to the automatic and human evaluations, we perform a qualitative analysis of common mistakes seen in system outputs. Table A.1 in Appendix A.2 shows some examples of common errors for PL_{Synth}, $E2E_{both}$ and BL_{split}. We can group these mistakes into 4 broad categories: *connective*, *content*, *splitting*, and *hallucinations*. Connective errors are those that use an incorrect connective or lack one entirely. Content errors are cases where the semantic content of the input is not maintained in the output. Splitting errors are cases where splitting has not been performed or has been carried out in the wrong place. We also occasionally see hallucinations where the output has included out-of-context information.

The BL_{split} model will often fail to use a valid adverbial, instead merely splitting the sentence at the position of the connective. We believe this is due to it not fully learning to maintain the discourse relation. It has also been observed to include hallucinated terms in the output.

We commonly see splitting errors for both *PL* and *E2E* models. The *PL* often splits at a position containing a known connective term, but where it is not acting as a connective given the context. This is due to the intermediary task incorrectly segmenting the input, possibly as a result of parser mistakes in the training data. On the other hand, the *E2E* will sometimes not perform any split, particularly where certain grammatical markers (e.g. semicolons) are present.

This is likely because there is no intermediary task enforcing a split operation.

2.7 Conclusion

In this chapter we introduced the task of discourse-based sentence splitting together with a large-scale dataset of both organic and synthetic discourse splits. Experimental evaluation revealed that discourse-based, pipeline models have better discourse relation preservation capabilities than end-to-end models, and that leveraging larger synthetic datasets is critical for learning models that can generalize (i.e. can generate multiple variants of the same discourse relation) but comes at the cost of less fluent outputs. In the next chapter, we will propose a controllable sentence simplification system that is capable of routinely performing discourse-splitting along with a diverse range of other operations.

There are some limitations of this current method, mostly stemming from our assumption that discourse relations are generally signalled by explicit markers. For example, implicit relations can often be inferred where there is no discourse connective and multiple relations could potentially exist within a single sentence (Webber, 2023). Currently our approach is not able to account for these possibilities, and so future directions could include extending the framework to support such cases.

Chapter 3

Controllable Sentence Simplification via Operation Classification

Contents

Different types of transformations have been used to model sentence simplification ranging from mainly local operations, such as phrasal or lexical rewriting, deletion and re-ordering to the more global affecting the entire input sentence such as sentence rephrasing, copying and splitting. In this chapter, we propose a novel approach to sentence simplification which encompasses four global operations: whether to rephrase or copy and whether to split based on syntactic or discourse structure. We create a novel dataset that can be used to train highly accurate classification systems for these four operations and propose a controllable simplification model that tailors simplifications to these operations. We show that it outperforms both end-to-end, non-controllable approaches and previous controllable approaches.

3.1 Introduction

Modern simplification systems are data-driven, learning to perform transformations from parallel corpora of complex-simple $\langle C, S \rangle$ pairs. Although many different approaches have been attempted in the past, including statistical machine translation (SMT)-based methods, nearly all systems proposed in recent years follow a neural sequence-to-sequence approach. As these systems are trained in an end-to-end manner they are able to perform lexical and syntactic operations in combination and produce outputs with very high fluency.

However, given the black-box nature of these end-to-end systems, they are forced to rely on imperfect training corpora to implicitly learn rewrite operations, many of which occur infrequently (Jiang et al., 2020). As a result, neural end-to-end systems have been found to be overly conservative, often making no changes to the original text or being limited to the paraphrasing of short word sequences (Alva-Manchego et al., 2017; Maddela et al., 2021). Much like what is discussed in Chapter 2 in the context of sentence splitting, these general simplification systems also provide limited capacity for controllability and are unable to express alternative variants of the simplified text (Alva-Manchego et al., 2017). In response, attempts have been made to produce controllable simplification systems that can constrain either the shape (length, amount of paraphrasing, lexical and syntactic complexity) of the output (Martin et al., 2020) or the type of transformation to be applied (e.g., copy, split, merge, rewrite, etc.) (Scarton and Specia, 2018; Dong et al., 2019; Scarton et al., 2020; Garbacea et al., 2021; Maddela et al., 2021).

In this chapter we propose an approach to sentence simplification that encompasses four global operations: whether to copy the input sentence (no simplification needed), rephrase it, split it based on syntax, or split it based on discourse structure (as explored in Chapter 2). We create a novel dataset that can be used to train highly accurate classification systems for these four operations and propose a controllable-simplification model that tailors simplification to them. We compare our model with various alternatives and previous work, using both quantitative metrics and human evaluation, and show that our model outperforms them. We also provide a qualitative analysis of the differences between the best models.

3.2 Related Work

3.2.1 Controllable Simplification

Scarton and Specia (2018), Nishihara et al. (2019) and Scarton et al. (2020) focus on tailoring outputs to specific reader groups based on the Newsela corpus (Xu et al., 2015), a popular simplification dataset which provides versions of news articles written for audiences of different reading levels. These works propose systems that adjust their simplifications to match one of these reading levels. Martin et al. (2020) introduce a wider array of control attributes concerning grammatical features of the desired text such as compression level, amount of paraphrasing, and lexical and syntactic complexity.

Most recently, Maddela et al. (2021) propose a system that first uses a rule-based component (Niklaus et al., 2019b) to generate candidates that have undergone splitting and deletion, before ranking them and sending the top n to a neural paraphasing model. Tunable settings in both components provide control over how much of the input is changed and whether to favour deletion or splitting. Their system received higher fluency and simplicity scores from human annotators compared to existing works.

However, at inference time, these methods all require the model to be explicitly informed about which reading level to cater to or which specific grammatical features or rewrite operations

to prioritize. In contrast, we develop an approach that can not only be tuned manually, but can also operate in an end-to-end manner by inferring tunable parameters from the input.

3.2.2 Operation Classification

Alva-Manchego et al. (2017) and Dong et al. (2019) consider sentence simplification as a sequence-labeling problem, proposing systems that predict rewrite operations at the token-level before realizing them downstream. Alva-Manchego et al. (2017) showed gains over previous approaches in terms of simplicity, but at the cost of fluency and meaning preservation. Dong et al. (2019) appears to resolve this trade off by introducing an enhanced interpreter that better constructs the resulting text.

Several existing works have attempted to use a classifier to determine which rewrite operation should be performed on an input at the sentence-level. Applying a sentence-level binary classifier as an initial step to predict whether simplification should be performed has been found to yield improved SARI results, reducing conservatism and spurious transformations (Scarton et al., 2020; Garbacea et al., 2021). Multi-class variants of such systems have been explored with limited success. Scarton and Specia (2018) and Scarton et al. (2020) predict one of 4 operations (identical, elaboration, split, and merge) and feed this into an end-to-end model alongside the C as either a control token or one-hot vector. While Scarton and Specia (2018) fail to produce an accurate classifier or show any improvement over baselines, Scarton et al. (2020) show some gains in SARI when using predicted operation labels. However, their best classifier only yields an accuracy of 70%.

In the multi-class setting, models tend to struggle to accurately predict identity cases (full sentence copying). We believe this is partially due to the training data used. All existing works use Cs from identical $\langle C, S \rangle$ pairs as training examples for this class, either alone or alongside standard Ss. The assumption made here is that these pairs are identical because the C is already simplified. We will show, however, that it is much more likely these items are unsimplified noise from the distribution of Cs and that excluding them from training data can dramatically improve accuracy.

We extend upon these sentence-level classification approaches by redefining the set of operations, creating comprehensive training and test data, and ultimately producing a classifier with much higher accuracy. We show that a pipeline approach that first predicts a rewrite operation outperforms existing end-to-end and controllable systems.

3.3 Operation Classification

We consider 4 operation types: *identity*, *rephrase*, *syntax-split*, and *discourse-split*. The *identity* and *rephrase* classes are equivalent to *identical* and *elaboration* from Scarton et al. (2020). In contrast, we split the *split* class into two distinct groups to capture further nuances of sentence splitting, as was explored in Chapter 2.

Syntax-split indicates that a split should be performed based on a syntactic construct, whereas *discourse-split* indicates that a split should be performed based on a discourse relation.

As we focus on single sentence simplification, we exclude the *merge* class used in Scarton and Specia (2018); Scarton et al. (2020).

3.3.1 Training Data

We construct training data for a simplification operation classifier by combining subsets of existing English datasets. We consider simplification datasets Wiki-auto, Newsela-auto[10] (Jiang et al., 2020), and MUSS (Martin et al., 2021) as well as dedicated splitting datasets WikiSplit (Botha et al., 2018) and D-CCNews (Chapter 2).

Wiki-auto and Newsela-auto are automatically aligned $\langle C, S \rangle$ pairs extracted from Wikipedia and Newsela, respectively. MUSS contains 2.7M pairs mined from Common Crawl web data which are estimated paraphrases based on embedding distance. WikiSplit contains 1M split pairs mined from Wikipedia edit history, while D-CCNews contains discourse-split pairs mined from the CCNews corpus (Nagel, 2016). As a reminder, D-CCNews has two subsets: D-CCNews-C which contains single Cs, and D-CCNews-S which contains pairs of organic Ss and synthetic Cs (we include samples from both subsets). Table 3.1 provides a breakdown of the inclusions from each source.

	Source						
Class	WikiSplit	MUSS	Wiki-auto	Newsela-auto	D-CCNews-C	D-CCNews-S	**Total**
Identity	-	-	513,436	338,798	-	-	**852,234**
Rephrase	-	461,702	366,382	171,508	-	-	**999,592**
Syntax-Split	633,900	53,008	68,357	88,669	-	-	**843,934**
Discourse-Split	269,666	1,002	5,277	2,060	250,062	249,958	**778,025**
Total	**903,566**	**515,712**	**953,452**	**601,035**	**250,062**	**249,958**	**3,473,785**

Table 3.1: Data source distributions for each operation class in IRSD_4^C.

We heuristically assign silver operation labels to sentences from these datasets as follows:

- **identity:** Ss from the Wiki-auto and Newsela-auto *rephrase* and *syntax-split* sets. We assume that the simple sentences (Ss) from known simplification datasets are sufficiently simplified.
- **rephrase:** Cs from MUSS, Wiki-auto and Newsela-auto where there is no split in the output S and Levenshtein similarity between the C and S is less than 1 standard deviation above the mean (< 0.92). This is to exclude near-identical pairs.
- **syntax-split:** Cs from WikiSplit, MUSS, Wiki-auto and Newsela-auto whose S exhibits a split and does not contain an identifiable discourse marker.
- **discourse-split:** Cs from all datasets whose S contains a split and a discourse adverbial.[11]

We call the resulting dataset $\textbf{IRSD}_4^C$.[12] We also consider a 3-class subset which excludes the *identity* class ($\textbf{IRSD}_3^C$) to explore how results change when models are trained to always simplify.

[10] We specifically use the aligned pairs used for simplification experiments in Jiang et al. (2020), which excludes identical pairs and those of readability levels 0-1, 1-2, and 2-3.

[11] D-CCNews is down-sampled to keep classes similar in size.

[12] Our data and code is available at `https://github.com/liamcripwell/control_simp`. Newsela data is excluded, subject to their terms of use, but can be provided upon request after receiving a license.

3.3.2 Test Data

We use two datasets for evaluation. A random sample of 1% of the training data is set aside as a large (34K examples) silver test set. We also create a smaller gold test set by randomly sampling 100 items from each of the 4 classes in our silver test set and presenting them to 3 annotators instructed to select the most appropriate operation with which to simplify the text. These annotators were students enrolled in a local NLP master's degree program.

Annotations were completed through a web form interface (e.g. in Figure 3.1). For each of the 400 items, they were presented with the input sentence and required to select one of the four class labels. They were also given the option to flag examples as being malformed or incomprehensible (which we removed from the final set). Prior to their completion of the task, they were given a detailed description of each class along with a range of examples.

We approved all annotations that received a majority label agreement and manually adjudicated cases where all annotators disagreed (11%). The mean Cohen's Kappa agreement score between annotators is 0.246, illustrating the difficulty of this task. In many cases, several operations could feasibly apply, and so assigning a single correct label is not always a perfect solution. Table 3.2 lists some examples of this kind.

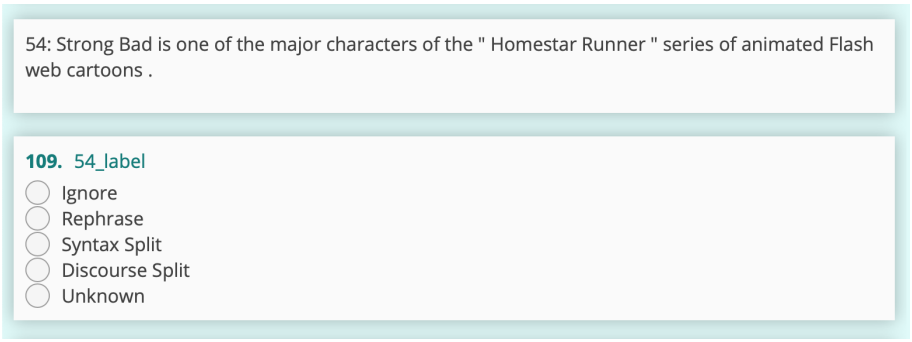


Figure 3.1: Section of annotation form used for gold-label classification test set creation.

3.3.3 Classification Model

We fine-tune pretrained RoBERTa models (Liu et al., 2019) with classification heads on IRSD_4^C and IRSD_3^C.[13] The network used contains 12 hidden layers, a hidden size of 768, and was pretrained with the masked language modeling objective on 160GB of books and web content. We used a learning rate of $2e^{-5}$, a batch size of 32, and performed dropout with a rate of 0.1 and early stopping as regularization measures.

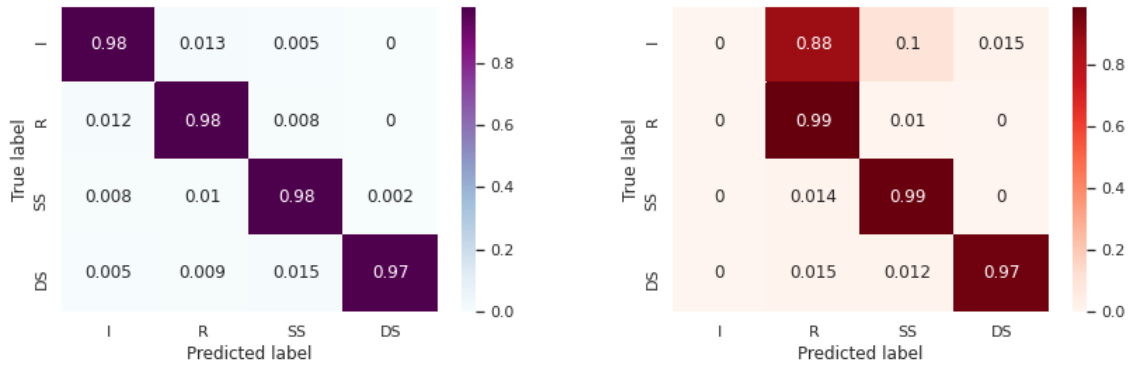
Results on Silver Test Data. Results can be seen in Figure 3.2. Accuracy on the silver test set (98%) is much higher than previous works: Scarton and Specia (2018) and Scarton et al. (2020) achieve mean accuracies of 51% and 70% for a similar 4-class task. Garbacea et al. (2021), who only train a binary (*simp*, *no-simp*) classifier achieve 81% accuracy.

Notably, the accuracy for the *identity* class is much higher than the 59% achieved by Scarton et al. (2020). This is perhaps in part due to our exclusion of Cs from identical $\langle C, S \rangle$ pairs in the *identity* training subset. We explored this hypothesis by using a test set containing Cs from identical pairs alongside the existing *identity* examples. Figure 3.3 shows that doing so reduces performance on the *identity* class dramatically; the model only classifies 9.8% of these

[13] We use the pretrained `roberta-base` model available at https://huggingface.co/roberta-base.

C1.	He served as Mayor of The Hague from 2008 to 2017; he then took two acting positions in Drenthe and Amsterdam.
rephrase	He was Mayor of The Hague from 2008 to 2017 then took two acting positions in Drenthe and Amsterdam.
syntax-split	He served as Mayor of The Hague from 2008 to 2017. He then took two acting positions in Drenthe and Amsterdam.
discourse-split	He served as Mayor of The Hague from 2008 to 2017. Later, he took two acting positions in Drenthe and Amsterdam.
C2.	A bus stop is a designated place where buses stop for passengers to get on and off the bus.
identity	A bus stop is a designated place where buses stop for passengers to get on and off the bus.
rephrase	A bus stop is a place where buses stop for passengers.
C3.	He led Villa to victory in the inaugural League Cup in 1961 but was then sacked in 1964 on grounds of ill health.
syntax-split	He led Villa to victory in the inaugural League Cup in 1961. He was sacked in 1964 on grounds of ill health.
discourse-split	He led Villa to victory in the inaugural League Cup in 1961. However, he was sacked in 1964 on grounds of ill health.

Table 3.2: Some complex sentence examples where multiple rewrite operations are plausible.



(a) 4-class (b) 3class

Figure 3.2: Normalized confusion matrix of (a) the four-class classifier and (b) the three-class classifier, evaluated on the silver-label test set.

Cs as *identity* and 82.4% of them as *rephrase*. This suggests that these examples are from a distribution more similar to the *rephrase* examples and are possibly complex sentences themselves that have not been fully simplified in the source data. We believe this observation validates our decision to exclude them from consideration.

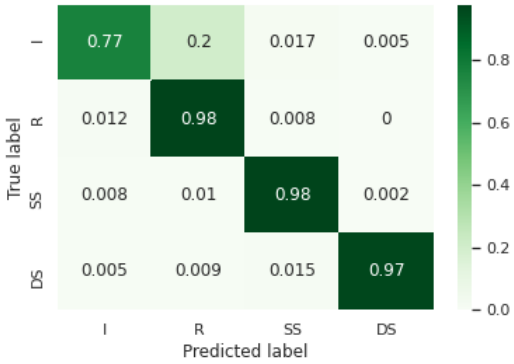
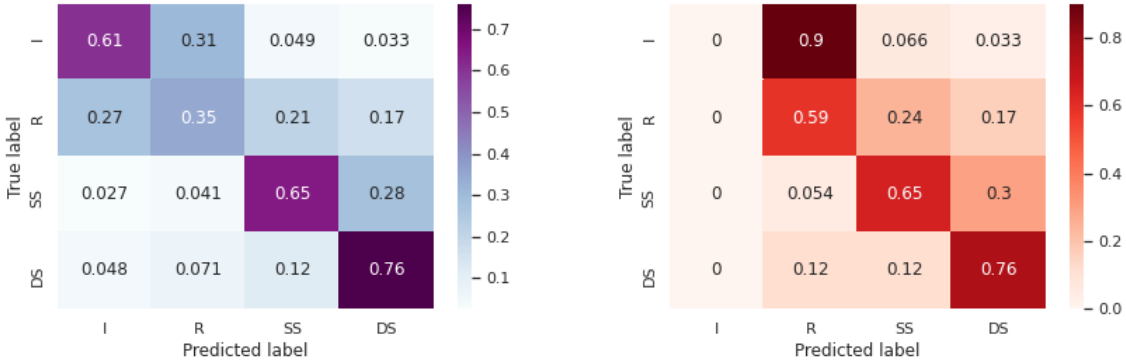


Figure 3.3: Normalized confusion matrix for the 4-class operation classifier, evaluated on the silver-label test set containing Cs from identical $\langle C, S \rangle$ pairs in the *identity* class.

Results on Gold Test Data. As shown in Figure 3.4, classification accuracy on the gold test set is considerably lower than on the silver data. *Identity* examples are often predicted as *rephrase*; *syntax-split* often as *discourse-split*; and *rephrase* examples regularly receive predictions across all four classes. However, this aligns with our observations with respect to manual labelling difficulties. Often it is not immediately clear whether a particular example should be ignored or slightly rephrased. Similarly, it often seems plausible for either type of split to be performed. *Rephrase* is the broadest of the four classes, and so cases where any one of the other three classes could also apply should be expected. Despite being lower than on the silver examples, we believe these results show a strong signal of acceptable performance, with common mistakes being analogous to difficulties encountered by human annotators.



(a) 4-class (b) 3-class

Figure 3.4: Normalized confusion matrix of (a) the four-class classifier and (b) the three-class classifier, evaluated on the human-annotated test set.

3.4 Sentence Simplification

3.4.1 Data

Training Data. For the sentence simplification task we use a modified version of IRSD^C which additionally includes target simplifications, i.e. $\langle C, o, S \rangle$ triples. We refer to this as **IRSD**$^S_4$ and its 3-class subset as **IRSD**$^S_3$.

For the *identity* class, we take all inputs from **IRSD**$^S_4$ labelled as *identity* and map them to themselves. For *rephrase* and *syntax-split*, we take the *rephrase* and *syntax-split* inputs and map them to their simplifications in the source datasets. We do the same for *discourse-split*, but, as D-CCNews-C instances do not contain simplifications, we replace them with additional $\langle C, S \rangle$ pairs from D-CCNews-S.

Test Data. We first train and test our systems on IRSD$^S_{3/4}$ and Newsela-auto. Next, in order to compare with past works, we perform evaluation on the Newsela-auto test set introduced by Maddela et al. (2021). It contains 24,035 *rephrases*, 9,208 *syntax-splits*, and 148 *discourse-splits*. We refer to this as **Newsela-M** and also include results on the subset with split in their reference S (Newsela-M (Split)). We use this test set so we can include pre-existing system outputs from past works for comparison.

Additionally, we evaluate on the ASSET corpus (Alva-Manchego et al., 2020a) which is a much smaller test set (359 examples) containing 10 human-written references per input. All test examples have at least one *rephrase* reference, 248 have at least one *syntax-split* reference, 12 have at least one *discourse-split* reference, and 0 have an *identity* reference.

Model	**IRSD**$^S_4$				**IRSD**$^S_3$		**Newsela-auto**			
	P_{BERT}	SARI	R_{Split}	P_{Split}	P_{BERT}	SARI	P_{BERT}	SARI	R_{Split}	P_{Split}
Input	0.83	27.4	0.00	0.00	0.77	25.4	0.53	15.9	0.00	0.00
Reference	0.99	80.1	1.00	1.00	0.99	95.3	0.99	94.1	1.00	1.00
End-to-End Models										
BART$_W$	0.81	35.0	0.18	0.85	0.76	34.7	0.54	24.6	0.05	**0.64**
BART$_N$	0.77	38.9	0.64	0.81	0.74	42.0	**0.56**	**35.9**	0.46	0.59
BART$_3$	0.85	50.6	0.82	0.94	0.81	54.9	0.55	27.3	0.27	0.59
BART$_4$	0.86	51.2	0.85	0.93	0.82	55.7	**0.56**	26.9	0.21	0.62
Controllable Models with predicted control-tokens										
Ctrl$_{3,3}$	0.83	50.6	**0.99**	0.93	0.82	58.5	0.54	33.6	0.48	0.54
Ctrl$_{3,4}$	0.84	51.2	**0.99**	0.93	**0.83**	59.4	0.55	**35.9**	**0.49**	0.54
Ctrl$_{4,3}$	0.86	52.9	**0.99**	**0.98**	**0.83**	59.5	0.55	30.7	0.45	0.56
Ctrl$_{4,4}$	**0.87**	**55.1**	**0.99**	**0.98**	**0.83**	**60.4**	**0.56**	32.4	0.45	0.56
Controllable Models with Oracle control-tokens										
Ctrl$_{Oracle}$	**0.87**	**55.5**	**1.00**	**1.00**	**0.83**	**60.7**	**0.57**	**38.3**	**0.99**	**1.00**

Table 3.3: Automatic sentence simplification results on the IRSD$^S_4$, IRSD$^S_3$ and Newsela-auto test sets.

3.4.2 Models

Existing Systems. We consider a number of past works for comparison: (i) **Hybrid** (Narayan and Gardent, 2014), an older system with a probabilistic splitting component combined with an

MT-based lexical paraphraser; (ii) **BERT**, pretrained encoder-decoder transformer (BERT$_{base}$) fine-tuned on simplification, which achieved state-of-the-art performance (Jiang et al., 2020); (iii) **EditNTS** (Dong et al., 2019), a recent model using token-level operation prediction; and (iv) **MadExP** (Maddela et al., 2021), current state-of-the-art controllable system.[14] We exclude other systems which require conditioning on specific reading levels.

Baseline End-to-End Model. We include end-to-end baselines that are trained to perform $C \rightarrow S$ with no additional information. These are used to gauge whether our controllable models are competitive with a black-box approach. We use the BART architecture (Lewis et al., 2020) and fine-tune a pretrained model with a language-modelling head on $\langle C, S \rangle$ pairs from 4 distinct datasets: IRSD$_4^S$ (**BART**$_4$), IRSD$_3^S$ (**BART**$_3$), Wiki-auto (**BART**$_W$) and Newsela-auto (**BART**$_N$).[15]

Controllable Model. Next, we train an end-to-end generative model to perform $\langle C, o \rangle \rightarrow S$, where o is an operation label. The o is used as a control token prepended to the input sequence for C. We use the same BART architecture as our end-to-end baselines.

From this model, we construct several systems: (i) an oracle baseline (**Ctrl**$_{Oracle}$) taking the silver operation label and performing generation as an end-to-end task; (ii) a pipeline system using a classifier to predict o before running the generative model.

We refer to different configurations as **Ctrl**$_{i,j}$, where i is the number of classes the classifier is trained on and j is the number of classes the generator is trained on. E.g. Ctrl$_{3,4}$ uses a classifier trained on IRSD$_3^C$ and a generator trained on IRSD$_4^S$.[16] We expect that using the 4-class classifier will result in more conservative outputs. Using the 3-class generator could allow more model capacity to focus on simplification. Conversely, the extra training data used by the 4-class generator could improve general performance.

3.5 Experimental Setup

3.5.1 Training Details

During training of the BART generative models, we used a learning rate of $3e^{-5}$. The network has 6 layers in each of the encoder and decoder, a hidden size of 768, and was pretrained to perform reconstruction of corrupted documents on a combination of books and Wikipedia data. All of our fine-tuning experiments used a batch size of 32, performed dropout with a rate of 0.1 and early stopping as regularization measures. As with the classifiers, all models were trained on a computing grid using 4 Nvidia RTX 2080 Ti GPUs (11GB memory). For each experiment we set aside 1% of the training set for validation. At test time we generate output sequences by performing beam search with a beam size of 5 and restrict output to a maximum length of 128 tokens.

3.5.2 Automatic Evaluation

The most common evaluation metrics used in text simplification are BLEU and SARI, with SARI being viewed as the more effective at describing simplicity. Both focus primarily on lexical similarities between the reference and system output without consideration for structural simplification. A recent meta-analysis of automated text simplification evaluation (Alva-Manchego

[14] We use system outputs from versions of all of these models that have been trained on Newsela-Auto.

