AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

École doctorale IAEM Lorraine

Text Generation with and without Retrieval

THÈSE

présentée et soutenue publiquement le 30 Oct 2021

pour l'obtention du

Doctorat de l'Université de Lorraine
(mention informatique)

par

Angela Fan

Composition du jury

Directeur de thèse:	Claire Gardent
Co-directeurs de thèse:	Antoine Bordes Chloé Braud
Rapporteurs :	Benoit Crabbé (President) André Martins
Examinateur :	Natalie Schluter

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

Text Generation with and without Retrieval

Angela Fan

Text Generation
with and without Retrieval

Text Generation with and without Retrieval

Thèse

Angela Fan

August 2021

présentée et soutenue publiquement
pour l'obtention du

Doctorat de l'Université de Lorraine
(mention informatique)

1st Reviewer	Benoit Crabbé
2nd Reviewer	André Martins

COLOPHON

This thesis was typeset using LaTeX and the `memoir` documentclass. It is based on Aaron Turon's thesis *Understanding and expressing scalable concurrency*[1], itself a mixture of `classicthesis`[2] by André Miede and `tufte-latex`[3], based on Edward Tufte's *Beautiful Evidence*.

The bibliography was processed by natbib. All graphics and plots are made with PGF/Ti*k*Z.

The body text is set 10/14pt (long primer) on a 26pc measure. The margin text is set 8/9pt (brevier) on a 12pc measure. Matthew Carter's Charter acts as both the text and display typeface. Monospaced text uses Jim Lyles's `Bitstream Vera Mono` ("Bera Mono").

[1] `https://people.mpi-sws.org/~turon/turon-thesis.pdf`

[2] `https://bitbucket.org/amiede/classicthesis/`

[3] `https://github.com/Tufte-LaTeX/tufte-latex`

I don’t feel that it is necessary to know exactly what I am.
The main interest in life and work is to become
someone else that you were not in the beginning.
—Michel Foucault

Abstract

Text generation is a research field with the goal of automatically creating natural language based on some input information. These systems have broad applicability — from simplifying sentences so they are easier to read and understand to the creation of dialogue systems that can chat with users or help them accomplish a desired task. Text generation methods rely on various different inputs to be able to produce natural language — from texts and documents to tables, knowledge graphs, and meaning representations. Not only is understanding the varied forms of input structure challenging, but also generation systems must be able to produce fluent, accurate natural language even in cases of little training data, which particularly arises in the multilingual case. We focus on two major challenges in this thesis: first, the creation and improvement of text generation systems that can operate in languages beyond English. Second, we work to incorporate knowledge into text generation, enabling systems that can consult external sources such as Wikipedia, databases, or the open web for information.

We begin with generation based on natural language texts for the task of sentence simplifcation, and then discuss generation based on abstract meaning representations. We then focus on generation tasks that require retrieval from large, external sources of information, such as Wikipedia or the open web. These tasks encompass generation from structured inputs like knowledge graphs, as well as generation from free-text inputs. We end with a conclusion of our main contributions as well as a discussion of open challenges.

Résumé

La génération de texte est un domaine de recherche dont l'objectif est de créer automatiquement un langage naturel à partir de certaines informations d'entrée. Ces systèmes ont un large champ d'application, allant de la simplification des phrases pour qu'elles soient plus faciles à lire et à comprendre à la création de systèmes de dialogue qui peuvent discuter avec des utilisateurs ou les aider à accomplir une tâche donnée. Les méthodes de génération de texte s'appuient sur différentes entrées pour pouvoir produire du langage naturel. Celles-ci peuvent être des textes ou documents, voire des tableaux, des graphiques de connaissances ou des représentations sémantique abstraite. Non seulement la compréhension des diverses formes de structure d'entrée est un défi, mais les systèmes de génération doivent également être capables de produire un langage naturel fluide et précis même en cas de données d'entraînement peu nombreuses, ce qui se produit particulièrement dans le cas du multilinguisme. Nous nous concentrons sur deux défis majeurs dans cette thèse: premièrement, la création et l'amélioration des systèmes de génération de texte qui peuvent fonctionner dans des langues autres que l'anglais. Deuxièmement, nous travaillons à l'incorporation de connaissances dans la génération de textes, permettant ainsi aux systèmes de consulter des sources externes telles que Wikipedia, des bases de données, ou le web ouvert pour obtenir des informations.

Nous commençons par la génération basée sur des textes en langage naturel pour la tâche de simplification des phrases, puis nous abordons la génération basée sur des représentations de sens abstraites. Nous nous concentrons ensuite sur les tâches de génération qui nécessitent l'extraction de grandes sources d'information externes, telles que Wikipédia ou le Web ouvert. Ces tâches englobent la génération à partir d'entrées structurées comme les graphes de connaissances, ainsi que la génération à partir d'entrées en texte libre. Nous terminons par une conclusion de nos principales contributions ainsi que par une discussion sur les problèmes encore ouverts.