[15] We use the pretrained `facebook/bart-base` model available at https://huggingface.co/facebook/bart-base.

[16] For Ctrl$_{4,3}$ any inputs classified as *ignore* are returned without being passed to the generator.

et al., 2021) shows that the precision-based BERTScore (P_{BERT}) (Zhang et al., 2019) is most highly correlated with human judgements. As P_{BERT} is very effective at identifying low quality simplifications, the authors recommend using it as a primary test of quality before referring to other metrics like SARI.

We report P_{BERT} and SARI as our primary metrics[17] and also use the split recall (R_{Split}) to evaluate how often the model performs splitting in known cases. We value recall over precision as it gives a better indication of whether a model regularly performs splits, but have also included the precision (P_{Split}) for clarity.

3.5.3 Human Evaluation

We perform a human evaluation of simplification systems by having 3 annotators evaluate outputs. In order to consider a range of structurally diverse examples we use our classifier to label the Newela-M test set with predicted operations and randomly select 25 from each of the 4 classes. The annotators were then presented with the input C from each $\langle C, S \rangle$ pair alongside the reference S and outputs from selected systems (e.g. interface in Figure 3.5).

Judgements are made with respect to 3 criteria: fluency, adequacy, and simplicity. Fluency refers to the grammaticality of the output; adequacy measures meaning preservation with respect to the input; and simplicity measures the overall simplicity of the result. We followed standard practice by having these criteria judged on a 1-5 Likert scale and averaging the results. For simplicity, we advised workers that a high score can be given to an output identical to the input if there is little to no *obvious* changes that would make the sentence simpler.

We consider the following systems for comparison: EditNTS, MadExp, BART_N, BART_4, and $\text{Ctrl}_{4,4}$. This allows us to compare our systems to strong recent works and examine the effect of (i) using IRSD^S vs Newsela training data and (ii) using our controllable model vs an end-to-end approach.

3.6 Results and Discussion

Automatic evaluation results are shown in Table 3.3.

IRSD^S vs Other Data Models trained with IRSD^S ($\text{BART}_{3/4}$ and $\text{Ctrl}_{*,*}$) greatly outperform those trained on other datasets ($\text{BART}_{N/W}$) across every metric on the IRSD^S test sets. On the Newsela test set, IRSD^S models perform at least as well as BART_N. This is unsurprising as IRSD^S is much larger than Newsela and contains many of the same examples. However, it shows the diversity of IRSD^S does not reduce Newsela-specific performance.

On Newsela test data, using the 3-class classifier ($\text{Ctrl}_{3,*}$) yields higher SARI and R_{Split} than the 4-class case. This is likely because *identity* is never predicted, thereby encouraging less conservative simplification on a test set where most examples are simplified (Maddela et al. (2021) exclude all examples with high or low similarity between the input and the reference from the test set).

End-to-End vs Controllable Controllable systems outperform their end-to-end counterpart on all metrics and datasets. In particular, they show a large increase in R_{Split}, suggesting that explicitly triggering splits via control tokens greatly improves a model's ability to correctly administer splits where needed. Using silver operation labels in Ctrl_{Oracle} leads to universally higher scores than classifier-based pipelines, indicating that there is still room for improvement in terms of classification performance.

[17] The EASSE python library (Alva-Manchego et al., 2019a) is used for calculation.

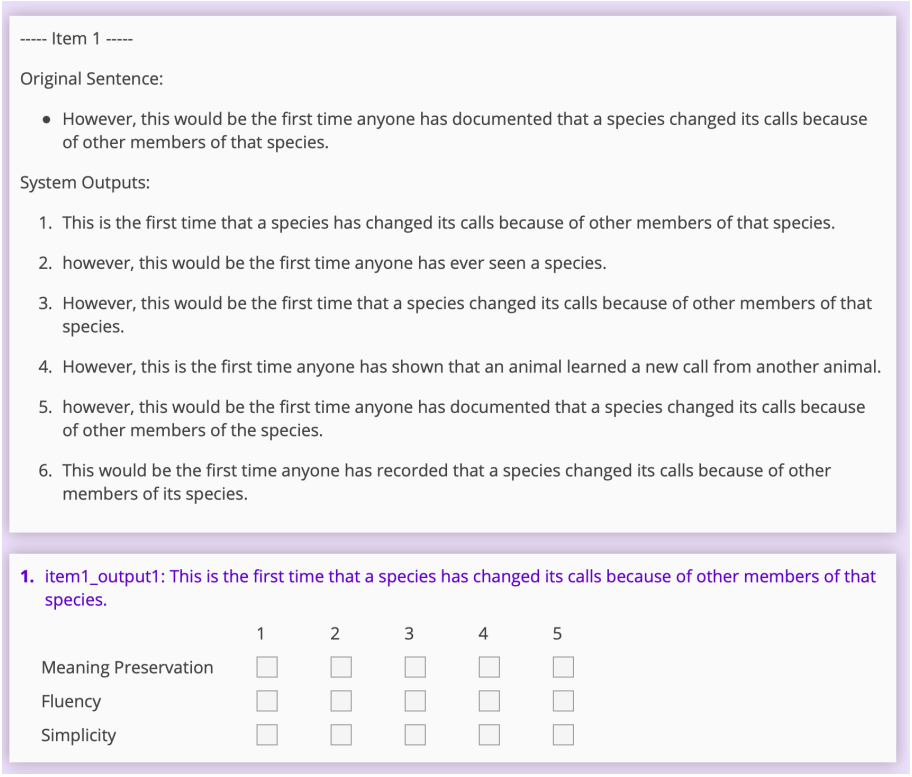


Figure 3.5: Section of annotation form used for human simplification evaluation.

Existing Systems Comparative results with existing systems are illustrated in Table 3.4. All of our systems achieve much higher P_{BERT} scores than any existing system. This suggests that merely using the BART architecture leads to much more fluent outputs than other models. MadExP, which receives the highest SARI scores, actually receives much lower P_{BERT} than almost any other system, indicating that although it may be simplifying the text well, outputs might be less fluent.

$\text{Ctrl}_{3,4}$ achieves the highest scores on Newsela-M, except for being slightly beaten by MadExP on SARI. Here, $\text{Ctrl}_{3,4}$ is expected to be better than $\text{Ctrl}_{4,4}$ as Newsela-M does not include any identical $\langle C, S \rangle$ pairs and therefore penalizes copying inputs. Ctrl_{Oracle} universally outperforms other systems.

On ASSET, BART_N actually achieves the highest SARI, closely followed by MadExP. We hypothesize that models trained on IRSD^S might achieve lower SARI as the training data includes examples from splitting datasets that do not exhibit any further simplification, leading to reduced lexical substitution when splitting is performed.

Human Evaluation Human evaluation results are shown in Table 3.5. $\text{Ctrl}_{4,4}$ scores highest in fluency and overall. Both $\text{Ctrl}_{4,4}$ and BART_4 score highest in adequacy. BART_N scores highest in simplicity. All of our systems are rated highly across all criteria and receive better average scores than even the references. This clearly highlights the capability of pretrained generative models like BART to produce highly fluent texts.

Further, we see that using a classifier trained on our data to predict operation-specific control-tokens further enhances performance in both fluency and simplicity. However, we believe our training data also limits simplicity at times due to the inclusion of aligned examples from pure

Model	**Training**	**Newsela-M**				**Newsela-M (Split)**		**ASSET**	
		P_{BERT}	SARI	R_{Split}	P_{Split}	P_{BERT}	SARI	P_{BERT}	SARI
Hybrid	Newsela-auto	0.39	30.2	0.17	0.42	0.39	31.9	0.43	30.5
BERT	Newsela-auto	0.46	32.2	0.40	0.46	0.47	34.5	0.59	35.2
EditNTS	Newsela-auto	0.49	29.3	0.32	0.45	0.53	30.8	0.54	31.4
MadExP	Newsela-auto	0.43	**36.0**	0.41	0.48	0.43	37.4	0.59	36.2
BART_N	Newsela-auto	0.54	34.0	0.52	0.48	0.58	37.1	**0.64**	**36.4**
BART_3	IRSD_3^S	0.54	25.0	0.31	0.49	0.58	28.8	**0.64**	34.3
BART_4	IRSD_4^S	**0.55**	25.3	0.25	**0.51**	0.58	28.6	**0.64**	33.7
$\text{Ctrl}_{3,3}$	IRSD_3^S	0.54	33.4	**0.54**	0.43	0.58	35.9	**0.64**	34.1
$\text{Ctrl}_{3,4}$	IRSD_4^S	**0.55**	35.6	**0.54**	0.43	**0.59**	**37.8**	**0.64**	33.8
$\text{Ctrl}_{4,4}$	IRSD_4^S	0.54	30.4	0.51	0.45	**0.59**	34.5	**0.64**	33.5
Ctrl_{Oracle}	IRSD_4^S	**0.56**	**37.3**	**1.00**	**0.99**	**0.59**	**38.6**	-	-

Table 3.4: Comparison with existing systems and baselines. Oracle labels are acquired by applying the same heuristics used in the creation of IRSD^S. Note that the oracle labels for these test sets do not contain *identity* cases.

splitting datasets, which exhibit no simplification edits but for the split itself. It is likely that the relatively low adequacy rating given to the references can partly be attributed to sentence alignment failures and cases where the S makes reference to terms mentioned earlier in their article that are not explicit in the C. To overcome this, it will be necessary to implement context-aware systems that are able to refer to information throughout the greater document. This will be the subject of the coming chapters.

System	Fluency	Adequacy	Simplicity	**Mean**
Ref.	4.65^{**}	3.95^{**}	4.37^{*}	4.32
EditNTS	3.81^{**}	3.83^{**}	3.91^{**}	3.85
MadExP	3.74^{**}	3.52^{**}	3.97^{**}	3.75
BART_N	4.68	4.26^{**}	$\mathbf{4.38}^{*}$	4.44
BART_4	4.71	**4.74**	4.14	4.53
$\text{Ctrl}_{4,4}$	**4.77**	**4.74**	4.20	**4.57**

Table 3.5: Human evaluation results for selected simplification systems and baselines. Ratings significantly different from $\text{Ctrl}_{4,4}$ are denoted with $*$ ($p < 0.05$) and $**$ ($p < 0.01$). Significance was determined with a Student's t-test.

Qualitative Analysis We perform a qualitative analysis of system outputs from the human evaluation to get a better idea of differences between our models. Table 3.6 illustrates common patterns.

BART_N regularly produces the most simple output, but often over-simplifies to the point of removing important contextual information (e.g. item 1). It also sometimes fails to maintain the correct meaning of the input (e.g. items 2 and 3). BART_4 often produces outputs very similar to $\text{Ctrl}_{4,4}$, but performs splitting much less regularly (e.g. items 1 and 4) which can uphold structural complexity.

$\text{Ctrl}_{4,4}$ outputs best retain the original meaning of the input. The benefit of having a classifier predict *identity* cases can be seen in item 2 where the other models end up rephrasing poorly

or deleting important information. However, when performing splits, $\text{Ctrl}_{4,4}$ sometimes fails to sufficiently rephrase, often keeping obviously complicated words (e.g. item 4).

	Output
(1) Complex	*An acidic goo from the stomach dissolves the snail's flesh so the worm can re-swallow both stomach and prey.*
Simple	*An acidic goo from the stomach dissolves the snail's flesh, and the flatworm can then re-swallow both stomach and snail.*
BART_N	*[del] The goo dissolves the snail's flesh so the worm can re-swallow both stomach and prey.*
BART_4	*An acidic goo from the stomach dissolves the snail's flesh so the worm can re-swallow both stomach and prey.*
$\text{Ctrl}_{4,4}$	*An acidic goo from the stomach dissolves the snail's flesh.* ***[split] This allows*** *the worm to re-swallow both stomach and prey.*
(2) Complex	*Archer's parents were Catholic in Protestant England, which became illegal.*
Simple	*Archer's parents were Catholic. Being Catholic was against the law in England at the time.*
BART_N	*Archer's parents were Catholic. They did not want the government to interfere with their religion.*
BART_4	*Archer's parents were Catholic in Protestant England. [del]*
$\text{Ctrl}_{4,4}$	*Archer's parents were Catholic in Protestant England, which became illegal.*
(3) Complex	*Many people around the world are now furious at an American dentist who paid to go lion hunting in Zimbabwe.*
Simple	*Palmer paid to go lion hunting in Zimbabwe, which is in Africa.*
BART_N	*People around the world are* ***mad*** *at a dentist. He was paid to go lion hunting in Africa.*
BART_4	*Many people around the world are now furious at an American dentist who paid to go lion hunting in Zimbabwe.*
$\text{Ctrl}_{4,4}$	*Many people around the world are now furious at an American dentist who paid to go lion hunting in Zimbabwe.*
(4) Complex	*The Senegalese capital, on the far western edge of Africa, looks nothing like the sleepy port city that Amadou Sy remembers from his youth.*
Simple	*Amadou Sy remembers Dakar as a quiet city.*
BART_N	*The Senegalese capital* ***[del]*** *does not look like the sleepy port city that Amadou Sy remembers from his youth.*
BART_4	*The Senegalese capital* ***[del]*** *looks nothing like the sleepy port city that Amadou Sy remembers from his youth.*
$\text{Ctrl}_{4,4}$	*The Senegalese capital is on the far western edge of Africa. [split] It looks nothing like the sleepy port city that Amadou Sy remembers from his youth.*

Table 3.6: Example system outputs illustrating commonly seen patterns. Blue/**bold** marks positive changes while red/underlined marks negative changes or errors.

3.7 Conclusion

In this chapter we presented a new dataset for simplification operation classification and showed that it can be used to produce classifiers of much higher accuracy than what has been proposed in existing studies. We show that a controllable system using such a classifier to predict

sentence-level control tokens outperforms end-to-end baselines and existing systems on a range of datasets and receives extremely high ratings in fluency, adequacy and simplicity from human evaluators. However, this system does result in slightly lower simplicity ratings compared to reference texts and a Newsela baseline, suggesting that further improvements can be made to the system or dataset in order to achieve the best possible results across all criteria.

Chapter 4

Document-Level Planning for Text Simplification

Contents

Most existing work on text simplification is limited to sentence-level inputs, with attempts to iteratively apply these approaches to document-level simplification failing to coherently preserve the discourse structure of the document. In this chapter, we shift our direction towards document-level simplification and hypothesize that by providing a high-level view of the target document, a simplification plan might help to guide generation. Building upon the controllability techniques explored in Chapter 3, we view a plan as a sequence of labels, each describing one of four sentence-level simplification operations (copy, rephrase, split, or delete). We propose a planning model that labels each sentence in the input document while considering both its context (a window of surrounding sentences) and its internal structure (a token-level representation). Experiments on two simplification benchmarks (Newsela-auto and Wiki-auto) show that our model outperforms strong baselines both on the planning task and when used to guide document-level simplification models.

4.1 Introduction

Previous simplification research has primarily considered the simplification of isolated sentences, mostly being focused on training a statistical or a neural model on pairs of complex and simplified sentences assuming that such models will learn to perform simplification operations (e.g. sentence splitting, lexical simplification or syntactic rephrasing) implicitly from the inductive bias present in the training data (Zhang and Lapata, 2017; Nisioi et al., 2017; Jiang et al., 2020).

However, the imbalanced representation of simplification operations throughout popular datasets, and the overly-conservative models arising from their use, have led to attempts at controllable simplification to achieve more variation and diversity in output texts (Alva-Manchego et al., 2017; Maddela et al., 2021). As discussed in Chapter 3, these usually aim to either constrain attributes of the output (length, amount of paraphrasing, lexical and syntactic complexity) (Martin et al., 2020) or explicitly specify which simplification operation to perform (Alva-Manchego et al., 2017; Dong et al., 2019; Malmi et al., 2019; Scarton et al., 2020; Maddela et al., 2021).

To guide the simplification of full documents, we combine the power of data-driven neural generative models with the controllable simplification strategies proposed in Chapter 3. Our hypothesis is that document-level simplification can be facilitated by a plan specifying how each complex input sentence should be transformed to yield a simplified version of that document - should it be copied, deleted, split or rewritten?

Within this chapter we make the following contributions:

- We present a model for predicting document simplification plans which leverages both the context of sentences and their internal structure (the words they consists of).
- We create the data necessary to train this model by labelling complex sentences in simplification corpora with the simplification operation that relates it to the corresponding simplified sentence.
- We compare our planning model with several alternative neural architectures and briefly examine the impact of planning on document simplification.

Experiments on two simplification benchmarks (Newsela-auto and Wiki-auto) show that our model outperforms strong baselines both on the planning task and when used to guide document-level simplification models.[18]

4.2 Related Work

Document-Level Simplification. There is limited existing work on document-level text simplification. Early attempts largely applied sentence-level techniques iteratively over a document (Woodsend and Lapata, 2011b; Alva-Manchego et al., 2019b). However, this is generally viewed as insufficient for certain operations and maintaining the discourse coherence of the document (Siddharthan, 2003; Alva-Manchego et al., 2019b).

Several works address sub-problems of simplification that only include a limited set of operations, like paraphrasing and sentence re-ordering (Lin et al., 2021), insertion (Srikanth and Li, 2021) or deletion (Zhong et al., 2020; Zhang et al., 2022). Others fully address simplification but only extend inputs to the level of paragraphs without clearly differentiating the problem from the sentence-level (Laban et al., 2021; Devaraj et al., 2021).

[18]Pretrained models, code, and data are available at `https://github.com/liamcripwell/plan_simp`.

Recently, Sun et al. (2020) proposed a sentence-level model (SUC) that uses an encoding of surrounding sentences as contextual information to influence the simplification. They use two extra encoders to build a representation of the two preceding and two following sentences, which are attended over in their encoder-decoder generative model. However, when applied to the document-level task, their system was unable to outperform any baseline systems (Sun et al., 2021).

Operation Prediction. Revision-based simplification models learn to predict edit operations to apply at the token-level rather than generating the entire simplification from scratch (Alva-Manchego et al., 2017; Dong et al., 2019; Kumar et al., 2020; Omelianchuk et al., 2021; Dehghan et al., 2022). This has the benefit of providing more control and interpretability over generative approaches, often at the cost of the ability to perform major structural changes. It also allows some systems to leverage non-autoregressive generation strategies, resulting in faster inference times (Malmi et al., 2019; Omelianchuk et al., 2021).

Some works have attempted to predict rewrite operations at the sentence-level. Applying a binary classifier to predict whether simplification should be performed has been found to improve SARI results, reducing conservatism and spurious transformations (Scarton et al., 2020; Garbacea et al., 2021). Others have proposed multi-class systems to predict sentence-level operations that are then used to condition a generative model (Scarton and Specia, 2018; Scarton et al., 2020), as is the case of the system we propose in Chapter 3. These show some capacity for general improvement over end-to-end systems, while also dramatically improving the performance and frequency of specific operations (e.g. splitting, as explored in Chapter 3).

At the document-level, there has been little interest to date. However, there are recent works specifically looking at predicting sentence deletions (Zhong et al., 2020; Zhang et al., 2022). Both of these use features of the discourse structure from surrounding sentences to identify likely deletion candidates.

We bring all of these methods together by proposing a system that uses both sentence and document-level information to predict a multi-class, sentence-level operation plan over an entire document.

4.3 Problem Formulation

Let C denote an English language document. The aim of document-level simplification is to produce a text S that simplifies the input document C.[19] As a plan can provide a high-level view of a document, we hypothesize that a document-level simplification model that is based on a plan specifying a simplification operation for each input sentence should fare better than a simplification model that directly simplifies an entire document.

We therefore decompose simplification into a two-stage generation process:

$$p(S \mid C) = p(S \mid C, O)p(O \mid C)$$

where input document $C = c_1 \dots c_n$ is a sequence of complex sentences, $S = s_1 \dots s_k$ is a sequence of simplified sentences and $O = o_1 \dots o_n$ is a sequence of sentence-level simplification operations for C.

We consider three simplification operations proposed in previous work on sentence simplification (*copy*, *rephrase*, and *split*) to which we add *delete*, an operation that is needed to account for the fact that, contrary to sentence simplification, document-level simplification can require for a

[19] Note C and S here are used to refer to full documents, whereas they were used to refer to individual sentences in the previous chapters.

sentence present in the input document to be excluded from the resulting simplified document entirely.

Given the input document C, the first-stage model aims to predict the sequence of simplification operations O that should be applied to each individual sentence in that document. The second-stage model generates the output simplified document S conditioned on the input document C and its accompanying simplification plan O.

In this chapter, we focus on the planning stage, comparing different architectures and demonstrating the impact of planning on three possible document-level simplification models. We leave further exploration of alternative, more complex architectures for the simplification stage to Chapter 5.

4.4 Data

In this section we introduce the datasets used, explain how annotation is performed for each complex sentence and describe other preprocessing steps.

Dataset. For all experiments, we utilize Wiki-auto and Newsela-auto (Jiang et al., 2020), two datasets of English documents paired with their simplification. These datasets were derived from WikiLarge (Zhang and Lapata, 2017) and Newsela (Xu et al., 2015) by aligning the input document with the output simplification at both the sentence and the paragraph level.

WikiLarge gathers three simplification datasets which were automatically-collated from English Wikipedia and Wikipedia simple (Zhu et al., 2010; Woodsend and Lapata, 2011a; Kauchak, 2013).

Newsela consists of news articles, each manually rewritten at five different levels of simplification, corresponding to discrete reading levels (0-4) of increasingly simplicity. Aligned pairs are created by pairing every article version with each other version corresponding to a higher reading level. Because of this, there can be up to four aligned document pairs that contain the same document as either the input or the output.

The types of operations present in different reading level pairings differs significantly, with adjacent level transitions being extremely conservative (no instances of deletion throughout entire dataset). To mitigate any issues arising from this, all models we train with Newsela-auto receive a control-token at the start of the input which specifies the target reading level.

We do not use the D-Wikipedia dataset from Sun et al. (2021) as it does not contain sentence/paragraph alignments and is poorly formatted. In particular, all text is lower-cased and pretokenized in a way that makes it difficult to accurately parse sentences. There are also regular formatting issues at points where references exist in the source article.

Annotating Complex Sentences. Using the pairs (c_i, s_j) of complex and simplified sentences available in Wiki-auto and Newsela-auto, we heuristically assign a silver simplification operation label to each complex sentence c_i in these two datasets as follows:

- ***Delete:*** c_i is not aligned to any s_j.
- ***Copy:*** c_i is aligned to a single s_j with a Levenshtein similarity above 0.92.
- ***Rephrase:*** c_i is aligned to a single s_j with a Levenshtein similarity below 0.92.
- ***Split:*** c_i is aligned to multiple s_js.

Preprocessing. Wiki-auto contains many document pairs with wildly different sizes. We therefore clip all complex documents after the last aligned paragraph. Many simple articles resemble a summarization, rather than a simplification of the complex article (lots of deletion, often consisting of about one sentence from each paragraph in C). Because of this, we also remove documents where more than 50% of aligned sentences are labelled as *delete*. Finally, we remove all articles that exceed 1024 tokens (so that we can fit them into a baseline BART generative model).

For Newsela-auto, article pairs are much more even in length as they are manually created to be gradual, direct simplifications of each other. We do, however, perform the same length-based filtering to exclude documents that will not fit into a baseline generative model.

Train/Dev/Test Split. For both datasets we use a train/validation/test split of 92.5/2.5/5. This is applied at the document-level so that sentences from the same document will not exist across different sets. For Newsela, this means that all reading level versions of a single article will exist within the same set. Table 4.1 and Figure 4.1 give some statistics and a graphical description of the two datasets after pre-processing.

	Wiki-auto	Newsela-auto
# Doc Pairs	85,123	18,319
# Sent Pairs	461,852	707,776
Avg. $\|C\|$	155.51	868.98
Avg. $\|S\|$	97.72	674.94
Avg. $\|c_i\|$	28.64	22.49
Avg. $\|s_i\|$	21.57	15.84
Avg. n	5.43	38.64
Avg. k	4.53	42.60

Table 4.1: Statistics of each dataset after preprocessing, where n is # sentences in C and k is # sentences in S.

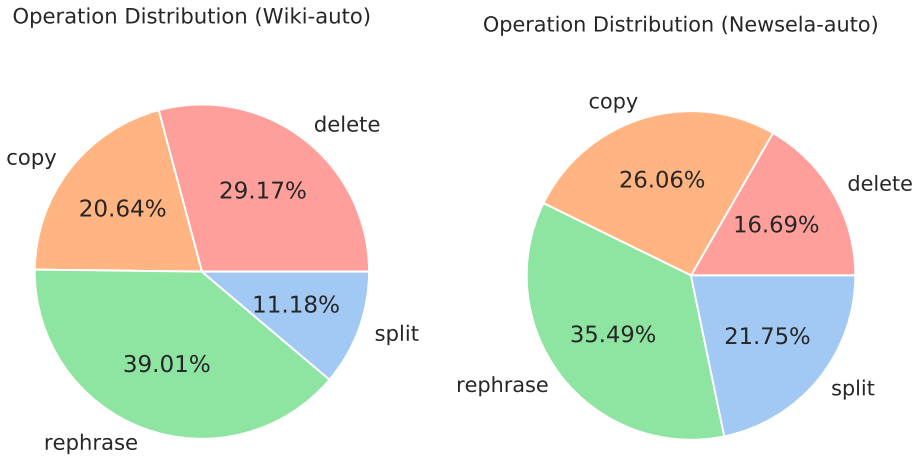


Figure 4.1: Operation class distributions for Wiki-auto (top) and Newsela-auto (bottom) datasets.

4.5 Planning

We present our primary model and four alternative models we explored for comparison. Each model was trained with a learning rate of $1e^{-5}$, a batch size of 32 and a dropout rate of 0.1. We ran experiments on a computing grid with 2× Nvidia A40 GPUs (45GB memory).

4.5.1 Model (Contextual Classifier)

Given some input document $C = c_1 \dots c_n$ consisting of n complex sentences c_i, the task of the planner is to predict a sequence $\hat{O} = \hat{o}_1 \dots \hat{o}_n$ of n simplification operations with $\hat{o}_i \in \{copy,\ rephrase,\ split,\ delete\}$.

One challenge with this is that the operations have different, sometimes conflicting contextual requirements. By construction, splitting (as well as rephrasing) is mostly context independent as it is mainly determined by the input sentence's internal structure: a sentence will be split only if it has the appropriate syntactic (e.g., The man who sleeps snores → The man sleeps. He snores.) or discourse (e.g., John went shopping after he left work → John left work. Afterwards he went shopping.) structure. For sentence splitting, context (the other sentences in the input document) has little, if any impact.

In contrast, deletion and to a lesser extent, copy and rephrase are mostly context dependent. Intuitively, a sentence can only be omitted in the simplified text in cases where it is either redundant with, or of minor semantic import relative to, other sentences in the document. That is, while for splitting, internal sentence structure is the key factor, for deletion, it is the semantics of the input sentence and how it relates to that of the other sentences which matters most.

We model these different requirements by using a token level encoder for the target document sentence c_i (the input sentence to be labelled with a simplification operation) and a sentence level representation of the context where each $c_p \in c_1 \dots c_{i-1}, c_{i+1} \dots c_n$ is represented by a sentence level embedding constructed using SBERT (Reimers and Gurevych, 2019). In this way both the internal structural information needed to capture splitting operations and the contextual information required by the other operations are provided. Specifically, we propose a model for planning that combines a classifier with cross-attention over the (dynamic or static) context and two types of positional embeddings. Figure 4.2 illustrates our model architecture.

Classifier with Cross-attention over the Context. We build upon a RoBERTa classifier architecture to enable conditioning upon the surrounding sentences in the document. We do this by inserting an additional cross-attention layer between the self-attention and the feed-forward layer of each transformer block, allowing the model to attend to a latent representation of the surrounding sentences, Z_i.

Context Representation. To obtain Z_i, we take a fixed window of radius r, extract the r sentences on either side of the target sentence to be simplified and concatenate the representation of each of these sentences. Each context sentence is encoded with the pretrained Sentence-BERT (SBERT) model[20] and combined with custom learned positional embeddings.[21]

To better simulate autoregressive inference, we consider a strategy where the left context consists of previously simplified sentences, rather than complex ones. We refer to this as *dynamic context*. At training time, we use the ground truth simplifications

$$Context_{i,r} = Concat(s_{j-r..j-1}, c_{i..i+r}) \tag{4.1}$$

[20] Specifically, all-mpnet-base-v2.

[21] At training time, we backpropogate to the positional embedding layers but keep the SBERT weights frozen.

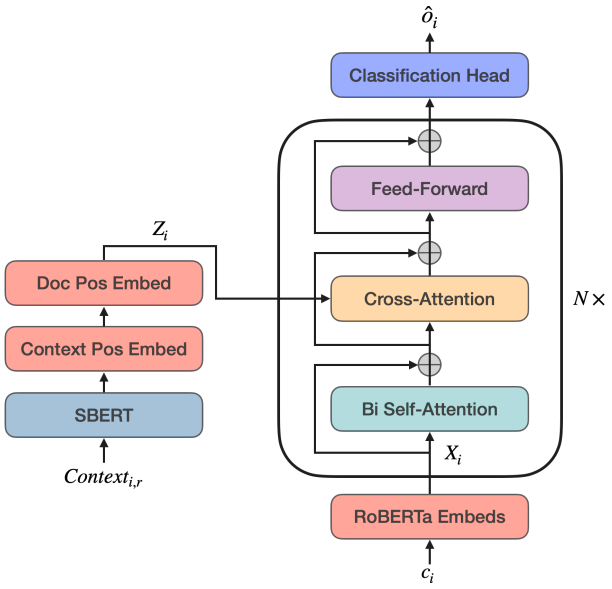


Figure 4.2: Contextual classifier model architecture.

where $j \in \{1, \ldots, |S|\}$ is the index of the first sentence aligned to c_i in the simple document S. During inference, the simplifications generated at preceding timesteps $\hat{s}_{j-r..j-1}$ are used.

Context Window Size To determine a good context window size we ran a series of experiments with varying values of the radius, r. We used 100,000 random examples from the Newsela-auto (non-adjacent reading levels) training set and trained a model with each of the configurations for 5 epochs. Results can bee seen in Figure 4.3.

The *deletion* operation is most affected by the inclusion of context, with performance rapidly rising as r grows to 13. The *rephrase* operation appears to slowly degrade in performance as r increases, while the other two operations show no obvious pattern. We also observe that $r = 9$ produces the highest macro F1.

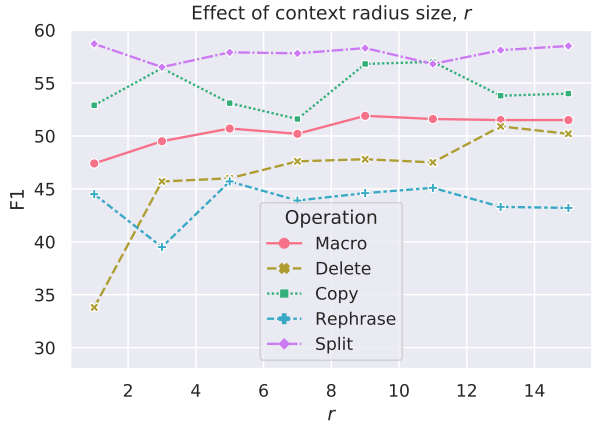


Figure 4.3: Effect of context window size on F1 scores.

Positional Embeddings. We use custom positional embeddings to encode both information about document, and relative context-window positions. These are each handled by a dedicated

embedding layer and added to the representations of the corresponding context sentence.

Document positional embedding indices are simply the document quintile (1-5) that a given sentence falls into. We use quintiles as this will ensure that all indices are encountered within the input document. The context positional embedding indices are the relative distance of a given sentence from the input sentence c_i, adjusted to be within $\mathbb{N}_0$: $ContextPosIdxs = \{p - i + r \mid p \in \{i - r, \ldots, i + r\}\}$.

Initialization. Given that the cross-attention layers must be trained from scratch, the start of training can see a lot of instability in the model, potentially making it more difficult to model context-independent features of the input sentence. To account for this, we initialize the RoBERTa layers with weights from a context-independent classifier. All layers in common with the standard RoBERTa architecture are initialized with the `RoBERTa-base` pretrained weights. All added positional embedding layers are also initialized with the pretrained weights from the `RoBERTa-base` positional embedding layer. All other layers are randomly initialized.

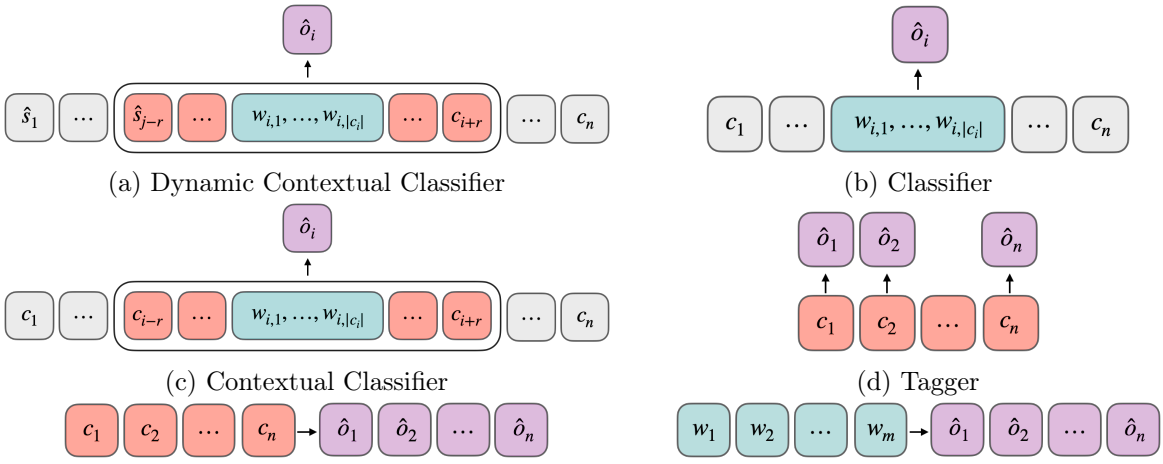


Figure 4.4: Visualization of the inputs/outputs of the various models, where $w_{i,t}$ is the tth token in c_i, n is the no. sentences in C and m is the no. tokens in C. Sentence-level representations are shown in red, token-level representations in teal, operation labels in pink, and unused parts of C in grey.

4.5.2 Alternative Models

We compare our model with four alternative models. The different inputs/outputs of the models are illustrated in Figure 4.4.

Classifier. We fine-tune pretrained `RoBERTa-base` (Liu et al., 2019), which has 12 hidden layers and a hidden size of 768, and add a pooled classification head which takes the final layer [CLS] representation as input.

Given an input sentence c_i, the model simply takes the tokenized sentence as input and outputs a prediction score for each operation class. The model is applied from left-to-right on the input document classifying each sentence in turn. Thus, in this approach, while the model has access to the tokens of the sentence to be classified, there is no notion of context which, intuitively, should be detrimental in particular for deletions and, to some extent, copying.

Tagger. We consider a model that frames the problem as a sequence tagging task over the full document, predicting the entirety of $\hat{O}$ at once. Each c_i is encoded using the same SBERT model as the contextual classifier, with the input document C therefore being represented as a sequence of sentence embeddings. In contrast to the classifier, the tagger makes predictions based both on the input sentence to be classified and on the context. However, because the input representation at each index is for an entire sentence, we lose some resolution with respect to token-level content. The approach is thus less adapted for splitting.

Tagger+Dec. We also consider an autoregressive variant of the tagger that better models the dependencies between predicted tags. Here, we include a 1-layer decoder and condition each prediction both on the input document and on the previously predicted operation tags for the earlier sentences. This approach is somewhat similar to Dong et al. (2019); Malmi et al. (2019), except that we abstract to the document-level and do not require explicit realization, as this will be handled downstream by the simplification model.

EncDec$_{\text{full}}$. Finally, we experiment with an encoder-decoder variant that conditions on a token-level representation of the entire input document, thereby combining a global view and a token-level representation of the input document. We use sentence separator tokens to delimit each sentence in the input document.

4.5.3 Evaluation Metrics

To evaluate the performance of the various planners we use F1-score, considering each individual prediction at the sentence-level. We report the F1 for each operation class as well as both the micro and macro averages. The micro F1 weights all examples equally, whereas the macro re-weights examples such that each class is represented equally in the final score. Given the slight class imbalances in the data, we regard macro F1 as our primary metric.

	Wiki-auto						**Newsela-auto**					
Model	C	R	S	D	Micro	**Macro**	C	R	S	D	Micro	**Macro**
EncDec$_{\text{full}}$	26.9	42.2	36.0	51.8	43.2	40.8	26.1	10.8	11.7	9.0	12.2	11.5
Tagger+Dec	29.3	54.5	30.0	51.8	47.7	41.4	72.2	73.9	75.9	79.7	75.0	75.4
Tagger	38.6	54.2	31.7	**58.5**	50.6	45.8	71.4	72.7	74.1	78.4	73.7	74.1
Classifier	42.1	52.9	42.6	49.0	48.4	46.7	77.0	75.6	80.0	78.5	77.4	77.8
Dyn. Context	**44.8**	**57.9**	42.4	54.8	**52.8**	**50.0**	79.3	77.3	82.8	81.4	79.7	80.2
+ docpos	43.7	55.4	**43.6**	56.7	52.3	49.9	**80.0**	**78.1**	**83.6**	**82.0**	**80.3**	**80.8**

Table 4.2: Planning Accuracy. **Dyn. Context** is the contextual classifier described in Section 4.5.1 with $r = 13$, dynamic context and weights initialized using the classifier weights (C: Copy, R: Rephrase, S: Split, D: Delete).

4.5.4 Results

Table 4.2 summarizes the results. Compared to the various baselines, our model consistently shows the best results on both datasets. However, the improvement over the context-free classifier is slightly less on Newsela-auto. We conjecture that the much larger dataset and additional guidance provided by the reading levels allows the classifier to achieve rather high accuracy without document-level context. We also note that the context-free classifier is markedly outperformed by other models with respect to *delete*, which confirms the intuition that context modeling particularly matters for this operation.

Of the four baselines, $\text{EncDec}_{\text{full}}$ performs worst presumably because the very long input (the whole context is modelled at the token level) challenges the attention mechanism which tends to become blurry as the length of the input increases. This is particularly apparent on the longer Newsela documents. The tagger models, which both use sentence-level encodings of the complex document, perform slightly worse than the classifier. This highlights the importance of having a token-level modeling of the input sentences.

We observe a strong difference in terms of absolute scores between the two datasets. In particular, the Wiki-auto results seem less stable; likely a result of it being an inferior simplification corpus (discussed in Section 4.4).

Figure 4.5 shows example snippets of planner model outputs. We have selected representative extracts that highlight the strengths and weaknesses of the main models. We do not include outputs from Tagger as they are virtually identical to Tagger+Dec in most cases and therefore do not provide further insight.

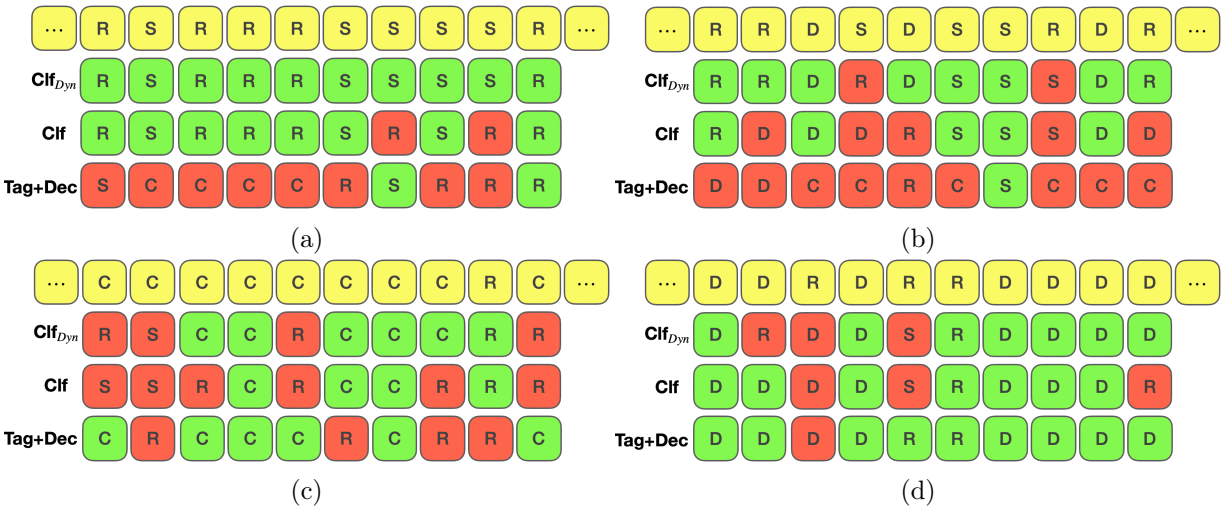


Figure 4.5: Example planning results for various models. Subfigures show representative snippets from Newsela-auto test-set documents. The silver labels are shown above in yellow, and system outputs are shown on the rows below with correct predictions in green and incorrect predictions in red. Clf_{Dyn} is our best performing model, the contextual classifier with dynamic context. Figure 4.5a shows a case where there are lots of context-agnostic operations (rephrase, split) resulting in poor performance from Tagger+Dec. Figure 4.5b shows a varied snippet where Clf_{Dyn} appears to be the best at identifying both rephrase and split, as well as delete. Figures 4.5c and 4.5d show that Tagger+Dec is capable of performing well in situations demanding a lot of context-dependent operations (copy, delete).

Next, we examine the impact of our modeling choices using ablation (Table 4.3) and focusing on the higher-quality, Newsela-auto dataset. Our best model is one with dynamic left-context, a context radius of 13, document position embeddings and weight initialization. We see (Sub-table **a**) that each of these components help improve performance (document position appears less important with a larger context window). Sub-tables **b-d** show that using a dynamic rather than a static context increases results by up to +6.7 Macro F1, while increasing the context radius from 9 to 13 sentences mostly improves performance when dynamic context is used. Using document positional embeddings also generally improves results (Sub-table **d**).

Model	Copy	Rephrase	Split	Delete	Micro	**Macro**
(a) Ablation on Best Model						
Dyn, $r = 13$, +init, +docpos	80.0	78.1	83.6	82.0	80.3	80.8
-docpos	79.3	77.3	82.8	81.4	79.7	80.2
-init	74.9	72.1	77.8	75.2	74.6	75.0
-init, -docpos	75.6	72.0	77.7	77.1	75.1	75.6
(b) Dynamic vs. Static Context						
Stat, $r = 9$	71.3	69.5	75.4	73.3	72.0	72.4
Stat, $r = 13$	72.2	65.3	69.9	68.3	68.5	68.9
Dyn, $r = 9$	73.1	70.1	75.5	75.9	73.1	73.6
Dyn, $r = 13$	75.6	72.0	77.7	77.1	75.1	75.6
(c) With vs without initialization						
Dyn, $r = 9$	73.1	70.1	75.5	75.9	73.1	73.6
Dyn, $r = 9$ +init	79.3	78.0	82.7	79.8	79.7	80.0
Dyn, $r = 13$	75.6	72.0	77.7	77.1	75.1	75.6
Dyn, $r = 13$ +init	79.3	77.3	82.8	81.4	79.7	80.2
(d) Window Size						
Stat, $r = 9$	71.3	69.5	75.4	73.3	72.0	72.4
Stat, $r = 13$	72.2	65.3	69.9	68.3	68.5	68.9
Dyn, $r = 9$	73.1	70.1	75.5	75.9	73.1	73.6
Dyn, $r = 13$	75.6	72.0	77.7	77.1	75.1	75.6
Dyn, $r = 9$ +docpos	73.8	72.9	77.2	75.8	74.6	74.9
Dyn, $r = 13$ +docpos	74.9	72.1	77.8	75.2	74.6	75.0
Dyn, $r = 9$ +init +docpos	79.4	78.0	83.1	82.0	80.1	80.6
Dyn, $r = 13$ +init +docpos	80.0	78.1	83.6	82.0	80.3	80.8

Table 4.3: Ablations on Newsela-auto TestSet.

4.6 Simplification

To assess whether document plans can help improve simplification model performance, we experiment with two simple document-level simplification models and compare their performance with and without guidance from a planning step.

4.6.1 Simplification Models

All models use the BART model (Lewis et al., 2020) fine-tuned on aligned text pairs.[22]

We consider two baselines for document-level simplification: (i) **Doc-BART**, which is fine-tuned on full document pairs; and (ii) **Sent-BART** which is fine-tuned on sentence pairs and iteratively applied to each input sentence at test time.

We compare these to various plan-guided (**PG**) systems wherein one of our planners predicts an $\hat{o}_i$ for each c_i and is given as a control-token to a sentence-level BART simplification model. In the case of the dynamic planner, $\hat{o}_i$ is predicted based on the sequence of previously simplified sentences $\hat{s}_{i-r} \dots \hat{s}_{i-1}$.

For all generative models, we used a learning rate of $3e^{-5}$, a batch size of 16, and performed dropout with a rate of 0.1 and early stopping. The network has 6 layers in each of the encoder and decoder, with a hidden size of 768. All models were trained on a computing grid using 2× Nvidia A40 GPUs (45GB memory) in under 24 hours.

4.6.2 Evaluation

To measure meaning preservation and fluency, we use BARTScore (Yuan et al., 2021), a state-of-the-art summarization metric that has proven effective on many other text generation tasks. We also compute SMART (Amplayo et al., 2022), a new metric that considers sentences as the primary unit of comparison. It was shown to be highly effective for document summarization and does not use any neural model, making it very fast to compute (we use the SMARTL+CHRF version). We cannot use other model-based metrics, such as BERTScore or QuestEval, as these do not support texts longer than 512 tokens.

To assess simplicity, we use the Flesch-Kincaid grade level (FKGL), a document-level metric used to measure text readability, which has been found to have the highest correlation with simplicity measures of human-written simplifications (Scialom et al., 2021b). We also report the popular SARI (Xu et al., 2016). The EASSE python library (Alva-Manchego et al., 2019a) is used for calculation of FKGL and SARI. We include results for other popular metrics in Appendix B.1.

At test time we generate sequences using beam search with a beam size of 5 and a maximum length of 1024 tokens. We enforce a minimum length for Doc-BART, which is tuned on the validation set. We do not conduct a human evaluation as we intend the focus of this chapter to be on the planning component and include simplification results only to confirm its potential efficacy in guiding downstream models. A more in-depth investigation of the interaction between planning and document-level simplification will be considered in Chapter 5.

4.6.3 Results

Results can be seen in Table 5.2. PG_{Dyn} achieves the highest overall results of all systems. Using the silver operation labels ($\text{PG}_{\text{Oracle}}$) leads to a substantial further increase in performance across every metric, highlighting the impact of planning and pointing to the possibility of further improvements to be made.

[22] We use the pretrained `facebook/bart-base` model from https://huggingface.co/facebook/bart-base.

Using either PG_{Dyn} or PG_{Clf} yields generally better results than Sent-BART. Both systems achieve better FKGL and SARI, suggesting greater output simplicity. Sent-BART also achieves much higher source-oriented BARTScore (faithfulness) than even the references, suggesting more conservativity in its transformations.

PG_{Clf} achieves slightly higher recall BARTScore than PG_{Dyn}, while also generating the longest outputs, both in terms of tokens and sentences. This suggests it is less effective at identifying sentences for deletion, confirming our hypothesis that context is key for deletion. We can see here that the rank order of SMART matches that of BARTScore, suggesting it may be similarly suitable for evaluating simplification.

Both PG_{Tag} and $PG_{Tag+Dec}$ perform quite badly relative to the other PG systems and Sent-BART. However, Doc-BART is by far the worst performing system, presumably a result of it failing to properly handle the long document lengths.

System	**BARTScore ↑**				**SMART ↑**			**FKGL ↓**	**SARI ↑**	**Length**	
	Faith. $(s \to h)$	P $(r \to h)$	R $(h \to r)$	F1	P	R	F1			Tokens	Sents
Input	-0.93	-2.47	-1.99	-2.23	63.2	62.7	62.8	8.44	20.52	866.9	38.6
Reference	-1.99	-0.93	-0.93	-0.93	100	100	100	4.93	99.99	671.5	42.6
Doc-BART	-2.48	-2.68	-2.76	-2.72	61.9	43.9	50.6	10.01	47.07	600.8	20.7
Sent-BART	**-1.86**	-1.63	-1.56	-1.60	78.9	80.1	79.3	5.03	73.02	666.4	42.6
PG_{Tag}	-1.95	-2.22	-2.18	-2.20	62.0	62.6	61.6	5.07	56.13	657.4	41.8
$PG_{Tag+Dec}$	-1.94	-2.22	-2.18	-2.20	62.2	62.5	61.6	5.09	56.06	654.2	41.4
PG_{Clf}	-1.91	-1.68	**-1.53**	-1.60	77.8	**81.2**	79.3	**4.95**	73.83	688.8	44.5
PG_{Dyn}	-1.91	**-1.60**	-1.54	**-1.57**	**80.2**	81.0	**80.5**	4.98	**75.00**	667.2	42.6
PG_{Oracle}	-1.93	**-1.39**	**-1.40**	**-1.40**	**85.5**	**85.0**	**85.3**	**4.91**	**80.74**	655.6	42.1

Table 4.4: Results of document simplification systems on Newsela-auto. For BARTScore, s is the source, h is the hypothesis, and r is the reference.

4.6.4 Example Simplification Outputs

Table 4.5 shows system output examples for the simplification models. We only show texts from Wiki-auto as they are easier to showcase due to their shorter length, as well as not being subject to the licensing restrictions of Newsela.

4.7 Conclusion

In this chapter we presented an approach to document simplification that decomposes the task into a two-stage process of planning and generation. We proposed a planning system that is able to take document context and structure into account to produce a coherent high-level simplification plan. By using this plan to guide a sentence-level simplification model, we are able to outperform end-to-end systems in terms of both meaning preservation and simplicity. In the next chapter, we explore dedicated simplification models that can leverage a document-level plan while also considering contextual information directly during generation.

System	Output
Complex	Silvano "Nano" Campeggi (January 23, 1923 – August 29, 2018) was an Italian artist who designed and produced the artwork for the posters of many classic Hollywood films. His iconic images are associated with the golden era of Hollywood and Campeggi is now generally regarded as the most important graphic artist and poster designer in the history of American cinema. In the following decades, Campeggi designed and produced the poster and advertising graphics for over 3000 films, working not only under contract with the MGM studios, but also with Warner Brothers, Paramount, Universal, Columbia Pictures, United Artists, RKO, Twentieth-Century Fox and several other movie studios. Sixty-four of the films he illustrated won Oscars, including "Casablanca", "Ben-Hur", "Singin' in the Rain", "An American in Paris", "West Side Story", "Exodus", "Breakfast at Tiffany's", and "Gigi". Campeggi died on 29 August 2018, at the age of 95.
Simple	Silvano "Nano" Campeggi (January 23, 1923 – August 29, 2018) was an Italian artist. He designed and produced the artwork for the posters of many classic Hollywood movies. Campeggi was known for his poster designs for "Casablanca", "Singin' in the Rain", and "Breakfast at Tiffany's". Campeggi died on August 29, 2018 in Florence from respiratory failure at the age of 95.
Doc-BART	Silvano "Nano" Campeggi (January 23, 1923 – August 29, 2018) was an Italian artist. He designed and produced the artwork for the posters of many classic Hollywood movies, including "Casablanca", "Ben-Hur", "Singin' in the Rain", "An American in Paris", "West Side Story", "Exodus", "Breakfast at Tiffany's", "Gigi", and "".
Sent-BART	Silvano "Nano" Campeggi (23 January 1923 – 29 August 2018) was an Italian artist. He designed and produced the artwork for the posters of many classic Hollywood movies. Campeggi is generally regarded as the most important graphic artist and poster designer in the history of American cinema. Campeggi designed and produced the poster and advertising graphics for over 3000 movies, working not only under contract with the MGM studios, but also with Warner Brothers, Paramount, Universal, Columbia Pictures, United Artists, RKO, Twentieth Century Fox and several other movie studios. Sixty-four of the movies he illustrated won Oscars, including "Casablanca", "Ben-Hur", "Singin' in the Rain", "An American in Paris", "West Side Story", "Exodus", "Breakfast at Tiffany's", and "Gigi". Campeggi died on 29 August 2018, at the age of 95.
PG_{Dyn}	Silvano "Nano" Campeggi (January 23, 1923 – August 29, 2018) was an Italian artist. He designed and produced the artwork for the posters of many classic Hollywood movies. Sixty-four of the movies he illustrated won Oscars, including "Casablanca", "Ben-Hur", "Singin' in the Rain", "An American in Paris", "West Side Story", "Exodus", "Breakfast at Tiffany's", and "Gigi". Campeggi died on 29 August 2018 at the age of 95.

Table 4.5: Simplification outputs for a specific document pair example. Although Newsela-auto is the focus of our simplification experiments, we can only include example documents from Wiki-auto due to licensing constraints.

Chapter 5

Context-Aware Document Simplification

Contents

In the previous chapter, we showed that strategies from controllable simplification can be leveraged to achieve state-of-the-art results on document simplification by first generating a document-level plan (a sequence of sentence-level simplification operations) and using it to guide sentence-level simplification downstream. However, this is still limited in that the simplification model has no direct access to the local inter-sentence document context (context is only used by the operation classifier used for planning), likely having a negative impact on surface realization. We explore various systems that use document context within the simplification process itself, either by iterating over larger text units or by extending the system architecture to attend over a high-level representation of document context. In doing so, we achieve state-of-the-art performance on the document simplification task, even when not relying on plan-guidance. Further, we investigate the performance and efficiency trade-offs of system variants and make suggestions for when each should be preferred.

5.1 Introduction

In Chapter 4 we discussed the limitations of past attempts at document-level simplification and showed that strategies from controllable simplification can be leveraged to achieve state-of-the-art results. Specifically, by using a planning model capable of considering the sentences surrounding a complex sentence, a sentence-level simplification model can be guided such that the structure of the resulting document remains more coherent. Despite this success, the sentence simplification model still has no direct access to document context which we believe limits the extent to which it can accurately produce simplified sentences that are consistent with the larger document.

As such, in this chapter we propose various systems that allow access to some representation of surrounding content within the simplification module itself, while still allowing for the possibility of plan-guidance. We show that in doing so, we are able to achieve state-of-the-art document simplification performance on the Newsela dataset, even without relying on a generated plan. Further, we investigate the performance and efficiency trade-offs of various system variants.[23]

The key contributions of this chapter are (i) a detailed investigation of how document context, input text and simplification plans impact document-level simplification, and (ii) several state of the art models for document simplification. We show in particular that document level simplification is improved by combining a representation of the local context surrounding complex sentences with a simplification plan indicating how complex sentences should be simplified (whether they should be deleted, rephrased, split or copied).

5.2 Related Work

Context in Controlled Text Generation The use of external context within controlled text generation pipelines has seen recent success in areas outside of simplification. Li et al. (2021) control review generation by using document and sentence-level plans in the form of knowledge graph subgraphs. Smith et al. (2020) control the style of generated dialog responses by conditioning on a desired style token appended to other contextual utterances. Hazarika et al. (2022) modulate the amount of attention paid to different parts of a dialog context and show that using contextual encodings of question phrases can guide a model to more often generate responses in the form of questions. Slobodkin et al. (2022) consider summarization where salient spans are first identified before being used to control the generation, while Narayan et al. (2023) first generate a summarization plan consisting of question-answer pairs.

Simplification Planning Certain controllable sentence simplification works have approached simplification as a planning problem whereby an operation plan is first generated before being realized downstream to form the simplified text. The first of these are revision-based models that predict a sequence of token-level operations (delete, substitute, etc.), allowing for more control and interpretability (Alva-Manchego et al., 2017; Dong et al., 2019; Kumar et al., 2020; Omelianchuk et al., 2021; Dehghan et al., 2022). Others have taken a sentence-level approach by predicting a high-level operation (sentence split, rephrase, etc.) and using this to condition more typical neural systems (Scarton and Specia, 2018; Scarton et al., 2020; Garbacea et al., 2021).

We took the latter style of approach in Chapter 3 and later leveraged it for document simplification in Chapter 4, where it obtained state-of-the-art performance. There, a sequence of sentence-level operations is predicted for an entire document and then used to iteratively condition a sentence-level simplification model. The system considers both local (token representation

[23] Pretrained models, code, and data are available at https://github.com/liamcripwell/plan_simp.

of the sentence) and global document context (sequence of sentence-level encodings) when predicting an operation for a given sentence, but does not consider context within the generation component.

5.3 Problem Formulation

The goal of text simplification is to generate a text S that simplifies an input text C. In the document-level case, $C = c_1 \dots c_n$ is a sequence of complex sentences and $S = s_1 \dots s_m$ is a sequence of simple sentences. In Chapter 4, we further decomposed this task into a two-stage process wherein a generated plan conditions the simplification:

$$P(S \mid C) = P(S \mid C, O)P(O \mid C)$$

where $O = o_1 \dots o_n$ is a simplification plan, i.e. a sequence of sentence-level simplification operations for C (*copy*, *rephrase*, *split*, or *delete*). The motivation here is that the plan provides a high-level description of how to transform C into S, which can in turn be used to guide the iterative generation of the simplified document across sentences.

Although we found that the use of such plans can lead to improved results, little attention has been given to how the generation stage itself can be modified to improve document-level simplification. In this chapter, we investigate whether further changes can be made to simplification models in order to make better use of high-level plans, or alternatively, whether it is possible to forego the planning stage entirely by incorporating high-level document context into the generative model directly.

Terminology and Notations. We use the following terminology and notational conventions throughout this chapter:

- $C = c_1 \dots c_n$ is a complex document of n sentences;
- p_i is the ith paragraph from the complex document C;
- $S = s_1 \dots s_m$ is a ground-truth simplified version of C, containing m sentences;
- $\hat{S} = \hat{s}_1 \dots \hat{s}_{m'}$ is a predicted simplification of C, generated by a simplification model;
- o is a simplification operation with value *copy*, *rephrase*, *split*, or *delete*;
- $\hat{O} = \hat{o}_1 \dots \hat{o}_n$ is a predicted simplification plan stipulating specific sentence-level operations that should be applied to each $c_i \in C$ so as to arrive at some $\hat{S}$;
- Z_i is a high-level representation of the document context for c_i. It is a sequence of vector encodings for a fixed window of sentences surrounding c_i within C.

5.4 Data

For all experiments, we use Newsela-auto (Jiang et al., 2020) which is currently the highest-quality document-level simplification dataset available. It consists of 1,130 English news articles from the original Newsela (Xu et al., 2015) dataset which are each manually rewritten at five different levels of simplification, corresponding to discrete reading levels (0-4) of increasing simplicity. It also includes both sentence and paragraph alignments for each document pair. Like previous works, for all our models we prepend a control-token to the input specifying the target document reading level.

We use the same filtered version of Newsela-auto used in Chapter 4, along with the same train/validation/test splits to allow for model comparison. This also includes plan labels, consisting of an operation (*copy*, *rephrase*, *split*, or *delete*) assigned to each sentence pair.

5.5 Models

We distinguish three model categories: (i) models whose sole input is text and which simplify a document either by iterating over its sentences/paragraphs or by handling the entire document as a single input; (ii) models that take both a complex sentence and some representation of its document context as input and simplify a document by iterating over its sentences (in a similar fashion to the contextual planner from Chapter 4); and (iii) models that are guided by a plan via control-tokens denoting sentence-level simplification operations prepended to the input sequence. These are illustrated in Table 5.1 and presented in more detail in the following subsections.

For all simplification models, we used a learning rate of $2e^{-5}$, a batch size of 16, and a 0.1 dropout rate. All models were trained on a computing grid using 2× Nvidia A40 GPUs (45GB memory) until convergence or a maximum of 48 hours. For ConBART and planning pipelines we use the same settings as in Chapter 4 for construction of the high-level document context. Specifically, this includes a fixed context window radius of size 13 and use of a dynamic context mechanism.

System	**Input**						
	Text			Context	Plan		
	Document	*Paragraph*	*Sentence*		*Document*	*Paragraph*	*Sentence*
BART	C	p_i	c_i	-	-	-	-
LED	C	p_i	-	-	-	-	-
ConBART	-	-	c_i	Z_i	-		
PG_{Dyn}	-	-	c_i	-	-	-	$\hat{o}_i$
$\hat{O} \rightarrow$ BART	C	p_i	c_i	-	$\hat{O}$	$\hat{o}_{j..j+\lvert p_i \rvert}$	$\hat{o}_i$
$\hat{O} \rightarrow$ LED	C	p_i	-	-	$\hat{O}$	$\hat{o}_{j..j+\lvert p_i \rvert}$	-
$\hat{O} \rightarrow$ ConBART	-	-	c_i	Z_i	-	-	$\hat{o}_i$

Table 5.1: **Different system types** and the specific forms of text, context, and plan inputs they consume. C is a complex document, c_i is the ith sentence of C, and p_i is the ith paragraph of C. $\hat{O}$ is a predicted document simplification plan, $\hat{o}_i$ is the individual operation predicted for the ith sentence, and $\hat{o}_{j..j+|p_i|}$ is the plan extract for a specific paragraph p_i, where j is the index of the first sentence in p_i.

5.5.1 Text-Only Models

The most basic group of models we test are those that simply take a text sequence as input. We use baseline models trained to take entire documents or individual sentences. We also experiment with using paragraph inputs, the results of which we believe should scale better to the document-level than isolated sentences. Because paragraphs contain a wider token-level representation of local context, this might provide enough information to maintain coherency in the discourse structure of the final document.

BART. We fine-tune BART (Lewis et al., 2020) to perform simplification at the document (**BART$_{\textbf{doc}}$**), sentence (**BART$_{\textbf{sent}}$**), and paragraph (**BART$_{\textbf{para}}$**) levels.[24] Both BART$_{\text{sent}}$ and BART$_{\text{para}}$ are applied iteratively over a document and outputs are concatenated to form the final simplification result.

Longformer. Encoder-decoder models like BART often produce worse outputs and become much slower the longer the input documents are. Longformer (Beltagy et al., 2020) is one proposal that aims to overcome these limitations by using a modified self-attention mechanism that scales linearly with sequence length. We fine-tune a Longformer encoder-decoder to perform the simplification on documents (**LED$_{\textbf{doc}}$**) and paragraphs (**LED$_{\textbf{para}}$**).[25]

5.5.2 Context-Aware Model (ConBART)

We propose a context-aware modification of the BART architecture (**ConBART**) that is able to condition its generation on both an input sentence c_i and a high-level representation of its document context Z_i (a sequence of vectors representing surrounding sentences in the document). This is done via extra cross-attention layers in each decoder attention block that specifically focus on Z_i. The ConBART architecture is illustrated in Figure 5.1.

We produce Z_i by employing the same context representation strategy used for planning in Chapter 4. Specifically, the document context is obtained by taking a fixed window of sentences surrounding the target c_i, encoding them with SBERT, and applying custom positional embeddings to represent location within the document.

By generating the plan autoregressively, it is also possible to use previously simplified sentences within the left context of the current complex sentence, a method we refer to as *dynamic context*. In this case, the window of sentences represented within Z_i is defined:

$$Context_{i,r} = Concat(\hat{s}_{i-r..i-1}, c_{i..i+r}) \tag{5.1}$$

where r is the context window radius and $\hat{s}_i$ is the simplification output for the ith sentence c_i. We use the same optimal setting of $r = 13$ as we did in Chapter 4.

The intuition behind the ConBART architecture is that the contextual information should allow for the simplification model to implicitly learn useful features of the discourse structure of the document in a similar way to the planner in Chapter 4.

5.5.3 Plan-Guided Systems

Existing System. We compare with the state-of-the-art system proposed in Chapter 4, **PG$_{\textbf{Dyn}}$**, which consists of a standard sentence-level BART model that is guided by a planner, which predicts the simplification operation to be applied each input sentence given a left and right context window of sentence representations, Z_i. The planner uses dynamic document context, allowing it to auto-regressively update the left context part of Z_i during planning as each sentence is simplified (see Equation 5.1).

Pipelines. We construct pipeline systems that consist of each of our proposed models, guided by a document plan generated by the planner from Chapter 4 (the same as is used by PG$_{\text{Dyn}}$). For this, we use modified versions of each simplification model that are trained to take an operation control-token at the beginning of each text input. We refer to each of these pipeline systems as

[24] All models are initialized with the pretrained `facebook/bart-base` model from https://huggingface.co/facebook/bart-base.

[25] All models are initialized with the pretrained `allenai/led-base-16384` model from https://huggingface.co/allenai/led-base-16384.

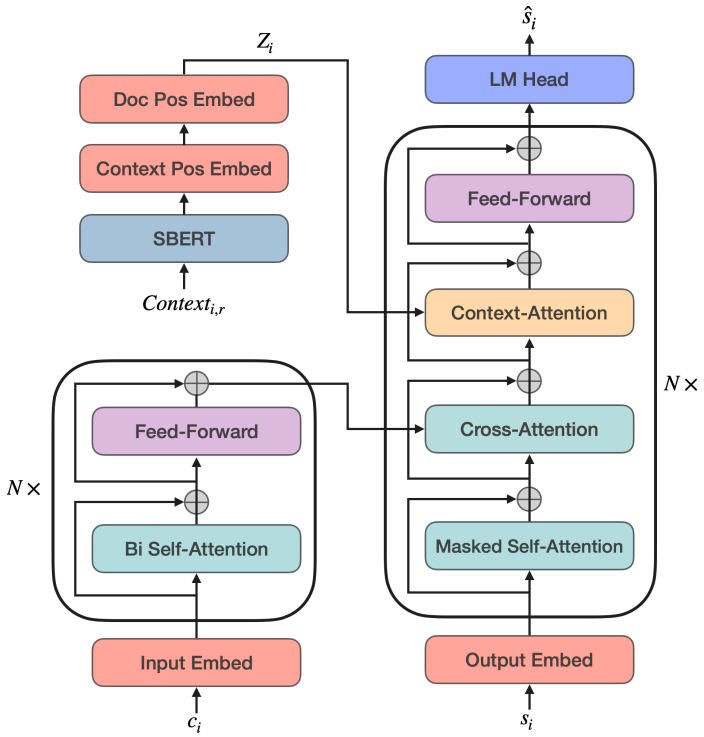


Figure 5.1: **ConBART model architecture.** The added context attention layer is shown in yellow, which allows for cross-attention over high-level document content, Z_i.

$\hat{O} \rightarrow h$, where h is the simplification model. We also report results where the ground-truth/oracle plans are used to condition models ($O \rightarrow h$).

Note that because the planner updates its document context autoregressively at the sentence-level, this does not interface perfectly with paragraph-level simplification models. As such, for pipelines using a paragraph-level simplification model ($\hat{O} \rightarrow \text{BART}_{\text{para}}$, $\hat{O} \rightarrow \text{LED}_{\text{para}}$), we only update the planner's context after each paragraph has been processed. Thus, for those paragraph level models, the left context of a complex sentence c_i is only simplified up to the first sentence of the paragraph containing c_i, i.e.

$$Context_{i,r} = Concat(\hat{s}_{i-r..j-1}, c_{j..i+r}) \tag{5.2}$$

where j is the index of the first sentence within the same paragraph as c_i, assuming $j > i - r$.

We also experimented with multi-task systems that are trained to perform both planning and simplification within a single model, therefore not requiring a pipeline setup. However, this ultimately proved unsuccessful (further details in Appendix C.1).

5.6 Evaluation

5.6.1 Automatic Evaluation

For automatic evaluation, we use BARTScore (Yuan et al., 2021) and SMART (Amplayo et al., 2022) as analogs for both adequacy and fluency. Both are reference-based metrics that have previously been used for document simplification as well as other text generation tasks. For assessing simplicity, we use both the Flesch-Kincaid grade level (FKGL) and SARI (Xu et al.,

2016). FKGL is a document-level metric of text readability that has the highest correlation with human judgements (Scialom et al., 2021b), while SARI is a simplification metric that has become a staple in the sentence-level simplification literature. We use EASSE (Alva-Manchego et al., 2019a) to calculate both of these. At test time we generate sequences using beam search with a beam size of 5 and a maximum length of 1024 tokens.

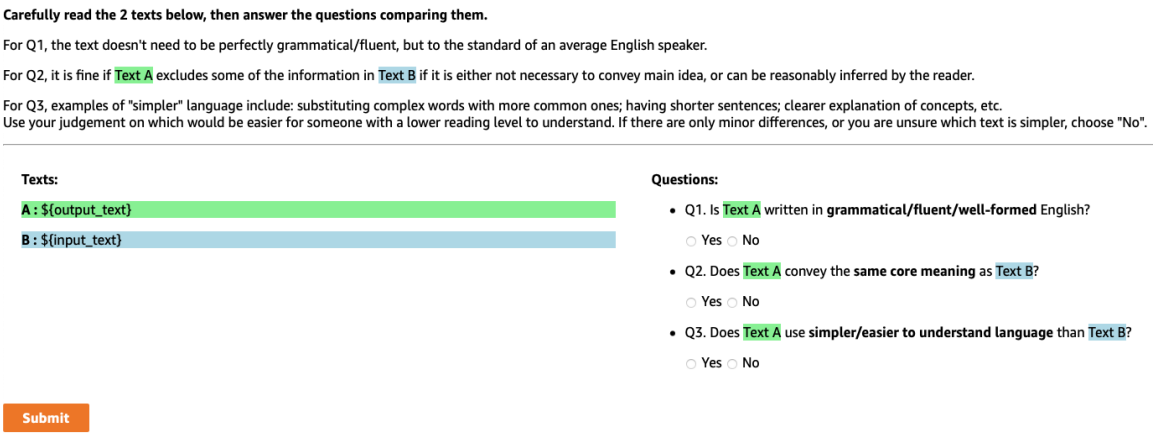


Figure 5.2: Submission form used in human evaluation.

5.6.2 Human Evaluation

Historically, automatic evaluation of long-form text generation has been very difficult to perform (Howcroft et al., 2020; Thomson and Reiter, 2020). As such, we conduct a human-evaluation of proposed systems to more accurately gauge performance.

As full documents are very long and difficult to compare, we conduct evaluations at the paragraph-level. For each comparison, a complex paragraph is shown next to an extract from a generated simplification corresponding to that paragraph. The Newsela-auto paragraph alignments were used to identify valid references for each test paragraph. In order to align correct extracts from generated system outputs we took different steps depending on the system. For paragraph-level models (those using LED_{para}), we simply use the full simplification output for each source paragraph. For sentence-level models (ConBART, PG_{Dyn}), we first used the alignments to identify which paragraph the source sentence belongs to, then concatenated their simplification results.

Using the test set, we randomly sample 33 complex paragraphs from each non-adjacent reading-level transition pairing, for a total of 198 paragraphs. We take the references and outputs from 4 high performing systems (PG_{Dyn}, LED_{para}, $\hat{O} \rightarrow \text{LED}_{\text{para}}$, $\hat{O} \rightarrow$ ConBART) for each (990 outputs in total) and have an annotator judge whether the generated text (i) is fluent (**fluency**); (ii) preserves the core meaning of the input (**adequacy**); and (iii) is simpler to read/understand (**simplicity**).

Human judgements were crowdsourced on the MTurk platform. Because we use a large pool of annotators, we impose a binary answering scheme (yes/no) in order to avoid the inter-annotator subjectivity that is inherent when using a Likert scale. The proportion of positive results is used as the final score for a given system. We sourced workers from English speaking countries (AU, CA, GB, IE, NZ, US) and paid them \$0.2 USD for each individual evaluation. We ran an initial test ourselves and timed how many evaluations could be completed within an hour. According to this, subjects should earn approximately \$18 USD per hour (which is above the minimum wage

in all of these countries). The form and instructions presented to human evaluators is shown in Figure 5.2.

System	BARTScore ↑			SMART ↑			FKGL ↓	SARI ↑	Length	
	P ($r \to h$)	R ($h \to r$)	F1	P	R	F1			Tok.	Sent.
Input	-2.47	-1.99	-2.23	63.2	62.7	62.8	8.44	20.5	866.9	38.6
Reference	-0.93	-0.93	-0.93	100	100	100	4.93	99.9	671.5	42.6
BART_{doc}	-2.68	-2.76	-2.72	61.9	43.9	50.6	10.01	47.1	600.8	20.7
$\text{BART}_{\text{sent}}$	-1.63	-1.56	-1.60	78.9	80.1	79.3	5.03	73.0	666.4	42.6
$\text{BART}_{\text{para}}$	-1.85	-1.49*	-1.67	77.2	82.8*	79.6	5.28	73.7	752.8	45.6*
LED_{doc}	-1.68	-1.73	-1.70	75.3	74.9	74.8	4.87	68.7	643.7	41.5
LED_{para}	-1.61	-1.40*	-1.50*	81.1	**85.5***	83.0*	5.15	76.9*	712.9	44.9*
ConBART	-1.59	-1.50	-1.54*	81.2	82.5*	81.7	5.01	75.8	669.8	42.8
PG_{Dyn}	-1.60	-1.54	-1.57	80.2	81.0	80.5	4.98	75.0	667.2	42.6
$\hat{O} \to$ ConBART	-1.52*	-1.45*	-1.48*	82.8*	84.0*	83.2*	4.96	78.3*	671.6	43.0
$\hat{O} \to \text{BART}_{\text{para}}$	-1.75	-1.47*	-1.61	79.4	81.9	80.4	5.11	74.9	715.3	42.7
$\hat{O} \to \text{LED}_{\text{para}}$	**-1.50***	**-1.42***	**-1.46***	**83.7***	84.9*	**84.1***	5.09	**78.5***	683.1	42.8
$\text{PG}_{\text{Oracle}}$	-1.39*	-1.40*	-1.40*	85.5*	85.0*	85.3*	4.91	80.7*	655.6	42.1
$O \to$ ConBART	**-1.32***	**-1.32***	**-1.32***	**88.0***	**87.7***	**87.8***	4.92	**83.8***	659.6	42.3
$O \to \text{BART}_{\text{para}}$	-1.60	-1.36*	-1.48*	83.6*	85.3*	84.3*	5.07	79.7*	706.2	42.3
$O \to \text{LED}_{\text{para}}$	-1.36*	-1.33*	-1.35*	87.0*	87.3*	87.1*	5.03	82.3*	673.6	42.4

Table 5.2: **Results of document simplification systems on Newsela-auto.** For BARTScore, h is the hypothesis and r is the reference. Scores significantly higher than PG_{Dyn} are denoted with $*$ ($p < 0.005$). Significance was determined with Student's t-tests.

5.7 Results and Discussion

Results are shown in Table 5.2. We also report results for other commonly used metrics in Appendix C.2.

Context Awareness Matters. Considering all metrics, we find that text-only models that take as input either a sentence ($\text{BART}_{\text{sent}}$) or a whole document ($\text{BART}_{\text{doc}}, \text{LED}_{\text{doc}}$) underperform models whose input is more local to the input sentence, either because they work at the paragraph level (LED_{para}) or because they take both the complex sentence and its local document context as input (ConBART). In other words, models that have access to a *local document context* (LED_{para}, ConBART) perform best overall.

LED vs BART. LED models ($\text{LED}_{\text{doc/para}}$) outperform their standard BART model counterpart ($\text{BART}_{\text{doc/para}}$) showing that modified self-attention is not only more efficient but also more precise than standard self-attention in the case of long input.

The Utility of Planning. Plan guided models (4th horizontal block in Table 5.2) outperform their standard counterpart on all metrics, showing once again that a predicted plan has a positive impact on simplification, even when the simplification model already has access to document context. This is further supported by the fact that models guided by an oracle plan (5th block) provide even greater performance.

Comparison with the State-of-the-Art (PG_{Dyn}). $O \to$ ConBART is similar to PG_{Dyn} in that, in both cases, a document is simplified by iterating over its sentences and prediction

is guided by the local context of the sentence to be simplified. A key difference is that in PG_{Dyn}, this context is exclusively used to predict a simplification operation, while in $O \rightarrow$ ConBART it is additionally used to condition the generation of the simplified sentences. We find that adding this extra control results in significantly better scores compared to the state-of-the-art PG_{Dyn} model. This illustrates that document context has utility for both planning (predicting the correct simplification operation) and realization (simplifying a given sentence). While $\hat{O} \rightarrow \text{LED}_{\text{para}}$ achieves the best overall results of any system, it is slightly outperformed by $O \rightarrow$ ConBART when oracle plans are used, suggesting that an improved planner would provide better simplifications when used by $\hat{O} \rightarrow$ ConBART over $\hat{O} \rightarrow \text{LED}_{\text{para}}$.

System	Fluency			Adequacy			Simplicity			Mean
	Minor	Major	All	Minor	Major	All	Minor	Major	All	
Reference	90.9*	**96.0**	93.4	80.8	70.7*	75.8	83.8*	82.8*	83.3	84.2
PG_{Dyn}	91.9*	94.9	93.4	**83.8**	73.7	78.8	88.9	85.9	87.4	86.5
LED_{para}	**98.0**	92.9	**95.5**	81.8	80.8	81.3	**92.9**	85.9	**89.4**	**88.7**
$\hat{O} \rightarrow \text{LED}_{\text{para}}$	90.9*	**96.0**	93.4	80.8	**82.8**	**81.8**	83.8*	90.9	87.4	87.5
$\hat{O} \rightarrow$ ConBART	89.9**	**96.0**	92.9	81.8	79.8	80.8	86.9	**91.9**	**89.4**	87.7

Table 5.3: **Human evaluation results for selected simplification systems.** The *minor* group includes those examples with a reading-level transition of 2 levels (e.g. 0-2, 1-3, etc.), whereas the *major* class includes those of 3-4 levels. Each of these groups make up half of the entire set. Ratings significantly different from the highest score in each column are denoted with $*$ ($p < 0.05$) and $**$ ($p < 0.01$). Significance was determined with two proportion Z-tests.

Human Evaluation Results from the human evaluation are shown in Table 5.3. To better identify where each model excels, we report separate scores for test paragraph pairs with minor (reading-level transition of 2) and major (>2) degrees of simplification, as well as total average scores.

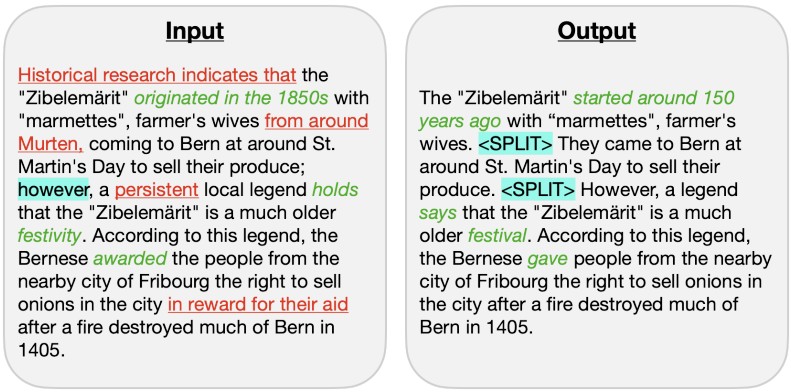
On fluency, all of the systems achieve very high ratings, which is unsurprising given the recognized ability of modern language models to produce highly fluent texts. For adequacy, $\hat{O} \rightarrow \text{LED}_{\text{para}}$ achieves the highest overall score, closely followed by LED_{para} and $\hat{O} \rightarrow$ ConBART. In terms of simplicity, LED_{para} and $\hat{O} \rightarrow$ ConBART equally achieve the highest score. Across all criteria, LED_{para} achieves the highest average ratings, although very few scores are significantly better than other systems. One potential limitation of the paragraph-level models, however, is that their context does not extend beyond the paragraph being simplified, meaning that some important information could be missing particularly at the beginning or end of the input (see the third document of Figure C.1 where anaphora resolution fails at the beginning of the second paragraph). The fact that the human evaluation is also done at the paragraph level could potentially mask some of these issues.

When considering performance differences between the minor and major simplification groups, we observe some clear trends. Systems that are not guided by a high-level document plan or do not have access to some contextual information during generation (PG_{Dyn} and LED_{para}) perform notably worse on examples requiring major simplification than they do on minor cases. Conversely, the models with both of these features appear to either perform equally as well or even excel on major cases. This suggests potential conservativity in the simplifications performed by PG_{Dyn} and LED_{para}.

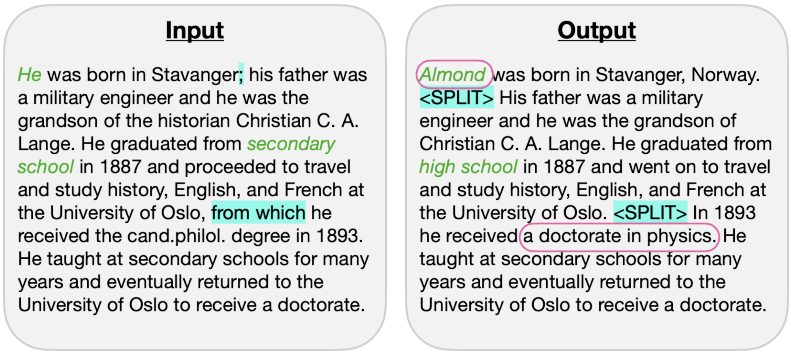
Another interesting observation is the relatively low ratings given to the references compared

to the system outputs. In particular, they receive a much lower adequacy score than any other system on major cases. This could perhaps be a result of the systems generating outputs that bear more of a resemblance to the inputs than those written by humans (see faithfulness BARTScores in Appendix C.2). For instance, human editors might have been able to confidently delete more content, or refer to some of the information in different paragraphs which the evaluators were not privy to. Despite this, the references still receive fluency ratings competitive with the other systems.

Example Simplifications Figure 5.3 shows some example simplification outputs from $\hat{O} \rightarrow \text{LED}_{\text{para}}$. These are paragraph-level extracts from larger document outputs, which are provided in Appendix C.3. Due to licensing constraints imposed by Newsela, we use out-of-domain documents from the WikiLarge dataset (Zhang and Lapata, 2017) in these examples.



(a) a good result



(b) a poor result

Figure 5.3: **Example WikiLarge simplification output extracts from $\hat{O} \rightarrow \text{LED}_{\text{para}}$** (a target reading-level of 3 was used in each case). Note that these are small extracts from larger documents shown in Appendix C.3. Deletions are underlined and in red; rephrasings are *italicised and in green*; splitting points are highlighted in cyan; and factual errors are circled.

5.8 Model Efficiency

There are various others factors to consider when comparing systems, beyond their raw performance. For instance, the size of the model(s) and how much time/resources are required for each to perform inference, are important practical considerations that must be made when selecting a model for real-world use. As such, we compare each system based on the time taken

to simplify the test set and their total parameter counts. Table 5.4 shows these results.

All inference processes were run on a single Nvidia A40 GPU, using a batch size of 16, 32 CPU workers for data loading, and a beam size of 5 for generation. Algorithm 1 shows the process used to handle dynamic context generation for appropriate models. As each document needs to be simplified autoregressively at the sentence level, we construct batches of sentences with the same index from different documents in order to speed up processing. Note that this could potentially be further optimized (e.g. via parallelism) and merely serves as a reasonable baseline algorithm.

In our case, any system that uses a plan requires a second model, approximately doubling the number of parameters that must be loaded. These pipeline setups also naturally add to overall inference time. Moreover, both plan pipelines and ConBART make use of dynamic context, which imposes an autoregressive bottleneck on the simplification of individual documents.

Because of the linearly scaling attention mechanism, Longformer-based models are the fastest of proposed systems. Because of this and its overall high performance, we recommend LED_{para} in situations where time or computing resources are at all limited. Alternatively, $\hat{O} \rightarrow$ ConBART offers a good compromise that provides the high performance of a plan-guided system while mitigating further increases to inference time. This is because it uses the same autoregressive protocol as the planner and can therefore share the generated context representations.

Algorithm 1 Generation strategy for systems using a dynamic context mechanism. Inference is performed autoregressively in batches containing 1 sentence per document. At the end of each time step, the simplified sentences are encoded for use within the context of the next step. This naturally extends to the paragraph-level case by replacing sentences with paragraphs.

procedure DynamicGeneration(*test_set*)
    $g \leftarrow load\_planner()$
    $h \leftarrow load\_simplifier()$
    $max\_idx \leftarrow \max_{C \in test\_set} |C|$
    for $i \leftarrow 1$ to max_idx **do**
        $sents \leftarrow \{c_i \mid C \in test\_set\}$ ▷ ith sentence from each document
        $context \leftarrow load\_context(sents)$
        if pipeline system **then**
            $plans \leftarrow g(sents, context)$
            $sents \leftarrow plans + sents$ ▷ Prepend plans to texts
        end if
        $preds \leftarrow h(sents, context)$
        $context \leftarrow update\_context(preds)$
    end for
end procedure

5.9 Limitations

Newsela Dataset One limitation to this study is our use of the Newsela dataset. Because this requires a license to access, researchers cannot fully reproduce our work without first obtaining permission from Newsela Inc. Unfortunately there is currently no other large dataset offering high quality aligned documents for simplification under an open source license. The only other datasets so far used for document-level simplification are based on WikiLarge, which has very

	Inference Time ↓	**# Params ↓**
BART_{doc}	182.6	140
$\text{BART}_{\text{sent}}$	54.0	140
$\text{BART}_{\text{para}}$	68.9	140
LED_{doc}	49.1	162
LED_{para}	45.9	162
ConBART	74.7	156
PG_{Dyn}	76.6	154+140
$\hat{O} \rightarrow \text{BART}_{\text{para}}$	119.1	154+140
$\hat{O} \rightarrow \text{LED}_{\text{para}}$	103.3	154+162
$\hat{O} \rightarrow$ ConBART	82.7	154+156

Table 5.4: **Model efficiency statistics.** All times are in milliseconds and model parameters are in millions. Inference times are calculated on the test set and normalised by the total number of sentences (i.e. # ms per sentence).

poor and inconsistent alignments at the document-level (Xu et al., 2015; Sun et al., 2021), as also discussed in the previous chapter.

Paragraph-Level Human Evaluation In order to reduce complexity, our human evaluation was performed on paragraphs rather than full documents. As a result, there is a potential limit to the accuracy of human judgements when certain discourse phenomena are present. For example, important information may be excluded from a specific output paragraph (therefore prompting a low adequacy rating), but this could actually be present in a different part of the true simplified document.

Monolinguality This study focused entirely on simplification for English-language documents. Reproducing the proposed systems for use on other languages would require dedicated datasets of similar scale, along with sentence/paragraph alignments and operation labels (which likely do not currently exist). Further, the nature of simplification in other languages may differ quite a lot from English with respect to the types of operations that are performed, potentially reducing the suitability of the proposed framework.

Generalized Target Audience We approach this study with our definition of "simplification" being based on that of a generalized audience, following the standard set out by the assigned reading-levels of the Newsela dataset. Existing works often outline the intent for their systems to be used to simultaneously assist a wide array of different target users, such as those with cognitive impairments, non-native speakers, and children (Maddela et al., 2021; Garbacea et al., 2021; Sun et al., 2021). However, they rarely go into any detail about which simplification strategies work for each of these different groups or perform human evaluation with annotators from the same target demographics (Gooding, 2022). As such, we acknowledge that using our systems for a specific demographic might prove insufficient to enable their consumption of media without first making further revisions to support their precise needs.

5.10 Conclusion

In this chapter, we developed a range of document simplification models that are able to use different combinations of text, context, and simplification plans as input, with several models outperforming the model we proposed in the previous chapter, both on automatic metrics

and according to human judgements. Our results show that a high-level representation of the document can be useful for low-level surface realization as well as global planning. Further, simplification models with access to local document context, either by working at the paragraph level or handling an additional input representation of surrounding sentences, lead to better meaning preservation than those that operate on individual sentences. We concluded by evaluating the model efficiency of each system and making recommendations for their selection under different circumstances.

Chapter 6

A Learned Reference-Less Metric for Sentence Simplification

Contents

Automatic evaluation for sentence simplification remains a challenging problem. Most popular evaluation metrics require multiple high-quality references – something not readily available for simplification – which makes it difficult to test performance on unseen domains. Furthermore, most existing metrics conflate simplicity with correlated attributes such as fluency or meaning preservation. In this chapter, we propose a new learned evaluation metric (SLE) which focuses on simplicity, outperforming almost all existing metrics in terms of correlation with human judgements.

6.1 Introduction

Text simplification involves the rewriting of a text to make it easier to read and understand by a wider audience, while still expressing the same core meaning. This has potential benefits for disadvantaged end-users (Gooding, 2022), while also showing promise as a preprocessing step for downstream NLP tasks (Miwa et al., 2010; Mishra et al., 2014; Štajner and Popovic, 2016; Niklaus et al., 2016). As discussed in Chapters 4 and 5, despite some recent work considering simplification of entire documents (Sun et al., 2021) the majority of work focuses on individual sentences, given the lack of high-quality document-level resources (Nisioi et al., 2017; Martin et al., 2020, 2021).

Metric	Simplification	Semantic	Ref-less
BLEU	✗	✗	✗
BERTScore	✗	✓	✗
QuestEval	✗	✓	✓
SARI	✓	✗	✗
FKGL	✓	✗	✓
Lens	✓	✓	✗
SLE	✓	✓	✓

Table 6.1: Desirable attributes of popular simplification evaluation metrics — whether they are designed with simplification in mind, use semantic representations, or do not require references.

A major limitation in evaluating sentence simplification is that most popular metrics require high-quality references, which are rare and expensive to produce. This also makes it difficult to assess models on new domains where labeled data is unavailable. Another limitation is that many metrics evaluate simplification quality by combining multiple criteria (fluency, adequacy, simplicity) which makes it difficult to determine where exactly systems succeed and fail, as these criteria are often highly correlated — meaning that high scores could be spurious indications of simplicity (Scialom et al., 2021b). Table 6.1 describes how popular metrics conform to various desirability standards.

We propose SLE (**S**implicity **L**evel **E**stimate), a learned reference-less metric that is trained to estimate the simplicity of a sentence. Different from reference-based metrics (which estimate simplicity with respect to a reference), SLE can be used as an absolute measure of simplicity, a relative measure of simplicity gain compared to the input, or to measure error with respect to a target simplicity level. We will focus primarily on simplicity gain with respect to the input in this chapter and show that SLE is highly correlated with human judgements of simplicity, competitive with the best performing reference-based metric. We also show that, when controlling for meaning preservation and fluency, many existing metrics used to assess simplifications do not correlate well with human ratings of simplicity.

6.2 A Metric for Simplicity

The SLE Metric. We propose SLE, a learned metric which predicts a real-valued simplicity level for a given sentence without the need for references. Given some sentence t, the system predicts a score $\text{SLE}(t) \in \mathbb{R}$, with high values indicating higher simplicity. This can not only be used as an absolute measure of simplicity for system output $\hat{y}$, but also to measure the simplicity gain relative to input x:

$$\Delta\text{SLE}(\hat{y}) = \text{SLE}(\hat{y}) - \text{SLE}(x) \tag{6.1}$$

In this chapter we primarily focus on ΔSLE, as it is the most applicable under common sentence simplification standards.

Model. As the basis for the metric, we fine-tune a pretrained RoBERTa (Liu et al., 2019) model[26] to perform regression over simplicity levels given sentence inputs, using a batch size of 32 and $lr = 1e^{-5}$.

[26]We fine-tune the pretrained `roberta-base` model from https://huggingface.co/roberta-base with an added regression head.

Data. We use Newsela (Xu et al., 2015), which consists of 1,130 news articles manually rewritten at five discrete reading levels, each increasing in simplicity. Existing works often assume sentences have the same reading level as the document they are from (Lee and Vajjala, 2022; Yanamoto et al., 2022); however, we expect there to be a lot of variation in the simplicity of sentences within documents and overlap across levels. As such, merely training to minimize error with respect to these labels would likely result in mode collapse within levels (peaky, low-entropy distribution) and strong overfitting to the Newsela corpus. To address this mismatch between document- and sentence-level simplicity, we take the following two mitigating steps to allow the model to better differentiate between sentences from documents of the same reading level.

Label Softening. We attempt to mitigate peakiness in the output distribution by softening the quantized reading levels assigned to each sentence in the training data. Specifically, we interpolate regression labels throughout overlapping class regions (± 1) according to their Flesch-Kincaid grade level (FKGL) (Kincaid et al., 1975). FKGL is a readability metric often used in education as a means to judge the suitability of books for students (high values = high complexity).

If L is the set of sentences belonging to some reading level, we define an intra-level ranking according to re-scaled, negative FKGLs:

$$\begin{aligned} f_L &= \{-\text{fkgl}(x_i) \mid x_i \in L\} \\ f'_{L,i} &= 2 \cdot \frac{f_{L,i} - \min f_L}{\max f_L - \min f_L} \end{aligned} \tag{6.2}$$

where $f'_{L,i}$ is the revised FKGL score of sentence x_i. Intuitively, this inverts FKGL scores (so that higher values = higher simplicity) and rescales them to be $\in [0, 2]$.

From this, we derive the final revised labels:

$$l'_{L,i} = f'_{L,i} - \bar{f'_L} + l_{L,i} \tag{6.3}$$

where $\bar{f'_L}$ is the mean of f'_L, $l_{L,i}$ is the reading level for the ith sentence of L, and $l'_{L,i}$ is its revised soft version. We report results for a model using softened labels (SLE) as well as a variant using the original quantized labels (SLE$_{\mathbb{Z}}$). Figure 6.1 shows the distributional differences between the original reading levels and the resulting softened versions for the training data.

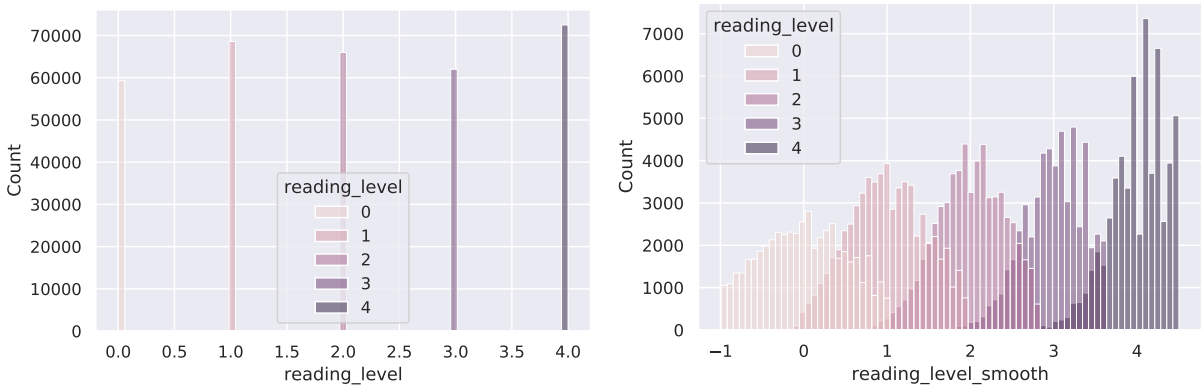
Document-Level Optimization. Given that Newsela reading levels are assigned at the document level, the labels of individual sentences are likely noisy, but approach the document label on average. We therefore observe and perform early stopping with respect to the document-level validation MSE (Mean Squared Error) and use a train/dev/test split (90/5/5) that keeps sentences from all versions of a given article together.

6.3 Experiments

6.3.1 Similarity Metrics

We compare SLE with four reference-based and two reference-less metrics previously used to assess the output of simplification models. Table 6.1 summarizes their main features.

SARI. The most commonly used evaluation metric is SARI (Xu et al., 2016), which compares n-gram edits between the output, input and references. Despite its widespread usage, SARI has known limitations. The small set of operations it considers makes it much more focused towards lexical simplifications, showing very low correlations with human ratings in cases where structural changes (e.g. sentence splitting) have occurred (Sulem et al., 2018b). As it is token-based, it is totally reliant on the references, without any robustness towards synonomy.



(a) Original (b) Softened

Figure 6.1: Distribution of (a) original quantized and (b) softened labels for sentences in the SLE training data.

BERTScore. (Zhang et al., 2019) overcomes some of these shortcomings given its use of embeddings to compute similarities, and has been found to correlate highly with human ratings of simplicity (Alva-Manchego et al., 2021), but still requires references. It is reportedly worse than SARI at differentiating conservative edits (Maddela et al., 2022) and its high correlation with simplicity ratings may be spurious (Scialom et al., 2021b). We consider the precision-oriented variant of BERTScore, which is considered the most effective for simplification according to previous work and our own experiments.

Lens. Recently, Maddela et al. (2022) propose Lens, a learnable metric specifically designed for simplification, which aims to better account for different operation types. It is trained to focus on semantic similarity without respect to writing style via a reference-adaptive loss. A simplification quality score is predicted given an output and a set of references. Lens shows higher correlations to human quality judgements than any previous metric, but still requires multiple references to work optimally.[27]

FKGL. The Flesch-Kincaid grade level (FKGL) (Kincaid et al., 1975) is a document-level metric used to measure text readability without any references. It is based on basic surface-level features like word and sentence lengths. It has seen some success in evaluating simplification (Scialom et al., 2021b). Unlike most other metrics, it does not explicitly consider the adequacy and fluency dimensions, as it is reference-less and assumes the text is already well-formed (Xu et al., 2016).

QuestEval. Scialom et al. (2021a) propose QuestEval, a reference-less metric that compares two texts by generating and answering questions between them. Although originally intended for summarization, it has shown some promise as a potential meaning preservation metric for simplification (Scialom et al., 2021b).

6.3.2 Evaluation

We evaluate SLE both in terms of its ability to perform the regression task and how well it correlates with human judgements of simplicity. For the latter we consider ΔSLE, as this

[27]The reason for Lens not being discussed or used as an evaluation metric in earlier parts of this thesis is because it was only published after the majority of work herein had already been completed.

Model	MAE ↓	Doc-MAE ↓	F1 ↑
$SLE_{\mathbb{Z}}$ (quantized)	0.825	0.544	0.401
SLE (softened)	0.924	0.448	0.402

Table 6.2: Accuracy results for reading level estimators. Errors are calculated according to the original quantized reading level labels.

conforms with what human evaluators were asked when giving ratings (to measure simplicity gain vs. the input).

Regression. To evaluate regression models we consider (i) the mean absolute error (MAE) with respect to the original quantized reading levels, (ii) the document-level error when averaging all sentence estimates from a given document (Doc-MAE), and (iii) the F1 score as if performing a classification task, after rounding estimates. We expect the best model for our purposes to achieve a lower Doc-MAE as it should better approximate true document-level simplicity labels in aggregate.

Correlation with Human Simplicity Judgments. We test the effectiveness of the metric by comparing its correlation with two datasets of human simplicity ratings: Simplicity-DA (Alva-Manchego et al., 2021) and Human-Likert (Scialom et al., 2021b). Simplicity-DA contains 600 system outputs, each with 15 ratings and 22 references, whereas Human-Likert contains 100 human-written sentence simplifications, each with ∼60 simplicity ratings and 10 references. We use all references when computing the reference-based metrics and consider the average human simplicity rating for each item.

As Simplicity-DA consists of system output simplifications, it naturally contains some sentences that are not fluent or semantically adequate. In such cases, humans would likely give low scores to the simplicity dimension as well (e.g. it is not simple to understand non-fluent text) — this is reflected in the inter-correlation between simplicity and the two other dimensions (Pearsons's r of 0.771 for fluency and 0.758 for adequacy). Thus, we only consider a subset (Simplicity-DA✓) containing those system outputs with both human fluency and meaning preservation ratings at least 0.3 standard deviations above the mean (top ∼30%)[28] which allows us to more appropriately consider how well metrics identify simplicity alone. For Human-Likert, the inter-correlation with fluency and meaning preservation are less pronounced, but do still exist (0.736 and 0.370).[29]

6.4 Results

Results on the regression task can be seen in Table 6.2. We observe that although using soft labels obviously worsens MAE with respect to the original reading levels, the document-level MAE is improved, suggesting that quantized labels lead to more extreme false negatives under uncertainty, as scores are drawn towards integer values. When treated as a classification task (by rounding predictions) both systems show similar performance (F1). This shows us that SLE is better able to approximate document-level simplicity ratings on average, with little to no drawback at the sentence level (assuming quantized labels were correct).

Correlations with human ratings of simplicity are shown in Table 6.3. The best metric on Human-Likert is Lens, closely followed by ΔSLE, with other metrics lagging quite far behind.

[28] We also exclude 166 examples that exist within the Lens training data.
[29] We do not perform any filtering on Human-Likert.

Metric	Human-Likert	Simplicity-DA✓
LENS	**0.531****	**0.429****
SARI	0.395**	0.109
BERTScore	0.389**	0.142
BLEU	0.333**	0.084
ΔSLE	**0.516****	**0.381****
$\Delta\text{SLE}_{\mathbb{Z}}$	0.479**	0.328**
FKGL	0.354**	0.260*
QUESTEVAL	0.134	0.090

Table 6.3: Absolute Pearson correlations with human judgements of simplicity. The top tier contains reference-based metrics and the bottom reference-less. * indicates significance with p-value < 0.01 and ** < 0.001.

This clearly shows the effectiveness of ΔSLE as it is able to outperform all existing metrics but for LENS, without requiring any references. On Simplicity-DA✓, metrics follow a similar rank order except for certain metrics dropping substantially (SARI, BERTScore, BLEU).[30] As Human-Likert still has moderate inter-correlation between evaluation dimensions, the large drops in performance can likely be attributed to these mostly measuring semantic similarity with references rather than the actual simplicity. Accounting for the inter-correlation between dimensions has less impact on metrics like ΔSLE and FKGL, confirming the validity of readability-based metrics as potential measures of pure simplicity.

6.5 Conclusion

In this chapter we presented SLE — a reference-less evaluation metric for sentence simplification that is competitive with or better than the best performing reference-based metrics in terms of correlation to human judgements of simplicity. We reconsider the ability of popular metrics to accurately gauge simplicity when controlling for other factors, such as fluency and semantic adequacy, confirming suspicions that many do not measure simplicity directly. We hope this work motivates further investigation into the efficacy of standard simplification evaluation techniques and the proposal of new methodologies.

6.5.1 Related Work

Štajner et al. (2014) attempt to assess each quality dimension of simplifications by training classifiers of two (good, bad) or three (good, medium, bad) classes using existing evaluation metrics as features. However, when the simplicity dimension is considered, performance was poor (Štajner et al., 2016). Later, Martin et al. (2018) were able to slightly improve this after exploring a wide range of features. However, these works do not predict real-valued estimates of simplicity nor have been adopted as evaluation metrics.

Some studies from the automatic readability assessment (ARA) literature use quantized Newsela reading levels as labels to train regression models. Lee and Vajjala (2022) do so in order to predict the readability of full documents, which does not extend to sentence simplification. Yanamoto et al. (2022) predict a reading level accuracy within an RL reward for sentence

[30] Scialom et al. (2021b) report very poor reference-based metric correlations on Human-Likert, substantially lower than our results. When discussed with the authors, they were no longer in possession of code that could reproduce their originally reported results.

simplification, but do so using the reading levels that were assigned to each document. This too does not transfer well to sentence-level evaluation, given the imprecision and noise introduced by the use of quantized ratings that were assigned at the document level. These approaches have not been applied to the actual evaluation of sentence simplification systems.

6.5.2 Future Directions

In this chapter we explored the efficacy of SLE as a measure of raw simplicity or relative simplicity gain (ΔSLE). However, given the flexibility of not needing to rely on references, SLE can potentially be used in other ways too. For example, one could measure an error with respect to a target simplicity level, l^*:

$$\epsilon\text{SLE}(\hat{y}) = |\text{SLE}(\hat{y}) - l^*| \tag{6.4}$$

This could be useful in the evaluation of controllable simplification systems, which should be able to satisfy the simplification requirements of specific user groups or reading levels (Martin et al., 2020; Yanamoto et al., 2022). As it is trained with aggregate document-level accuracy in mind, SLE could also potentially be used to evaluate document simplification, either via averaging sentence scores or using some other aggregation method. In Chapter 7, we will experiment with both of these applications.

6.5.3 Limitations

The SLE metric model is trained entirely on English-language data and therefore will not be effective for evaluating simplification in other languages. Producing a multilingual version of the metric could be possible by using either different datasets or adapting other methods from the multilingual NLP literature. Furthermore, as SLE has been primarily trained on news articles, it could exhibit a drop in performance quality when used to evaluate text from specialized domains that use vocabularies likely not encountered during training (e.g. medical, legal domains). In such cases, producing a domain-specific version of SLE via specialized pretraining or fine-tuning should also be feasible, given sufficient data.

Chapter 7

Semantic Faithfulness and Out-of-Domain Performance in Document Simplification

Contents

Current automatic evaluation methods for text simplification pay relatively little attention to faithfulness to inputs, with most metrics considering simplification quality as a single dimension despite the oftentimes inverse relationship between simplicity and faithfulness. While this has received some recent attention in the case of sentence simplification, many observations do not hold when transitioning to the document-level case. In this chapter, we explore the applicability of faithfulness evaluation metrics from the summarization literature to document simplification. Furthermore, with recent advances in the mainstream accessibility of large language models, it is important to understand how capable models are at simplifying documents from unseen domains, while ensuring the semantic faithfulness of outputs. As such, we also investigate the performance of existing systems when applied to unseen data (where reference simplifications are unavailable) and test the ability of LLMs to perform zero-shot document simplification. We find that plan-guided systems proposed in earlier chapters perform the best at simplifying out-of-domain documents, but automatic semantic faithfulness metrics struggle to accurately recognize

this without being able to benchmark the simplicity/faithfulness trade-off against reference simplifications.

7.1 Introduction

Recent investigation into the quality of sentence-level test data and system outputs has found many instances of factual incoherence not previously detected during data collection or evaluation (Devaraj et al., 2022). This raises questions of how faithful simplifications are to their inputs and whether or not these concerns also apply to the document-level task. Although attempts to automatically evaluate semantic faithfulness in sentence simplification have seen limited success (Devaraj et al., 2022), summarization literature contains a lot of work that could be transferable to document simplification (Laban et al., 2022; Fabbri et al., 2022).

Despite their ability to generate highly fluent texts, the commonly used end-to-end neural systems rely heavily on the quality of data they are trained on. In text simplification, training data is scarce, with most existing corpora being compiled via automatic alignment methods. These are known to contain a lot of noise and imbalanced distributions of possible transformation types (Sulem et al., 2018c; Jiang et al., 2020). As a result, end-to-end systems are very conservative in the amount of editing they perform, often making little to no changes to the input (Alva-Manchego et al., 2017). With some works observing an inverse correlation between semantic adequacy and simplicity (Schwarzer and Kauchak, 2018; Vu et al., 2018), it raises the question of how to actually determine whether an output is sufficiently faithful to the input, when some amount of degradation is a requirement for performing simplification.

As discussed in Chapter 6, evaluation poses additional challenges, with the suitability of popular automatic metrics remaining unclear (Alva-Manchego et al., 2021; Scialom et al., 2021b). As most automatic metrics require multiple, high-quality references, studies are usually restricted to a small pool of imperfect datasets that include reference simplifications, making it difficult to gauge how well systems actually perform on real-world out-of-domain data.

The main contributions of this chapter are as follows: (i) an investigation into evaluating the semantic faithfulness of document simplification outputs and its relationship with model conservativity; (ii) a comparison of state-of-the-art system performances on in- and out-of-domain datasets; and (iii) a comparison between state-of-the-art document simplification systems and the zero-shot capabilities of ChatGPT.

7.2 Related Work

Faithfulness in Simplification. The goal of text simplification is not only to make a text easier to read, but also to ensure the same information is conveyed. Until recently, explicit evaluation of the faithfulness of simplification outputs has been somewhat overlooked. In general, semantic adequacy with the original complex text is only manually considered during human evaluation, with automatic metrics mostly focusing on semantic similarity to reference simplifications. Even during human evaluation, the typical criteria for semantic adequacy is rather relaxed, demanding only that the text continues to generally convey the core meaning.

A recent manual investigation into common faithfulness errors in both system outputs and test data found many issues undetected by common evaluation metrics (Devaraj et al., 2022). However, this analysis was limited to sentence-level simplification and many of the issues uncovered do not extend to the document-level case — a limitation which the authors acknowledge. For instance, content that appears to be wrongly inserted or deleted when considering a pair of aligned sentences in isolation could easily have been moved to or from other sentences in the

same document. They also attempted to train a model to automatically evaluate faithfulness, to limited success.

Outside of explicit evaluation, some sentence simplification works have considered faithfulness within their training processes. Guo et al. (2018) train a multi-task simplification model with entailment as an auxiliary task. Nakamachi et al. (2020) integrate the semantic similarity between an input and generated output within the reward function of their reinforcement learning (RL) framework for simplification, while Laban et al. (2021) include an inaccuracy guardrail that rejects generated sequences that contain named entities not present in the input. Ma et al. (2022) attempt to improve performance by down-scaling the training loss of examples with similar entity mismatches. However, these works either do not explicitly evaluate the faithfulness of their system outputs or find that they do not actually prevent the final model from generating unfaithful simplifications.

On the related task of summarization, there has been much more work on this front (Maynez et al., 2020; Pagnoni et al., 2021). The evaluation of semantic faithfulness in summarization is broadly split into either entailment-based (Falke et al., 2019; Kryscinski et al., 2020; Koto et al., 2022) or question answering (QA)-based methods (Wang et al., 2020; Durmus et al., 2020; Scialom et al., 2021a), with comprehensive benchmarks being established for each (Laban et al., 2022; Fabbri et al., 2022).

Simplicity Evaluation. The most popular evaluation metrics (e.g. SARI, BERTScore) used in simplification generally require multiple high-quality references to perform as intended (Xu et al., 2016; Zhang et al., 2019). This poses problems for practitioners seeking to apply simplification models to novel data, as it is impossible to gauge performance without going through the difficult and expensive process of manually creating references — a problem that is exacerbated in the document-level case.

As outlined in Chapter 6, investigations into the validity of these metrics also raise concerns over whether they do in fact measure simplicity itself and not correlated attributes like semantic similarity to references (Scialom et al., 2021b). However, reference-less sentence simplicity metrics (showing high correlations with human judgements) like SLE (Chapter 6) could allow for meaningful evaluation of out-of-domain performance. Despite this, the efficacy of existing evaluation metrics when applied at the document level remains unexplored.

7.3 Experimental Setup

Our global aim is to perform a more thorough investigation into the performance of existing document simplification systems, with particular focus on semantic faithfulness to the input. We also investigate the out-of-domain performance of existing systems and reconsider how this should be evaluated given a lack of diverse references. Finally, we examine the relative ability of large language models (LLMs) to perform the document simplification task in a zero-shot setting.

7.3.1 Data

We primarily rely on the Newsela (Xu et al., 2015) corpus, which is generally seen as the gold-standard document simplification dataset. It consists of 1,130 English news articles that have been manually rewritten by professional editors at five different discrete reading levels (0-4) of increasing simplicity.[31]

[31] We use the same document-level test set as in Chapter 4.

As we focus on reference-less evaluation, we can also consider model performance on out-of-domain data for which we have no reference simplifications. For this, we use standard English Wikipedia (EW) articles from Wiki-auto (Jiang et al., 2020). Although EW corpora with automatically aligned reference simplifications from Simple English Wikipedia (SEW) exist, they are known to contain a lot of noise, being of particularly poor quality when considered at the document level (Xu et al., 2015).

To assess performance on longer documents, we only consider those that contain at least 10 sentences and 3 paragraphs. To diversify the domain of articles, we annotate each with a semantic type according to their WikiData (Vrandečić and Krötzsch, 2014) entry and selected articles to cover a range of diverse categories (shown in Table 7.1). We select 19 of the most common types, group them into 5 broad categories and sample articles equally from each to obtain a final test set of 1000 documents.

Category	Sub-Category	Count
Biographical	Human	500
	Musical Group	250
	Fictional Human	250
Location	City	250
	Village	250
	Commune of France	250
	City in the United States	250
Media	Film	250
	Video Game	250
	Literary Work	250
	Television Series	250
Science	Taxon	250
	Class of Disease	250
	Chemical Compound	250
	Class of Anatomical Entity	250
Industry	Business	250
	Profession	250
	Organization	250
	Automobile Model	250

Table 7.1: Distribution of Wikipedia article categories.

7.3.2 Simplification Systems

We consider several document simplification systems (at or near state-of-the-art) from existing works, which have all been trained on Newsela (Table 7.2 provides a summary of model attributes). $\mathbf{PG_{Dyn}}$ (introduced in Chapter 4) is a pipeline system that first generates a document simplification plan using high-level context, then conditions a sentence simplification model on said plan. From Chapter 5 we include three systems: (i) $\mathbf{LED_{para}}$ — a paragraph-level Longformer (Beltagy et al., 2020) model which is the best performing end-to-end system; (ii) $\hat{O} \rightarrow \mathbf{LED_{para}}$, which uses the same Longformer model, but conditions it on a plan from the same planner as PG_{Dyn}; and (iii) $\hat{O} \rightarrow$ **ConBART** — a modification of the BART (Lewis et al., 2020) architecture that attends to a high-level document context during decoding, while also conditioning on a plan.

As these Newsela-trained models have all been prefixed with target reading-level control tokens during training, we must also specify this during inference. On the out-of-domain Wikipedia task, we set the target reading-level to 3 (on a scale of 0-4) for all models. Ideally, this will result in substantial editing during simplification while limiting the over-deletion of content. We also use **ChatGPT**[32] to perform simplification in a zero-shot fashion.[33] This is done to benchmark how well LLMs can perform the document simplification task out-of-the-box. We preliminarily tested various potential prompts and selected the following as a reliable option for our evaluation experiments: *Rewrite the following document, making it easier to understand: [document]*. See Appendix D.1 for examples of the prompts we tested.

System	Description
PG_{Dyn}	- Sentence-level text input - Plan-guided
LED_{para}	- Paragraph-level text input - No plan-guidance - Longformer-based end-to-end model
$\hat{O} \rightarrow \text{LED}_{\text{para}}$	- Paragraph-level text input - Plan-guided - Longformer-based simplification component
$\hat{O} \rightarrow$ ConBART	- Sentence-level text input - Plan-guided - Simplification model with cross-attention over high-level representation of document sentences

Table 7.2: Descriptions of the different document simplification systems we consider.

7.3.3 Evaluating Faithfulness

We consider two existing reference-less metrics for evaluating faithfulness: **SummaC** (an NLI entailment-based metric) (Laban et al., 2022) and **QAFactEval** (a QA-based metric) (Fabbri et al., 2022). Both are from the summarization literature and should therefore be considered with a level of caution when being applied to simplification. For example, as summarization outputs are generally much shorter than their inputs, it is likely that these metrics will skew in favour of very short and concise simplifications (i.e. precision) even when too much information has been removed. In response, we also use variations of each that focus more on recall.

SummaC (Summary Consistency) (Laban et al., 2022) first works by using an out-of-the-box NLI model[34] to compute an NLI entailment matrix over a document. This is an $M \times N$ matrix of entailment scores between each of the M input sentences and N output sentences. This is transformed into a histogram form of each column and a convolutional layer is used to convert the histograms into a single score for each output sentence, which are then averaged. As such, this metric is naturally more precision-oriented and therefore could favour shorter,

[32] chat.openai.com/

[33] Given the contractual restrictions around the sharing of Newsela corpus data, we only use EW data with ChatGPT to ensure we comply with our agreements.

[34] In our case, we use an implementation that uses the version of ALBERT-xlarge from Schuster et al. (2021) fine-tuned on the Vitamin C and MNLI datasets, available at https://huggingface.co/tals/albert-xlarge-vitaminc-mnli.

lexically conservative simplifications. In response, we also compute a recall-oriented version, whereby scores are calculated for each input sentence.

QAFactEval (Fabbri et al., 2022) is a state-of-the-art QA-based metric that consists of several components within a pipeline. In order they are: answer selection → question generation → question answering → overlap evaluation → question filtering. Questions and correct answers are first generated and selected given a summary, then answers are predicted given the input document as context. For each of these, an answer overlap score is computed using the LERC metric (Chen et al., 2020), which estimates the semantic similarity between the true and predicted answers. The final result is the average of these answer overlap scores for the questions remaining after a question filtering phase (those that are considered answerable are excluded).

If an overly short simplification leads to only a few questions being generated it is possible that this could achieve high scores. Further, the process of simplification itself (lexical substitution in particular) might challenge this metric as the QA model must be be able to accurately recognize the semantic similarity between substituted phrases in order to gauge the validity of an answer. As with SummaC, we compute both precision- and recall-oriented versions of this metric. In the recall case we generate questions from the source document instead of the output.

Entity Matching. Another heuristic for assessing the semantic faithfulness of generated text is to consider the similarity between entities present in the input vs. output — sometimes referred to as entity-based semantic adequacy (**ESA**) (Wiseman et al., 2017; Laban et al., 2021; Faille et al., 2021; Ma et al., 2022). We extract named entities from input documents using the spaCy library[35] and compute the precision, recall, and F1 with respect to those found in the generated simplifications.

Conservativity. Given the nature of semantic faithfulness being tied to the input, high scores for these metrics can be obtained by overly conservative models. So, to contextualize results, we also include the average lengths of outputs (no. of tokens and sentences) as well as the **BLEU** (Papineni et al., 2002) with respect to the input. Generally, simplifications are slightly shorter than their inputs and often contain more sentences (a result of splitting). This BLEU score will give a further indication of the amount of editing that has been performed and therefore flag whether a system has potentially achieved high faithfulness scores as a result of over-conservativity.

7.3.4 Evaluating Simplicity

Most popular evaluation metrics for simplification have well documented limitations, such as their reliance on high-quality references. Furthermore, their efficacy has not been fully explored for the document-level task. Given this and the fact that the scope of our study covers performance on out-of-domain data, for which there are no references, we instead rely on reference-less alternatives.

We report the Flesch-Kincaid grade level (**FKGL**) (Kincaid et al., 1975) — a simple document readability metric with relatively high correlation to human judgements of simplicity (Scialom et al., 2021b). It is based on a multiple regression model that considers the average length of sentences and syllable count of words in the document. However, FKGL gauges simplicity in absolute terms, assuming a simpler output is universally more valuable. Because of this, it is not ideal for evaluating simplicity for specific target groups (e.g. the different reading grade levels supported by Newsela).

[35] https://spacy.io

Given that most document simplification systems target a specific reading level during generation, it would be more useful to evaluate the divergence from this target level of simplicity, rather than measuring raw simplicity alone. To this end, we report the **SLE** metric (described in Chapter 6), which is trained to predict a sentence's simplicity according to a leveling scheme similar to Newsela's reading levels. We adapt this to the document level by computing the prediction for a document Y as the mean of its sentences' scores:

$$\text{SLE}(Y) = \frac{1}{|Y|} \sum_{i=1}^{|Y|} \text{SLE}(y_i) \tag{7.1}$$

where y_i is the ith sentence of document Y. We further adapt this to our task by deriving the simplicity level error (ϵ**SLE**) of a system as the mean absolute error (MAE) between the predicted and target document reading levels:

$$\epsilon\text{SLE} = \frac{1}{N} \sum_{i=1}^{N} \left| \text{SLE}(\hat{Y}_i) - l_i \right| \tag{7.2}$$

where l_i is a target simplicity level. ϵSLE is able to estimate how close a simplification is to the target reading level without relying on any references, allowing it to avoid the limitations and rigidity of most other popular evaluation metrics. Although we showed that this method has shown promise for sentence-level inputs in the previous chapter, we believe it could be even more suitable as a document-level metric, as the gold labels of Newsela (which SLE is trained on) are assigned at the document- rather than the sentence-level. Although individual sentences within a document might have diverse simplicity levels, in aggregate they should converge to the global document level, following the central limit theorem. Figure 7.1 shows the distribution of SLE scores predicted for reference sentences belonging to each original Newsela reading level group. We can see that although the mean is approximately equal to the reading level, there is substantial diversity within each group.

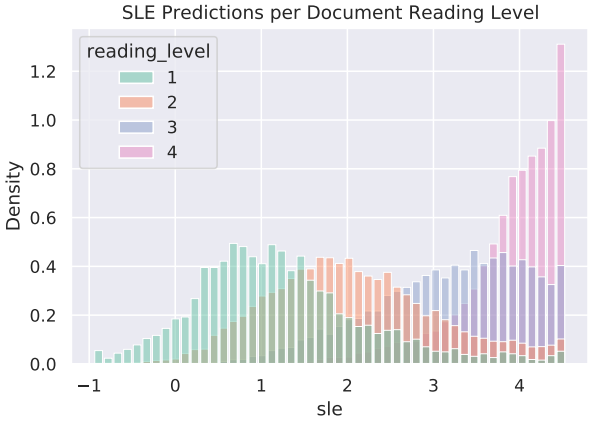


Figure 7.1: Distribution of SLE scores for reference sentences within each Newsela reading level group.

7.3.5 Human Evaluation

To confirm system performance on the out-of-domain data, we also conduct a human evaluation. Due to the difficulty of comparing full documents, we follow Chapter 5 in evaluating at

the paragraph-level. We present annotators with a complex paragraph and an extract from a generated simplification corresponding to that paragraph. Evaluators are then asked to judge whether the generated text is fluent, consistent with respect to, and simpler than the input.

We randomly sample 250 paragraphs from the test set that contain between 3-6 sentences. We consider the outputs from all tested systems, except ChatGPT (due to the overly short outputs, we cannot meaningfully align them to complex paragraphs), and ask annotators to rate them on each dimension. We pose each as a binary (yes/no) question in order to avoid the inter-annotator subjectivity that is inherent when using a Likert scale. The proportion of positive ratings is used as the final score.

Human judgements were obtained via the Amazon Mechanical Turk crowdsourcing platform. Annotators were sourced from majority English speaking countries (AU, CA, GB, IE, NZ, US) and were paid $0.18 USD per evaluation. According to preliminary tests, under this scheme participants earn approximately $16.2 USD per hour — which is higher than the minimum hourly wage of all countries. The form and instructions presented to human evaluators is shown in Figure 7.2.

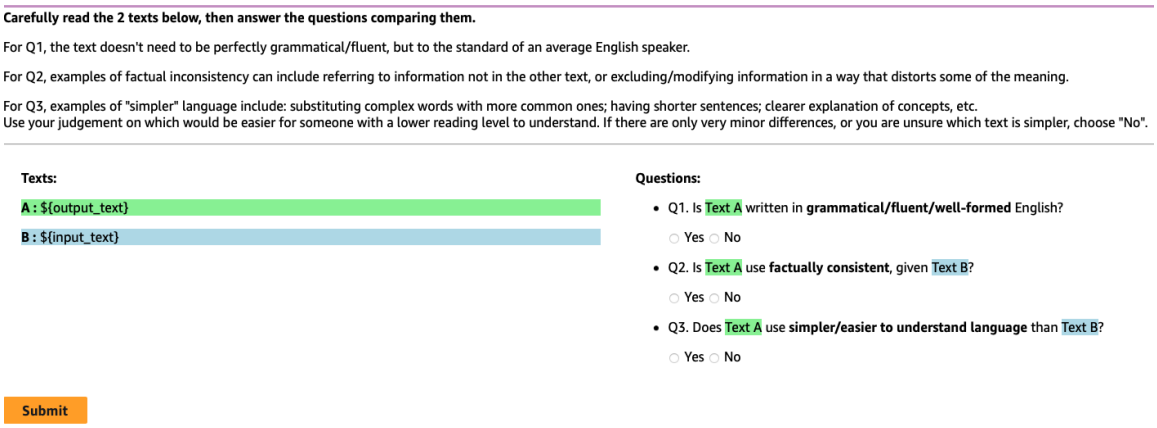


Figure 7.2: Submission form presented to annotators during the human evaluation.

7.4 Results and Discussion

7.4.1 Newsela Performance

Faithfulness and simplicity results on the Newsela test set are shown in Tables 7.3 and 7.4, respectively.

End-to-End vs Planning. The end-to-end model (LED_{para}) achieves the highest scores across all three faithfulness metrics. However, it also has the highest BLEU score, suggesting more conservativity than other systems — a commonly seen limitation of end-to-end simplification models (Nisioi et al., 2017; Alva-Manchego et al., 2017). Further, it produces outputs that are much longer than the references or any other system and achieves the worst simplicity performance, both in terms of absolute (FKGL) and relative (ϵSLE) criteria. The plan-guided models, however, achieve faithfulness results not too far from LED_{para} while still generating outputs much closer in length and BLEU to the references. This suggests that plan-guidance allows models to avoid conservativity and make necessary edits to achieve high simplicity, although at the cost of some reduced faithfulness to the input.

Local vs Global Context. The simplification components of the plan-guided models each consider document context differently. While PG_{Dyn} has no notion of document context, $\hat{O} \rightarrow \text{LED}_{\text{para}}$ considers the local, token-level context of the surrounding paragraph, and $\hat{O} \rightarrow$ ConBART considers a high-level representation of more global context (SBERT encodings of 26 surrounding sentences). Results show that the more local paragraph context leads to slight improvement in terms of faithfulness, but a reduction in simplicity performance. $\hat{O} \rightarrow$ ConBART achieves the best overall simplicity (FKGL) as well as ϵSLE. Interestingly, both $\hat{O} \rightarrow$ ConBART and $\hat{O} \rightarrow \text{LED}_{\text{para}}$ are much better than the other systems at simplifying to the highest level of simplicity, mirroring the human evaluation observations of Chapter 5 where plan-guided, context-aware systems appeared particularly strong in cases where major editing is required.

Reference Superiority. Although plan-guided systems seem less prone to over-conservative editing, they still appear to be more conservative than the references. The fact that they all achieve higher faithfulness scores than the references (which are assumed to be sufficiently faithful) further consolidates this point. Additionally, the references achieve much better FKGL and ϵSLE than any system. This shows that there is indeed a trade-off between faithfulness and simplicity, and there is still much general improvement that can be made to current document simplification systems.

System	**SummaC ↑**			**QAFactEval ↑**			**ESA ↑**			**Length**		**BLEU**
	P	R	F1	P	R	F1	P	R	F1	Tokens	Sents	
Input	-	-	-	-	-	-	-	-	-	866.9	38.6	-
Reference	0.61	0.47	0.53	3.86	3.02	3.39	0.59	0.47	0.52	671.5	42.6	44.6
PG_{Dyn}	0.65	0.47	0.55	3.95	3.10	3.47	**0.61**	0.48	0.53	667.2	42.6	47.6
LED_{para}	**0.66**	**0.52**	**0.58**	**4.00**	**3.29**	**3.61**	0.60	**0.51**	**0.55**	712.9	44.9	51.5
$\hat{O} \rightarrow \text{LED}_{\text{para}}$	0.65	0.50	0.57	3.98	3.16	3.52	0.60	0.49	0.54	683.1	42.8	49.1
$\hat{O} \rightarrow$ ConBART	0.65	0.48	0.56	3.95	3.11	3.48	0.60	0.48	0.53	671.6	43.0	47.5

Table 7.3: Faithfulness and Conservativity results for systems evaluated on the Newsela test set.

System	**FKGL ↓**	**ϵSLE ↓**				
		1	2	3	4	**Total**
Reference	4.93	0.22 (1.12)	0.21 (1.97)	0.24 (3.11)	0.22 (3.84)	0.23
PG_{Dyn}	4.98	0.30 (1.24)	**0.22** (2.02)	0.22 (3.07)	0.32 (3.69)	0.26
LED_{para}	5.15	0.29 (1.06)	0.24 (1.92)	0.24 (2.97)	0.34 (3.67)	0.28
$\hat{O} \rightarrow \text{LED}_{\text{para}}$	5.09	**0.26** (1.13)	0.24 (1.87)	0.23 (3.02)	0.30 (3.72)	0.26
$\hat{O} \rightarrow$ ConBART	**4.96**	0.28 (1.23)	**0.22** (1.98)	**0.21** (3.06)	**0.29** (3.73)	**0.25**

Table 7.4: Simplicity results for systems evaluated on the Newsela test set. Numbers in parentheses are the raw SLE averages (0-4).

7.4.2 Out-of-Domain Performance

Out-of-domain performance is assessed by testing the Newsela-trained models on EW data. Results are shown in Tables 7.5 and 7.6. Table 7.7 shows the relative change in automatic evaluation results when moving from in- to out-of-domain data (using the same target reading level of 3).

End-to-End vs Planning. The end-to-end, Longformer model (LED_{para}) produces much shorter output documents than the plan-guided models — the opposite of what is seen for Newsela. Although EW articles have slightly longer paragraphs on average, this is likely a result of over-deletion due to a lack of plan-guidance, as the other paragraph-level model ($\hat{O} \rightarrow \text{LED}_{\text{para}}$) does not share this behavior. This could suggest that planning also helps models better adapt to unseen domains.

On the other hand, $\hat{O} \rightarrow$ ConBART achieves the lowest faithfulness scores out of all dedicated systems, particularly on QAFactEval. As this model attends over a wider document context, it is possible that this increase in model variance could have led to some overfitting on the Newsela data. The ConBART network architecture also contains additional layers that were not pretrained before fine-tuning on the Newsela dataset, further pointing towards potential overfitting. However, it is still close to PG_{Dyn} on SummaC and ESA, while also achieving the best simplicity scores, which could mean the lower faithfulness scores are merely a result of the trade-off with simplicity. Without reference simplifications, it seems difficult to draw conclusions before examining human evaluation results.

Sentences vs Paragraphs. In terms of simplicity, the sentence-level models (PG_{Dyn} and $\hat{O} \rightarrow$ ConBART) achieve much better (lower) FKGL and ϵSLE than the two paragraph-level models. However, like on Newsela, they are markedly outperformed by the paragraph models on faithfulness metrics, particularly in terms of precision. While paragraph models produced longer outputs on in-domain data, they now produce shorter texts than sentence-level models, particularly in terms of the number of sentences. This could indicate potential conservativity with respect to sentence splitting, or an over-deletion of sentences. The latter is somewhat likely given the fact that Wikipedia articles contain longer paragraphs than those in Newsela, meaning the paragraph-level models could be biased towards generation of the shorter Newsela lengths.

ChatGPT Limitations. ChatGPT performs quite badly compared to dedicated simplification systems. It generates outputs that are much shorter than any other system, leading to particularly bad results on recall-based metrics. It also achieves the worst simplicity results by far. This seems to suggest that ChatGPT is performing something closer to document summarization rather than simplification — not totally surprising considering ChatGPT was likely trained with much more summarization data, given the relative scarcity of simplification corpora. It is possible that this could be improved with further prompt engineering, but all prompts we tested led to similar results (Appendix D.1).

System	**SummaC ↑**			**QAFactEval ↑**			**ESA ↑**			**Length**		**BLEU**
	P	R	F1	P	R	F1	P	R	F1	Tokens	Sents	
PG_{Dyn}	0.70	0.38	0.50	3.28	2.18	2.62	0.58	0.34	0.43	614.5	40.6	31.4
LED_{para}	**0.76**	0.39	0.51	**3.78**	2.11	2.71	**0.64**	0.35	0.45	513.7	32.5	27.4
$\hat{O} \rightarrow \text{LED}_{\text{para}}$	0.73	**0.41**	**0.53**	3.61	**2.28**	**2.79**	0.62	**0.37**	**0.47**	601.5	37.0	32.0
$\hat{O} \rightarrow$ ConBART	0.68	0.38	0.49	3.10	2.06	2.48	0.57	0.33	0.42	598.4	40.5	29.5
ChatGPT	0.50	0.29	0.37	3.62	1.48	2.10	0.61	0.20	0.30	269.1	11.3	7.6

Table 7.5: Faithfulness and Conservativity results for systems evaluated on the out-of-domain Wikipedia test set.

7.4.3 Human Evaluation Results

Table 7.8 shows the results of the human evaluation.

System	FKGL ↓	ϵSLE ↓
Input	10.07	- (0.89)
PG_{Dyn}	4.72	**0.21** (2.92)
LED_{para}	4.92	0.29 (2.78)
$\hat{O} \rightarrow LED_{para}$	5.02	0.31 (2.76)
$\hat{O} \rightarrow$ ConBART	**4.58**	**0.21** (3.00)
ChatGPT	8.87	1.61 (1.39)

Table 7.6: Simplicity results for systems evaluated on the out-of-domain Wikipedia test set. Numbers in parentheses are the raw SLE averages (0-4).

System	SummaC ↑		QAFactEval ↑		ESA ↑		BLEU	FKGL ↓	ϵSLE ↓
	P	R	P	R	P	R			
PG_{Dyn}	0.04	-0.11	-0.66	-0.89	-0.02	-0.13	-14.09	-0.11	0.17 (-0.25)
LED_{para}	0.09	-0.13	-0.19	0.05	-0.15	-0.09	-21.0	-0.09	0.22 (-0.28)
$\hat{O} \rightarrow LED_{para}$	0.07	-0.1	-0.33	-0.84	0.03	-0.11	-14.55	0.06	0.24 (-0.32)
$\hat{O} \rightarrow$ ConBART	0.02	-0.11	-0.86	-1.01	-0.03	-0.15	-16.04	-0.27	0.09 (-0.14)

Table 7.7: Difference in results for target-level 3 when moving from the in-domain Newsela to the out-of-domain Wikipedia test set.

Despite achieving the best fluency, the end-to-end model (LED_{para}) under-performs on faithfulness and simplicity compared to the plan-guided systems. This corroborates the automatic results in suggesting that planning can help systems to adapt better to unseen domains. The best overall results are achieved by PG_{Dyn}, but this can largely be attributed to its very high simplicity ratings as it falls below $\hat{O} \rightarrow$ ConBART in terms of faithfulness. Although this once again shows there may be no free lunch when accounting for both faithfulness and simplicity, $\hat{O} \rightarrow$ ConBART manages to achieve the best balance between the two.

In contrast to what is observed via the automatic faithfulness metrics, sentence-level systems also appear to outperform paragraph-level ones. This could be a result of the paragraph models having a wider text window in which to make potential mistakes/hallucinations, whereas the sentence-level systems are more constrained. The EW paragraphs are also longer on average than the Newsela ones used to train these models, which could result in them failing to maintain all information when extending to longer input sizes (this is alluded to by the drop in the number of sentences in paragraph-level model outputs when moving to the EW domain, Table 7.5).

Interestingly, $\hat{O} \rightarrow$ ConBART achieves the highest faithfulness ratings despite achieving the lowest results on each automatic faithfulness metric. This divide between human judgements and automatic results suggests current faithfulness metrics from the summarization literature are not entirely sufficient for reliably reporting system performance on the document simplification task, even when considering both precision and recall. As such, it appears some of the issues outlined in Devaraj et al. (2022) for sentence simplification do in fact extend to the document-level case — particularly with respect to the efficacy of existing faithfulness and semantic similarity metrics when applied to simplification.

System	Flu	Faith	Simp	Mean
PG_{Dyn}	0.898	0.732	**0.820**	**0.817**
LED_{para}	**0.932**	0.632	0.664	0.743
$\hat{O} \rightarrow LED_{para}$	0.890	0.684	0.760	0.778
$\hat{O} \rightarrow$ ConBART	0.890	**0.760**	0.764	0.805

Table 7.8: Human evaluation results on Wikipedia.

7.5 Limitations

Paragraph-Level Human Evaluation As with Chapter 5, our human evaluation was performed using only paragraph-level extracts from simplified documents, rather than the entire documents themselves. This was done to limit the complexity of each human evaluation task as full-document annotation would likely be challenging for many workers. Because of this, it is possible that certain long-distance discourse phenomena are not properly considered during the evaluation. For example, important information may be excluded from a specific output paragraph, but may actually be present in a different part of the document. However, given the iterative nature of most systems tested, we expect such cases to be uncommon.

English Only The datasets and systems we investigate are applicable only to English. It is possible that many of the insights from the study could equally apply in the case of other languages; however, independent analyses would need to be carried out to confirm this. Additionally, many of the evaluation metrics used (e.g. both simplification metrics – FKGL and SLE) are built specifically with English text in mind and therefore would not easily be adaptable to equivalent evaluations of simplification in other languages.

7.6 Conclusion

In this chapter we conducted an investigation into the semantic faithfulness of outputs from state-of-the-art document simplification systems. By leveraging recent advancements in automatic faithfulness evaluation for summarization and the reference-less evaluation of simplification, we were also able to carry out the first in-depth analysis of simplification performance on out-of-domain data. In addition, we explored the ability of ChatGPT to perform zero-shot document simplification, finding that it performs rather poorly, with a tendency to digress into summarization.

While a state-of-the-art end-to-end model appears to achieve the best in-domain faithfulness results, it is also much more conservative than plan-guided systems, generating outputs with low simplicity. Plan-guided systems also appear better at adapting to unseen domains, but we continue to observe a general trade-off between faithfulness and simplicity. Consideration of this trade-off using only automatic metrics is challenging for out-of-domain settings as it is unclear what exactly constitutes a sufficient level of faithfulness without having references to use as a baseline.

Human evaluation results indicate that plan-guided, sentence-level simplification systems produce outputs with the highest faithfulness to the input when moving across domains — a phenomenon not captured by the automatic faithfulness metrics. This highlights the need for further exploration into automatic methods of faithfulness evaluation for simplification systems. We hope our work motivates future investigations into training methods and architectures that can allow simplification systems to effectively adapt to unseen domains, rather than further optimizing performance on the most popular datasets.

Conclusion

In this thesis, we explored various issues concerning controllable and document-level approaches to automatic text simplification. First, we proposed strategies for mitigating the conservativity of ATS systems and allowing them to perform a wider range of transformation types. This included the creation of datasets containing instances of under-represented operations and the implementation of controllable systems capable of being tailored towards specific transformations and simplicity levels by exploiting prior knowledge concerning the suitability of each operation. We then went on to extend these strategies to document-level simplification, proposing systems that are able to consider surrounding document context to plan which sentence-level operations to perform ahead of time, allowing for both high performance and scalability to large documents. Next, we analyzed current evaluation processes and proposed a new metric that can be used to evaluate simplicity in both controllable and document-level settings without requiring any reference simplifications. Finally, we also explored strategies to automatically evaluate meaning preservation within document simplification, applying existing strategies from summarization and considering how this interacts with the notion of simplicity itself.

Overview of Findings

To summarize the main findings of this thesis, let us revisit the initial research questions set out in the introductory chapter.

How can we enable simplification systems to carry out a more diverse range of operations, despite limited representation in existing training data? In Chapter 2 we considered the previously unexplored operation of discourse-based sentence splitting. By constructing a dataset dedicated to this type of operation, we were able to show enhanced performance in its execution. In Chapter 3, by integrating this dataset into a collection of other corpora containing instances of more popular operations, we were able to train a classifier that is much more accurate in identifying appropriate simplification operations than existing works. By combining this with a controllable sentence simplification model, the resulting pipeline system performs splitting operations much more often than end-to-end systems and those trained with standard datasets, while also being manually configurable to perform specific operations.

Can we build document-level simplification systems that are capable of scaling to long inputs while also preserving discourse coherence in the resulting document? In Chapter 4 we proposed a novel framework for approaching document simplification whereby a plan of sentence-level operations is first generated and subsequently used to guide a sentence-level model, in a similar fashion to the controllable model of Chapter 3. We went on to propose a system that generates such a plan using both the internal structure of a given sentence and a high-level representation of document context, which is dynamically updated as each sentence in a document is simplified. We show that this system universally outperforms an iteratively applied

sentence-level model while also being able to scale to the long documents more effectively than an encoder-decoder that is simply given the full input document as a token-level sequence. In particular, the consideration of document context allows the system to more accurately perform context-dependent operations such as deletion and copying, in turn better preserving inter-sentence coherence of the simplified document.

Chapter 5 shows our further exploration of this strategy's potential with us proposing new simplification components that themselves make use of document context, either by iterating over larger text units (e.g. LED_{para}) or extending the system architecture to attend over a high-level representation of document context (i.e. ConBART), without explicit need for a planner. In doing so, we showed that this contributes more to performance than the planning phase, but that the combination of the two leads to even further significant improvement. According to our human evaluation, models that are both guided by a plan and have access to document context during generation are better able to handle cases where major transformation is required to complete the simplification. This suggests that our proposed systems are better able to preserve the meaning of the original document while also being less conservative in their editing. In addition, Chapter 7 shows that they are better able to adapt to unseen domains than end-to-end systems.

We also evaluate the efficiency of each document simplification system to determine whether they exhibit any significant limitations with respect to scaling. We find that incorporating a planner yields a relatively small inference time overhead despite needing a second model of similar size to the simplification model. Using a model like ConBART is slower than the optimizations offered by a Longformer as it requires an extra attention computation at each layer throughout the decoding process. However, this scales drastically better than a standard Transformer encoder-decoder over long sequences. Even when using a Longformer to speed things up, the performance of end-to-end models for long documents is greatly outclassed by our proposed systems. A planner can be combined with ConBART with very small performance overhead as the two are able to make use of the same contextual representations, while offering much greater performance across all quality dimensions. Moreover, in Chapter 7 we show that ConBART generating individual sentences at a time actually allows it to better adapt to unseen domains. For example, the length of sentences across domains is likely to remain relatively similar, whereas the length of paragraphs can vary greatly, potentially resulting in a bias towards what is seen in the training data. We see this manifesting when applying Newsela-trained paragraph models to Wikipedia articles (Newsela paragraphs are shorter on average those of Wikipedia) whereby the rate of sentence deletion increases substantially.

Given the existing reliance on references, how can we make the evaluation of simplification systems more scalable to larger texts and new datasets while being robust towards a diversity of simplifications? In Chapter 6 we propose SLE, a learned reference-less metric that is trained to estimate the simplicity of a sentence. We show that using the SLE difference between source and outputs sentences (ΔSLE) as an evaluation metric achieves much higher correlation with human judgements of simplicity than any existing reference-less metric and all but one (LENS) reference-based metric. The fact that no reference simplifications are required means that SLE could be used to evaluate simplification performance on novel and out-of-domain datasets, where no aligned texts exist (which we ourselves showcase in Chapter 7), and is not biased towards whichever operations happen to be represented in the reference texts. Furthermore, because it is trained to preserve document-level accuracy, we believe that SLE is suitable to evaluate document simplification as well.

In Chapter 7 we more specifically consider evaluation of document-level simplification. In

particular, we considered whether it is feasible to apply semantic faithfulness metrics from the summarization literature to the task. Human evaluation suggests that these metrics are insufficient for reliably reporting the performance of document simplification systems. This highlights the need for further exploration into automatic methods of faithfulness evaluation for simplification systems.

Future Directions

Document-Level Simplification Datasets. As has been mentioned throughout this thesis, Newsela, the dataset we primarily focus on for our document simplification studies in Chapters 4 and 5, is not publicly available, with only researchers who obtain explicit approval being authorized to use it. This severely limits its value within the research community as it makes it more difficult to share and reproduce results and leaves those who are unable to obtain a license at a distinct disadvantage with respect to their ability to contribute to the field. Including qualitative evaluation of outputs in published papers is also problematic as written approval is required to include any extracted Newsela text. It is crucial for new open-source corpora to be created specifically for document-level simplification which are of higher quality than the current variations of document-aligned EW/SEW.

More Sophisticated Controllability. Despite the controllable systems proposed in Chapters 3 and 4, which condition generation based on predefined sentence-level operations, showing clear utility in terms of model performance and output diversity, they are still somewhat limited in certain respects. For example, there are some operations (particularly at the document level) which are not explicitly considered, such as sentence re-ordering, fusion, and insertion. Although these could theoretically be carried out implicitly by models like ConBART, which have access to global context during generation, having explicit support for such operations within the planning framework could prove more beneficial. It is also important, however, to consider the necessity of each operation, as a limited set of possible operation types might be entirely sufficient to reliably produce high-quality simplifications.

Another obvious question regarding our operation-based controllability and planning models is the extent to which they allow for multiple operations to be carried out per sentence (e.g. splitting and rephrasing at the same time). In most cases this is fully supported as the simplification models are still trained to implicitly learn transformations from the aligned examples, many of which exhibit some combination of operations despite us only assigning a single operation label to them. The main exceptions here are the examples taken from dedicated splitting corpora in Chapter 3. However, extending the framework to allow for more explicit and fine-grained control of operations could be greatly beneficial. For example, being able to plan/specify operations to be carried out at the level of individual words or phrases within a given sentence.

Automatic Evaluation. Despite proposing a new metric for reference-less simplicity evaluation in Chapter 6, there are still many unsolved challenges concerning the evaluation of simplification systems. Our human correlation study indicates that even the best performing metrics only obtain Pearson coefficients of approximately 0.5 at best, meaning there is still a long way to go if we intend to approach human-level evaluation capability. As illustrated in Chapter 7, there is also currently no obvious way to accurately perform the automatic evaluation of semantic faithfulness/meaning preservation for document simplification, particularly in cases where there are no references to rely on. It also remains unclear how to evaluate the trade-off between simplicity and semantic adequacy without resorting to subjective human judgements of overall quality or again relying on reference simplifications.

Multilinguality. The work presented in this thesis exclusively considers the simplification of English language texts. Creating resources in other languages to allow for the development of non-English language or multilingual simplification systems is crucial if ATS systems are to be deployed globally. Other languages may also call for or prioritize different types of operations than what is generally seen in the case of English, which means that controllable systems could be of potential benefit.

Prior to the publication of this thesis, there has been a number of recent works proposing novel corpora in a range of languages (Ryan et al., 2023; Stodden et al., 2023; Toborek et al., 2023). However, the scale of these are still generally quite limited compared to the available English resources and attempts at developing multilingual simplification models remain rare (Martin et al., 2022; Ryan et al., 2023).

The Utility of Large Language Models. In Chapter 7 we explored the ability of ChatGPT to simplify full Wikipedia articles in a zero-shot fashion, ultimately finding that it had a tendency to over-delete content and not perform as much simplification as dedicated systems. However, in the time since our experiments were carried out, LLMs have seen a huge growth in mainstream popularity and application. More in-depth studies of whether and how these systems can be better leveraged to perform or evaluate text simplification should certainly be carried out in the near future.

Appendix A

Discourse-Splitting Experiments

A.1 Fine-tuning and Multi-Task Learning Experiments

In this work, we experimented with various fine-tuning and multi-task learning regimes in order to see if further performance gains could be met. In the case of multi-task learning, we trained a sequence-to-sequence model to simultaneously learn to perform the C2T, T2S and C2S tasks. The motivation behind this was that there could potentially be useful shared features between the tasks that would help overall learning performance. We used C2T data from D-CCNews, D-MUSS, and D-WikiSplit; and both T2S and C2S data from D-CCNews-S, D-MUSS, and D-Wikisplit.

For our pretraining and fine-tuning experiments we experimented with different dataset combinations and training strategies for both our end-to-end model and our pipeline system. Initially, we tried pretraining first on synthetic data and then fine-tuning on organic data; either as D-MUSS and D-WikiSplit in combination, or one after the other. We also went on to experiment with pretraining on the standard WikiSplit dataset in an attempt to see whether useful features could be learned from training on a generic splitting task. For each of these, we also experimented with freezing/unfreezing different layers in the network (embedding, encoder, and decoder).

As mentioned in Chapter 2, we were unable to observe any improvements over our standard models for any of these experiments. In the case of our pretraining and fine-tuning experiments, the best strategy we found for the end-to-end model was to simply train the BART architecture on standard WikiSplit and further fine-tune on D-MUSS and D-WikiSplit in combination. For the pipeline system, this was to train on the D-CCNews data, fine-tune on D-MUSS, then further fine-tune on D-WikiSplit. The performance of these models are reported in the Chapter 2, but, as can be seen, they failed to outperform other experiments in their respective categories.

A.2 Generation Examples

Model	Text	Good?
	Girls raised by working moms are more likely to be successful in life, while sons raised in homes with working mothers spend their adulthood caring for family members.	
PL_{Synth}	Girls raised by working moms are more likely to be successful in life. **In contrast,** son sons raised in homes with working mothers spend their adulthood caring for family members.	✓
$E2E_{Both}$	Girls raised by working moms are more likely to be successful in life. **By contrast,** sons raised in homes with working mothers spend their adulthood caring for family members.	✓
BL_{Split}	Girls raised by working moms are more likely to be successful in life**. While** sons raised in homes with working mothers spend their adulthood caring for family members.	✗
	These guidelines have been developed with the recognition that Internet technologies are rapidly evolving, and accordingly, guidelines are subject to change.	
PL_{Synth}	Internet technologies are rapidly evolving. **Therefore,** guidelines are subject to change.	✓
$E2E_{Both}$	These guidelines have been developed with the recognition that Internet technologies are rapidly evolving. **Accordingly,** guidelines are subject to change.	✓
BL_{Split}	These guidelines have been developed with the recognition that Internet technologies are rapidly evolving. **Accordingly,** guidelines are subject to change.	✓
	A thing which does not exist in another thing by the self of the latter is not produced from that other thing; for instance, oil is not produced from sand.	
PL_{Synth}	A thing which does not exist in another thing by the self of the latter is not produced from that other thing. For example, **oil is produced** from sand.	✗
$E2E_{Both}$	A thing which does not exist in another thing by the self of the latter is not produced from that other thing**; for instance,** oil is not generated from sand.	✗
BL_{Split}	A thing which does not exist in another thing by the self of the latter is not produced from that other thing. **For instance,** oil isn't produced from sand.	✓
	He was a mighty hunter before the Lord; therefore it is said, "Like Nimrod a mighty hunter before the Lord."	
PL_{Synth}	He was a mighty hunter**. Next,** the Lord; therefore it is said, "Like Nimrod a mightyHunter before the Lord.	✗
$E2E_{Both}$	He was a mighty hunter before the Lord. **Therefore,** it is said, "Like Nimrod a mighty Hunter before theLord."	✓
BL_{Split}	He was a mighty hunter before the Lord **Anthem**. Therefore it is said, like Nimrod, a mighty hunters before the lord Anthem.	✗
	However, Colorado voters denied funding by a 3:2 margin in November 1972 and, three months later, the games were awarded instead to Innsbruck in Austria.	
PL_{Synth}	However, Colorado voters denied funding by a 3:2 margin in November 1972 and, **three months. Eventually,** the games were awarded instead to Innsbruck in Austria.	✗
$E2E_{Both}$	However, Colorado voters denied funding by a 3:2 margin in November 1972. **Three months later,** the games were awarded instead to Innsbruck in Austria.	✓
BL_{Split}	However, Colorado voters denied funding by a 3:2 margin in November 1972. **Three months later,** the games were awarded instead to Innsbruck in Austria.	✓

Table A.1: Example generated texts illustrating the performance of several models (PL_{Synth}, $E2E_{Both}$, BL_{Split}) in various contexts. We try to showcase various ways each of the models can fail.

Appendix B

Plan-Guided Simplification

B.1 Extra Evaluation Results

For clarity, in Table B.1 we provide scores for a wider range of simplification evaluation metrics that were not included in the main body of Chapter 4. These mostly include popular metrics used for sentence simplification that we do not believe adapt as well to the document-level setting, do not provide further insight into system differences, or have not received much support in the literature. Specifically, we include BLEU (Papineni et al., 2002), and full operation scores for both SARI and D-SARI (Sun et al., 2021). For D-SARI, we apply the document-level penalties on top of the base EASSE implementation of SARI.

We can see that the main SARI differences between the context-free planner and Sent-BART is that Sent-BART achieves higher keep, while the planner achieves higher add. This suggests that Sent-BART is likely more conservative in edits. Further, as the context-free planner does not have access to contextual content, it is likely failing to consistently copy/delete the correct parts of the text.

System	BLEU ↑	**D-SARI ↑**	add	keep	delete	**SARI ↑**	add	keep	delete
Input	46.2	8.76	0.0	26.29	0.0	20.52	0.0	61.56	0.0
Reference	100.0	99.98	99.99	99.97	99.99	99.99	100	99.97	99.99
Doc-BART	31.13	30.60	16.54	25.01	50.24	47.07	20.41	55.40	65.40
Sent-BART	70.74	66.27	53.89	71.95	72.95	73.02	55.91	83.66	79.48
PG_{Tag}	48.08	42.96	31.70	44.01	53.17	56.13	35.61	65.61	67.18
$PG_{Tag+Dec}$	48.12	43.31	31.57	44.68	53.69	56.06	35.54	65.54	67.11
PG_{Clf}	70.84	62.97	56.31	65.15	67.47	73.83	57.62	83.56	80.32
PG_{Dyn}	**72.41**	**67.42**	**56.83**	**71.82**	**73.61**	**75.00**	**58.88**	**84.75**	**81.36**
PG_{Oracle}	**78.97**	**77.02**	**63.44**	**83.92**	**83.70**	**80.74**	**65.22**	**89.94**	**87.05**

Table B.1: Extra results for document simplification experiments on Newsela-auto.

Appendix C

Context-Aware Document Simplification Experiments

C.1 Multi-Task Systems

We also experimented with models that are explicitly trained to perform both the planning and simplification tasks using the same network. As high-level plans appear to improve the performance of simplification models, we hypothesize that learning both tasks in tandem could benefit overall performance. The motivation for this approach is to potentially produce a model that is capable of yielding similar or better simplification performance to the pipeline systems but with a more efficient single-model setup.

Specifically, these models were trained to generate the simplified text prefixed by a predicted plan in the form of operation-specific tokens. This was tested with both ConBART (**ConBART$_{\textbf{prefix}}$**) and a document-level Longformer (**LED$_{\textbf{prefix}}$**). In the case of the Longformer we also test a variant that generates the plan tokens as sentence separators (**LED$_{\textbf{sep}}$**). Results are shown in Table C.1.

Unfortunately, from our experiments none of these seemed to result in performance exceeding those of simplification-only models. Improvement could perhaps be reached given the correct tuning of hyperparameters and loss weightings, however we did not have the time nor resources to pursue this further in this study.

System	**BARTScore**				**SMART ↑**			**FKGL ↓**	**SARI ↑**	**Length**	
	Faith. ($s \to h$)	P ↑ ($r \to h$)	R ↑ ($h \to r$)	F1 ↑	P	R	F1			Tok.	Sent.
Input	-0.93	-2.47	-1.99	-2.23	63.2	62.7	62.8	8.44	20.52	866.9	38.6
Reference	-1.99	-0.93	-0.93	-0.93	100	100	100	4.93	99.99	671.5	42.6
LED$_{\text{prefix}}$	-1.83	-1.72	-2.00	-1.86	73.5	67.6	69.8	4.97	63.14	604.0	38.0
LED$_{\text{sep}}$	-1.82	-1.80	-1.88	-1.84	72.6	70.5	71.1	5.06	62.64	640.4	40.2
ConBART$_{\text{prefix}}$	-1.96	-1.62	-1.60	-1.61	80.4	79.6	79.9	4.90	74.31	643.6	41.5

Table C.1: Results of multi-task systems on the Newsela-auto test set.

C.2 Additional Evaluation Results

In Table C.2 we provide additional results for popular automatic evaluation metrics that were not included in the main text. Specifically, we include BLEU (Papineni et al., 2002), and full

operation-specific scores for SARI. In general, the results show similar patterns to those seen in Table 5.2, with $\hat{O} \rightarrow \text{LED}_{\text{para}}$ and $\hat{O} \rightarrow$ ConBART achieving the best results.

Faithfulness BARTScore is included for clarity rather than being a direct estimation of output quality. It shows how semantically similar system outputs are to their inputs, roughly equating to a measurement of conservativity.

System	**BARTScore**	**BLEU** ↑	**SARI** ↑	add	keep	delete
	Faith. $(s \rightarrow h)$					
Input	-0.93	46.2	20.5	0.0	61.6	0.0
Reference	-1.99	100	100	100	100	100
BART_{doc}	-2.48	31.1	47.1	20.4	55.4	65.4
$\text{BART}_{\text{sent}}$	-1.86	70.7	73.0	55.9	83.7	79.5
$\text{BART}_{\text{para}}$	-2.11	68.6	73.7	57.8	82.6	80.8
LED_{doc}	-1.90	63.7	68.7	52.2	78.2	75.7
LED_{para}	-1.86	74.5*	76.9*	64.3*	85.0	81.5
ConBART	-1.89	73.7	75.8	61.4	84.9	81.2
PG_{Dyn} (2023c)	-1.91	72.4	75.0	58.9	84.8	81.4
$\hat{O} \rightarrow$ ConBART	-1.92	76.0*	78.3*	64.6*	86.8*	**83.4**
$\hat{O} \rightarrow \text{BART}_{\text{para}}$	-2.05	71.3	74.9	58.5	84.7	81.4
$\hat{O} \rightarrow \text{LED}_{\text{para}}$	-1.87	**76.8***	**78.5***	**65.1***	**87.3***	83.0
$\text{PG}_{\text{Oracle}}$ (2023c)	-1.93	78.9*	80.7*	65.2*	89.9*	87.1*
$O \rightarrow$ ConBART	-1.93	**82.6***	**83.8***	**70.8***	**91.7***	**88.7***
$O \rightarrow \text{BART}_{\text{para}}$	-2.09	76.1*	79.7*	64.1*	88.7*	86.3*
$O \rightarrow \text{LED}_{\text{para}}$	-1.90	81.4*	82.3*	69.6*	90.6*	86.7*

Table C.2: Extra automatic evaluation results on Newsela-auto. For BARTScore, s is the source text and h is the hypothesis. Scores significantly higher than PG_{Dyn} are denoted with $*$ ($p < 0.005$). Significance was determined with Student's t-tests.

C.3 Example Simplifications

Figure C.1 shows several example simplifications by the $\hat{O} \rightarrow \text{LED}_{\text{para}}$ system on full documents. Due to licensing constraints imposed by Newsela, we use out-of-domain documents from the WikiLarge dataset here. As these are Wikipedia articles they are quite different in tone than the Newsela articles as well as being much shorter in length. Regardless, we still believe this provides clarity on the types of editing performed by the model.

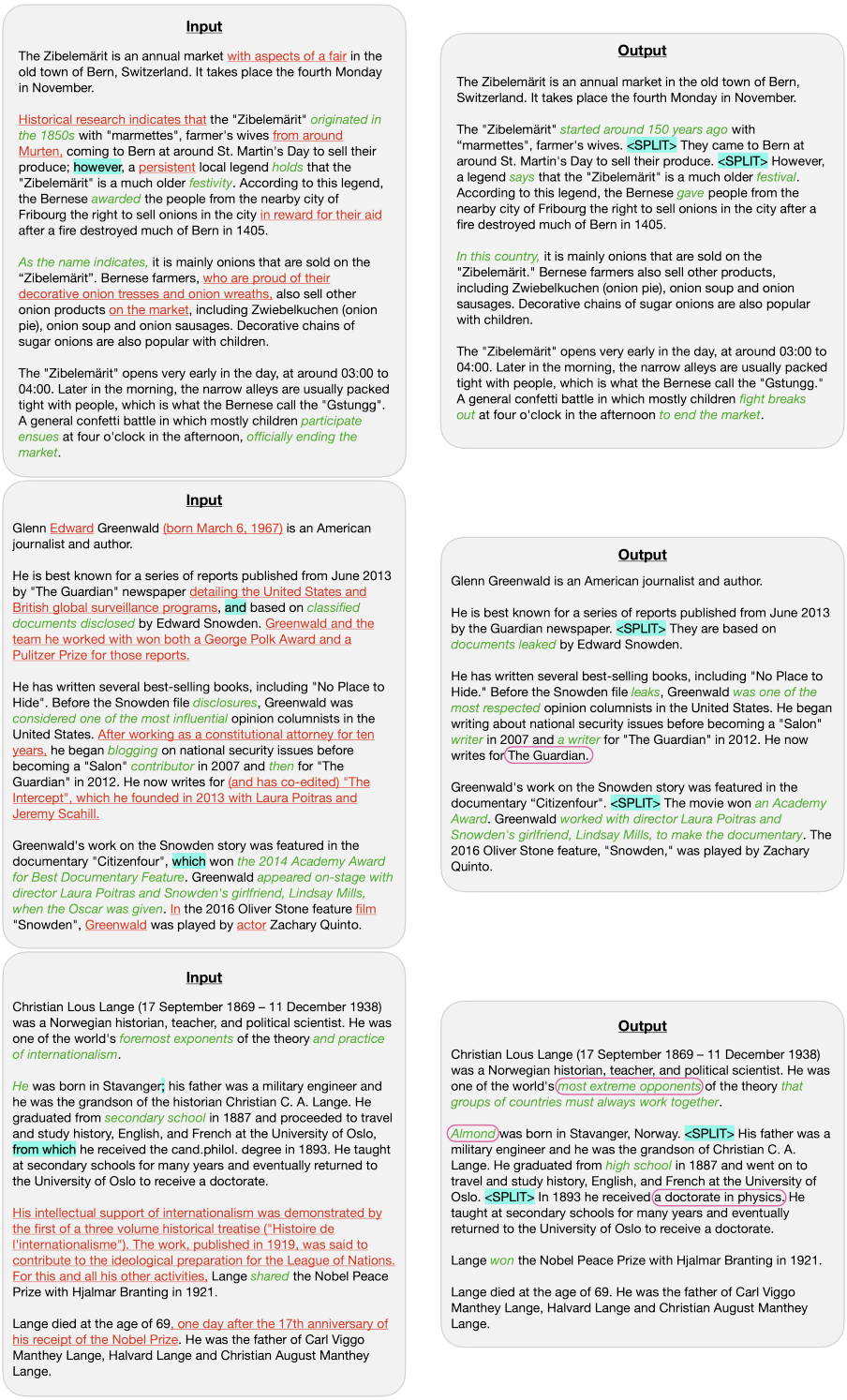


Figure C.1: Example simplification outputs for $\hat{O} \rightarrow \text{LED}_{\text{para}}$, illustrating both strong and poor performances (a target reading-level of 3 was used for all examples). Input documents are taken from WikiLarge due to licensing constraints around sharing Newsela content. Deletions are underlined and in red; rephrasings are *italicized and in green*; splitting points are highlighted in cyan; and factual errors are circled.

Appendix D

Out-of-Domain Document Simplification

D.1 ChatGPT Prompts

Table D.1 shows some of the ChatGPT prompts tested at the beginning of our study and their response simplifications for a specific Wikipedia article. All prompts lead to relatively short response sequences (even when explicitly instructing it within the prompt to not to delete too much) that do not contain obviously simpler language. This makes us believe it is rather performing summarization.

		Tokens	SLE
INPUT	WWE SmackDown vs. Raw 2007 is a professional wrestling video game developed by Yukeś and published by THQ. It is based on the professional wrestling promotion World Wrestling Entertainment (WWE) and is the eighth installment of the "WWE SmackDown!" series and is the successor to 2005ś "WWE SmackDown! vs. Raw 2006" and was succeeded by "WWE SmackDown vs. Raw 2008" in 2007. On 31 March 2006, THQ, Yukeś and WWE officially confirmed the game. The game was released worldwide in November 2006 for PlayStation 2 and Xbox 360 and a month later for PlayStation Portable. The game introduced a few key new features including an analog control system, interactive hotspots, and fighting within the arena crowd. The game also included several improvements on the previous gameś existing match types and modes. The Xbox 360 version was the first game in the "SmackDown" series to be published on a seventh generation video game console and the first game in the series to be released outside the PlayStation consoles ...	1,792	1.09
Prompt	**Result**		
Rewrite the following document, making it easier to understand: [document]	WWE SmackDown vs. Raw 2007 is a video game based on the professional wrestling promotion World Wrestling Entertainment (WWE). The game was developed by Yuke's and published by THQ. It was released in November 2006 for PlayStation 2 and Xbox 360, and a month later for PlayStation Portable. The game introduced new features, such as an analog control system, interactive hotspots, and fighting within the arena crowd. It also included improvements on existing match types and modes. The game's graphics, audio system, and Create-A-Superstar and Create-An-Entrance modes were also improved. The game is the eighth installment of the "WWE SmackDown!" series and is the successor to 2005's "WWE SmackDown! vs. Raw 2006". It was succeeded by "WWE SmackDown vs. Raw 2008" in 2007. ...	219	1.38
Simplify the following document: [document]	This is a description of the video game "WWE SmackDown vs. Raw 2007," developed by Yuke's and published by THQ. The game is based on the professional wrestling promotion World Wrestling Entertainment (WWE) and is the eighth installment in the "WWE SmackDown!" series. It was released in November 2006 for PlayStation 2 and Xbox 360, and a month later for PlayStation Portable. The game features a new analog control system, interactive hotspots, and fighting within the arena crowd. The Xbox 360 version was the first in the series to be released outside the PlayStation consoles. The game's audio system has been improved with new sound effects, and the graphics have also been improved. The game includes improvements on existing match types and modes, but no new match types have been added. ...	243	1.18
Simplify the following document, without deleting too much: [document]	WWE SmackDown vs. Raw 2007 is a video game by Yuke's and THQ based on professional wrestling promotion World Wrestling Entertainment (WWE). It is the eighth installment of the "WWE SmackDown!" series and was released in November 2006 for PlayStation 2 and Xbox 360 and a month later for PlayStation Portable. The game features new analog control, interactive hotspots, fighting within the arena crowd, and improvements to existing match types and modes. The Xbox 360 version is the first in the "SmackDown" series to be published on a seventh-generation video game console. The graphics, audio, and Create-A-Superstar and Create-An-Entrance modes have also been improved. ...	199	1.32

Table D.1: Example ChatGPT prompts and their responses. Rendered texts have been cut short to fit on the page. Token lengths displayed are for the full responses.

Bibliography

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Sweta Agrawal, Weijia Xu, and Marine Carpuat. 2021. A non-autoregressive edit-based approach to controllable text simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3757–3769, Online. Association for Computational Linguistics.

Roee Aharoni and Yoav Goldberg. 2018. Split and rephrase: Better evaluation and stronger baselines. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 719–724, Melbourne, Australia. Association for Computational Linguistics.

Sandra Aluísio and Caroline Gasperin. 2010. Fostering digital inclusion and accessibility: The PorSimples project for simplification of Portuguese texts. In *Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas*, pages 46–53, Los Angeles, California. Association for Computational Linguistics.

Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295–305, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020a. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online. Association for Computational Linguistics.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019a. EASSE: Easier automatic sentence simplification evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2019b. Cross-sentence transformations in text simplification. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 181–184, Florence, Italy. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020b. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (un)suitability of automatic evaluation metrics for text simplification. *Computational Linguistics*, 47(4):861–889.

Bharat Ram Ambati, Siva Reddy, and Mark Steedman. 2016. Assessing relative sentence complexity using an incremental ccg parser. In *Proceedings of NAACL-HLT*, pages 1051–1057.

Reinald Kim Amplayo, Peter J. Liu, Yao Zhao, and Shashi Narayan. 2022. Smart: Sentences as basic units for text evaluation.

Mandya Angrosh, Tadashi Nomoto, and Advaith Siddharthan. 2014. Lexico-syntactic text simplification and compression with typed dependencies. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1996–2006, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Dennis Aumiller and Michael Gertz. 2022. Klexikon: A German dataset for joint summarization and simplification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2693–2701, Marseille, France. European Language Resources Association.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 25–32.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501, Portland, Oregon, USA. Association for Computational Linguistics.

Sofia Blinova, Xinyu Zhou, Martin Jaggi, Carsten Eickhoff, and Seyed Ali Bahrainian. 2023. SIMSUM: Document-level text simplification via simultaneous summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9927–9944, Toronto, Canada. Association for Computational Linguistics.

Johan Bos. 2015. Open-domain semantic parsing with boxer. In *Proceedings of the 20th nordic conference of computational linguistics (NODALIDA 2015)*, pages 301–304.

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. 2018. Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 Conference*

on Empirical Methods in Natural Language Processing, pages 732–737, Brussels, Belgium. Association for Computational Linguistics.

Dominique Brunato, Andrea Cimino, Felice Dell'Orletta, and Giulia Venturi. 2016. PaCCSS-IT: A parallel corpus of complex-simple sentences for automatic text simplification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 351–361, Austin, Texas. Association for Computational Linguistics.

Dominique Brunato, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni. 2015. Design and annotation of the first Italian corpus for text simplification. In *Proceedings of the 9th Linguistic Annotation Workshop*, pages 31–41, Denver, Colorado, USA. Association for Computational Linguistics.

Rémi Cardon and Natalia Grabar. 2020. French biomedical text simplification: When small and precise helps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 710–716, Barcelona, Spain (Online). International Committee on Computational Linguistics.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10. Association for the Advancement of Artificial Intelligence.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2020. MOCHA: A dataset for training and evaluating generative reading comprehension metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6521–6532, Online. Association for Computational Linguistics.

Prafulla Kumar Choubey, Aaron Lee, Ruihong Huang, and Lu Wang. 2020. Discourse as a function of event: Profiling discourse structure in news articles around the main event. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5374–5386, Online. Association for Computational Linguistics.

Trevor Anthony Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34:637–674.

William Coster and David Kauchak. 2011. Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, Portland, Oregon, USA. Association for Computational Linguistics.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2021. Discourse-based sentence splitting. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 261–273, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2022. Controllable sentence simplification via operation classification. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2091–2103, Seattle, United States. Association for Computational Linguistics.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2023a. Context-aware document simplification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13190–13206, Toronto, Canada. Association for Computational Linguistics.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2023b. Document-level planning for text simplification. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 993–1006, Dubrovnik, Croatia. Association for Computational Linguistics.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2023c. Document-level planning for text simplification. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 993–1006, Dubrovnik, Croatia. Association for Computational Linguistics.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2023d. On semantic faithfulness and out-of-domain performance in document simplification.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2023e. Simplicity level estimate (sle): A learned reference-less metric for sentence simplification.

Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Prroceedings of the SIGIR workshop on accessible search systems*, pages 19–26. ACM; New York.

Mohammad Dehghan, Dhruv Kumar, and Lukasz Golab. 2022. GRS: Combining generation and revision in unsupervised sentence simplification. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 949–960, Dublin, Ireland. Association for Computational Linguistics.

Ashwin Devaraj, Iain Marshall, Byron Wallace, and Junyi Jessy Li. 2021. Paragraph-level simplification of medical texts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4972–4984, Online. Association for Computational Linguistics.

Ashwin Devaraj, William Sheffield, Byron Wallace, and Junyi Jessy Li. 2022. Evaluating factuality in text simplification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7331–7345, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Anna Dmitrieva and Jörg Tiedemann. 2021. Creating an aligned Russian text simplification dataset from language learner data. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 73–79, Kiyv, Ukraine. Association for Computational Linguistics.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Juliette Faille, Albert Gatt, and Claire Gardent. 2021. Entity-based semantic adequacy for data-to-text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1530–1540, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Dan Feblowitz and David Kauchak. 2013. Sentence simplification as tree transduction. In *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 1–10, Sofia, Bulgaria. Association for Computational Linguistics.

Daniel Ferrés, Montserrat Marimon, Horacio Saggion, and Ahmed AbuRa'ed. 2016. Yats: yet another text simplifier. In *Natural Language Processing and Information Systems: 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK, June 22-24, 2016, Proceedings 21*, pages 335–342. Springer.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 25–32, Salt Fork, Ohio, USA. Association for Computational Linguistics.

Núria Gala, Anaïs Tack, Ludivine Javourey-Drevet, Thomas François, and Johannes C. Ziegler. 2020. Alector: A parallel corpus of simplified French texts with alignments of misreadings by poor and dyslexic readers. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1353–1361, Marseille, France. European Language Resources Association.

Cristina Garbacea, Mengtian Guo, Samuel Carton, and Qiaozhu Mei. 2021. Explainable prediction of text complexity: The missing preliminaries for text simplification. 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)*, pages 1086–1097, Online. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating Training Corpora for NLG Micro-Planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? 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)*, pages 63–68, Beijing, China. Association for Computational Linguistics.

Sian Gooding. 2022. On the ethical considerations of text simplification. In *Ninth Workshop on Speech and Language Processing for Assistive Technologies (SLPAT-2022)*, pages 50–57, Dublin, Ireland. Association for Computational Linguistics.

Sian Gooding, Ekaterina Kochmar, Seid Muhie Yimam, and Chris Biemann. 2021. Word complexity is in the eye of the beholder. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4439–4449, Online. Association for Computational Linguistics.

Natalia Grabar and Rémi Cardon. 2018. CLEAR – simple corpus for medical French. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, pages 3–9, Tilburg, the Netherlands. Association for Computational Linguistics.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Dynamic multi-level multi-task learning for sentence simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 462–476, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, pages 218–223.

Devamanyu Hazarika, Mahdi Namazifar, and Dilek Hakkani-Tür. 2022. Zero-shot controlled generation with encoder-decoder transformers.

Michael Heilman and Noah A Smith. 2010. Extracting simplified statements for factual question generation. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 11–20.

Ralf Herbrich. 2000. Large margin rank boundaries for ordinal regression. *Advances in large margin classifiers*, pages 115–132.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1138, Vancouver, Canada. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

David M. Howcroft and Vera Demberg. 2017. Psycholinguistic models of sentence processing improve sentence readability ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 958–968, Valencia, Spain. Association for Computational Linguistics.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960, Online. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546, Sofia, Bulgaria. Association for Computational Linguistics.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Sigrid Klerke and Anders Søgaard. 2012. DSim, a Danish parallel corpus for text simplification. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 4015–4018, Istanbul, Turkey. European Language Resources Association (ELRA).

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 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*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Fajri Koto, Timothy Baldwin, and Jey Han Lau. 2022. Ffci: A framework for interpretable automatic evaluation of summarization. *Journal of Artificial Intelligence Research*, 73:1553–1607.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928, Online. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A. Hearst. 2021. Keep it simple: Unsupervised simplification of multi-paragraph text. 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)*, pages 6365–6378, Online. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Revisiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Justin Lee and Sowmya Vajjala. 2022. A neural pairwise ranking model for readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3802–3813, Dublin, Ireland. Association for Computational Linguistics.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Junyi Li, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021. Knowledge-based review generation by coherence enhanced text planning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 183–192, New York, NY, USA. Association for Computing Machinery.

Junyi Jessy Li and Ani Nenkova. 2015. Detecting content-heavy sentences: A cross-language case study. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1271–1281, Lisbon, Portugal. Association for Computational Linguistics.

Zhe Lin, Yitao Cai, and Xiaojun Wan. 2021. Towards document-level paraphrase generation with sentence rewriting and reordering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1033–1044, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 20:151–184.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Yuan Ma, Sandaru Seneviratne, and Elena Daskalaki. 2022. Improving text simplification with factuality error detection. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 173–178, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2021. Controllable text simplification with explicit paraphrasing. In *Proceedings of the 2021 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3536–3553, Online. Association for Computational Linguistics.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2022. Lens: A learnable evaluation metric for text simplification.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

William C Mann and Sandra A Thompson. 1987. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2021. Muss: Multilingual unsupervised sentence simplification by mining paraphrases. *arXiv preprint arXiv:2005.00352*.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2022. MUSS: Multilingual unsupervised sentence simplification by mining paraphrases. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1651–1664, Marseille, France. European Language Resources Association.

Louis Martin, Samuel Humeau, Pierre-Emmanuel Mazaré, Éric de La Clergerie, Antoine Bordes, and Benoît Sagot. 2018. Reference-less quality estimation of text simplification systems. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, pages 29–38, Tilburg, the Netherlands. Association for Computational Linguistics.

Takumi Maruyama and Kazuhide Yamamoto. 2018. Simplified corpus with core vocabulary. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Kshitij Mishra, Ankush Soni, Rahul Sharma, and Dipti Sharma. 2014. Exploring the effects of sentence simplification on Hindi to English machine translation system. In *Proceedings of the Workshop on Automatic Text Simplification - Methods and Applications in the Multilingual Society (ATS-MA 2014)*, pages 21–29, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 788–796, Beijing, China. Coling 2010 Organizing Committee.

Tsendsuren Munkhdalai and Hong Yu. 2017. Neural semantic encoders. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 397–407, Valencia, Spain. Association for Computational Linguistics.

Sebastian Nagel. 2016. Cc-news.

Akifumi Nakamachi, Tomoyuki Kajiwara, and Yuki Arase. 2020. Text simplification with reinforcement learning using supervised rewards on grammaticality, meaning preservation, and simplicity. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 153–159, Suzhou, China. Association for Computational Linguistics.

Shashi Narayan and Claire Gardent. 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)*, pages 435–445, Baltimore, Maryland. Association for Computational Linguistics.

Shashi Narayan and Claire Gardent. 2016. Unsupervised sentence simplification using deep semantics. In *Proceedings of the 9th International Natural Language Generation conference*, pages 111–120, Edinburgh, UK. Association for Computational Linguistics.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 606–616, Copenhagen, Denmark. Association for Computational Linguistics.

Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Conditional generation with a question-answering blueprint.

Christina Niklaus. 2022. *From Complex Sentences to a Formal Semantic Representation using Syntactic Text Simplification and Open Information Extraction.* Ph.D. thesis, Universität Passau.

Christina Niklaus, Bernhard Bermeitinger, Siegfried Handschuh, and André Freitas. 2016. A sentence simplification system for improving relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 170–174, Osaka, Japan. The COLING 2016 Organizing Committee.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019a. DisSim: A discourse-aware syntactic text simplification framework for English and German. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 504–507, Tokyo, Japan. Association for Computational Linguistics.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019b. Transforming complex sentences into a semantic hierarchy. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3415–3427, Florence, Italy. Association for Computational Linguistics.

Christina Niklaus, André Freitas, and Siegfried Handschuh. 2019c. MinWikiSplit: A sentence splitting corpus with minimal propositions. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 118–123, Tokyo, Japan. Association for Computational Linguistics.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266, Florence, Italy. Association for Computational Linguistics.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2023. Deep learning approaches to lexical simplification: A survey. *arXiv preprint arXiv:2305.12000*.

Kostiantyn Omelianchuk, Vipul Raheja, and Oleksandr Skurzhanskyi. 2021. Text Simplification by Tagging. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–25, Online. Association for Computational Linguistics.

Gustavo H. Paetzold and Lucia Specia. 2013. Text simplification as tree transduction. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*.

Gustavo H Paetzold and Lucia Specia. 2017. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60:549–593.

Gustavo Henrique Paetzold. 2016. *Lexical simplification for non-native english speakers*. Ph.D. thesis, University of Sheffield.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ellie Pavlick and Chris Callison-Burch. 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)*, pages 143–148, Berlin, Germany. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Sarah Elizabeth Petersen and Mari Ostendorf. 2007. *Natural Language Processing Tools for Reading Level Assessment and Text Simplication for Bilingual Education*. Citeseer.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pretrained encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8649–8656.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Luz Rello, Ricardo Baeza-Yates, Stefan Bott, and Horacio Saggion. 2013. Simplify or help? text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, pages 1–10.

Annette Rios, Nicolas Spring, Tannon Kew, Marek Kostrzewa, Andreas Säuberli, Mathias Müller, and Sarah Ebling. 2021. A new dataset and efficient baselines for document-level text simplification in German. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 152–161, Online and in Dominican Republic. Association for Computational Linguistics.

Michael Ryan, Tarek Naous, and Wei Xu. 2023. Revisiting non-English text simplification: A unified multilingual benchmark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4898–4927, Toronto, Canada. Association for Computational Linguistics.

Horacio Saggion, Sanja Štajner, Stefan Bott, Simon Mille, Luz Rello, and Biljana Drndarevic. 2015. Making it simplext: Implementation and evaluation of a text simplification system for spanish. *ACM Trans. Access. Comput.*, 6(4).

Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2023. Findings of the tsar-2022 shared task on multilingual lexical simplification.

Andrey Sakhovskiy, Alexandra Izhevskaya, Alena Pestova, Elena Tutubalina, Valentin Malykh, Ivana Smurov, and Ekaterina Artemova. 2021. RuSimpleSentEval-2021 shared task: evaluating sentence simplification for russian. In *Proceedings of the International Conference "Dialogue*, pages 607–617.

Andreas Säuberli, Sarah Ebling, and Martin Volk. 2020. Benchmarking data-driven automatic text simplification for German. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 41–48, Marseille, France. European Language Resources Association.

Carolina Scarton, Pranava Madhyastha, and Lucia Specia. 2020. Deciding when, how and for whom to simplify. © 2020 The Author(s) and IOS Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial Licence (http://creativecommons.org/licenses/by-nc/4.0/).

Carolina Scarton and Lucia Specia. 2018. Learning simplifications for specific target audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 712–718, Melbourne, Australia. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Max Schwarzer and David Kauchak. 2018. Human evaluation for text simplification: The simplicity-adequacy tradeoff. In *SoCal NLP Symposium*.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021a. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Scialom, Louis Martin, Jacopo Staiano, Éric Villemonte de la Clergerie, and Benoît Sagot. 2021b. Rethinking automatic evaluation in sentence simplification. *CoRR*, abs/2104.07560.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Advaith Siddharthan. 2002. An architecture for a text simplification system. In *Language Engineering Conference, 2002. Proceedings*, pages 64–71.

Advaith Siddharthan. 2003. Preserving discourse structure when simplifying text. In *Proceedings of the 9th European Workshop on Natural Language Generation (ENLG-2003) at EACL 2003*, Budapest, Hungary. Association for Computational Linguistics.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.

Advaith Siddharthan and Angrosh Mandya. 2014. Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 722–731, Gothenburg, Sweden. Association for Computational Linguistics.

Aviv Slobodkin, Paul Roit, Eran Hirsch, Ori Ernst, and Ido Dagan. 2022. Controlled text reduction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5699–5715.

Eric Michael Smith, Diana Gonzalez-Rico, Emily Dinan, and Y-Lan Boureau. 2020. Controlling style in generated dialogue. *CoRR*, abs/2009.10855.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Lucia Specia. 2010. Translating from complex to simplified sentences. In *Computational Processing of the Portuguese Language: 9th International Conference, PROPOR 2010, Porto Alegre, RS, Brazil, April 27-30, 2010. Proceedings 9*, pages 30–39. Springer.

Neha Srikanth and Junyi Jessy Li. 2021. Elaborative simplification: Content addition and explanation generation in text simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5123–5137, Online. Association for Computational Linguistics.

Sanja Štajner and Goran Glavaš. 2017. Leveraging event-based semantics for automated text simplification. *Expert systems with applications*, 82:383–395.

Sanja Štajner, Ruslan Mitkov, and Horacio Saggion. 2014. One step closer to automatic evaluation of text simplification systems. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 1–10, Gothenburg, Sweden. Association for Computational Linguistics.

Sanja Štajner and Maja Popovic. 2016. Can text simplification help machine translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 230–242.

Sanja Štajner, Maja Popovic, Horacio Saggion, Lucia Specia, and Mark Fishel. 2016. Shared task on quality assessment for text simplification. *Training*, 218(95):192.

Regina Stodden, Omar Momen, and Laura Kallmeyer. 2023. DEplain: A German parallel corpus with intralingual translations into plain language for sentence and document simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16441–16463, Toronto, Canada. Association for Computational Linguistics.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018a. BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744, Brussels, Belgium. Association for Computational Linguistics.

Elior Sulem, Omri Abend, and Ari Rappoport. 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)*, pages 685–696, New Orleans, Louisiana. Association for Computational Linguistics.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018c. Simple and effective text simplification using semantic and neural methods. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 162–173, Melbourne, Australia. Association for Computational Linguistics.

Hong Sun and Ming Zhou. 2012. Joint learning of a dual SMT system for paraphrase generation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–42, Jeju Island, Korea. Association for Computational Linguistics.

Renliang Sun, Hanqi Jin, and Xiaojun Wan. 2021. Document-level text simplification: Dataset, criteria and baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7997–8013, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Renliang Sun, Zhe Lin, and Xiaojun Wan. 2020. On the helpfulness of document context to sentence simplification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1411–1423, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068, Florence, Italy. Association for Computational Linguistics.

Craig Thomson and Ehud Reiter. 2020. A gold standard methodology for evaluating accuracy in data-to-text systems. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 158–168, Dublin, Ireland. Association for Computational Linguistics.

Vanessa Toborek, Moritz Busch, Malte Bošert, Christian Bauckhage, and Pascal Welke. 2023. A new aligned simple German corpus. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11393–11412, Toronto, Canada. Association for Computational Linguistics.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.

Sowmya Vajjala and Detmar Meurers. 2014. Assessing the relative reading level of sentence pairs for text simplification. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 288–297, Gothenburg, Sweden. Association for Computational Linguistics.

Sowmya Vajjala and Detmar Meurers. 2016. Readability-based sentence ranking for evaluating text simplification. *CoRR*, abs/1603.06009.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Tu Vu, Baotian Hu, Tsendsuren Munkhdalai, and Hong Yu. 2018. Sentence simplification with memory-augmented neural networks. In *Proceedings of the 2018 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 79–85, New Orleans, Louisiana. Association for Computational Linguistics.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. Facilita: Reading assistance for low-literacy readers. In *Proceedings of the 27th ACM International Conference on Design of Communication*, SIGDOC '09, page 29–36, New York, NY, USA. Association for Computing Machinery.

Bonnie Webber. 2023. Advancing discourse-based sentence splitting. GardentFest Workshop Invited Talk.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pages 5–32.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Kristian Woodsend and Mirella Lapata. 2011a. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Kristian Woodsend and Mirella Lapata. 2011b. Wikisimple: Automatic simplification of wikipedia articles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 927–932.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530, Toulouse, France. Association for Computational Linguistics.

Kenji Yamada and Kevin Knight. 2002. A decoder for syntax-based statistical MT. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 303–310, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Daiki Yanamoto, Tomoki Ikawa, Tomoyuki Kajiwara, Takashi Ninomiya, Satoru Uchida, and Yuki Arase. 2022. Controllable text simplification with deep reinforcement learning. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 398–404, Online only. Association for Computational Linguistics.

Taha Yasseri, András Kornai, and János Kertész. 2012. A Practical Approach to Language Complexity: A Wikipedia Case Study.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Bohan Zhang, Prafulla Kumar Choubey, and Ruihong Huang. 2022. Predicting sentence deletions for text simplification using a functional discourse structure. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 255–261, Dublin, Ireland. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173, Brussels, Belgium. Association for Computational Linguistics.

Yang Zhong, Chao Jiang, Wei Xu, and Junyi Jessy Li. 2020. Discourse level factors for sentence deletion in text simplification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9709–9716.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.