Sommaire

INTRODUCTION

Tous les jours, nous écrivons — qu'il s'agisse d'envoyer un texte rapide à votre mère ou de rédiger un article scientifique tel que cette thèse. L'écriture que nous faisons va souvent de pair avec une assistance automatisée. Par exemple, les logiciels modernes de messagerie instantanée suggèrent souvent le mot à écrire ensuite, les courriers électroniques peuvent être lancés à l'aide d'un autocomposeur et les rédactions sont améliorées grâce à des suggestions d'édition faites par des machines. Ces technologies sont le fruit d'années de recherche sur la "génération de texte", un domaine du traitement du langage naturel dont l'objectif est de produire automatiquement un langage naturel fluide et lisible par l'homme. À petite échelle, les systèmes de génération de texte peuvent générer des mots ou des phrases isolés, mais leurs applications vont bien au-delà. Par exemple, les systèmes de résumé, de dialogue et même la rédaction d'articles entiers de Wikipédia reposent sur la technologie fondamentale de génération de texte.

La production d'un langage naturel fluide, précis et utile est confrontée à de nombreux défis. Les progrès récents en matière de génération de texte, qui s'appuient principalement sur l'apprentissage d'architectures de réseaux neuronaux sur de grands ensembles de données, ont considérablement amélioré la lisibilité de surface du texte généré par la machine. Cependant, les systèmes actuels nécessitent des améliorations sur de nombreux axes, notamment la génération de textes autres que l'anglais et la rédaction de textes de plus en plus longs. Bien que le domaine ait connu des progrès rapides, la recherche s'est surtout concentrée sur la langue anglaise, où des ensembles de données d'entraînement et d'évaluation à grande échelle pour diverses tâches sont facilement disponibles. Néanmoins, les applications allant de l'autocorrection à l'autocomposition de texte devraient être disponibles universellement. Après tout, la majorité de la population mondiale n'écrit pas en anglais. Dans ce travail, nous créons des systèmes de génération de texte pour diverses tâches avec la capacité d'incorporer des langues autres que l'anglais, soit sous forme d'algorithmes qui s'étendent facilement à de nouvelles langues, soit sous forme de modèles multilingues englobant jusqu'à 20 langues dans un seul modèle.

Au-delà de notre travail sur la génération de textes multilingues, nous nous concentrons sur un élément essentiel des systèmes de génération: la connaissance du texte. Pour bien écrire, il faut d'abord savoir quoi écrire. Ce concept de connaissance est incroyablement important dans les systèmes de génération de texte. Par exemple, la rédaction automatique d'un article complet sur Wikipédia nécessite une recherche approfondie sur le sujet de l'article. L'instinct de recherche est souvent intuitif — il y a quelques décennies, les gens se seraient rendus dans une bibliothèque, remplacés aujourd'hui par les informations disponibles sur le World Wide Web. Cependant, pour les systèmes automatisés, la question n'est pas seulement de savoir quelles connaissances utiliser pour générer du texte, mais aussi comment *retrouver* ces connaissances et les utiliser au mieux pour atteindre l'objectif de communication visé.

Nous relevons le défi de la génération de texte basée sur la recherche. Nous présentons plusieurs techniques d'identification des connaissances pertinentes à différentes échelles : de la connaissance locale disponible dans un paragraphe à l'identification de l'aiguille dans la botte de foin à l'échelle du web complet, en passant par le passage au crible de Wikipedia. Nous décrivons des architectures de réseaux neuronaux capables d'effectuer efficacement des recherches à grande échelle, en utilisant des mécanismes de précalcul et de mise en cache. Au-delà de la manière d'extraire les connaissances, nous étudions plus en détail la *forme* que les connaissances devraient prendre — du langage naturel comme les articles de Wikipedia ou le texte sur le web aux entrées structurées sous forme de graphes de connaissances. Enfin, nous utilisons ces architectures dans des tâches nouvelles, beaucoup plus difficiles, qui repoussent les limites des modèles de génération de texte qui fonctionnent bien aujourd'hui : des tâches qui nécessitent des connaissances mais qui exigent également que les modèles produisent des résultats longs et structurés en langage naturel, comme répondre à des questions complexes ou écrire des articles complets sur Wikipédia.

Un thème récurrent dans ce travail est le *applicabilité* des systèmes de génération non seulement pour la recherche, mais aussi pour une utilisation générale. Comme les systèmes d'intelligence artificielle continuent d'avoir de plus grandes capacités et fonctionnalités grâce aux progrès de la recherche, ce qui était autrefois des méthodologies de recherche aura des impacts plus larges sur la vie quotidienne des gens. Dans cette thèse, nous développons une variété d'architectures neuronales fondamentales, de jeux de données et de méthodologies d'évaluation qui sont largement applicables à différentes tâches, mais qui se concentrent également sur des domaines spécifiques où ils pourraient être appliqués. Ainsi, nous considérons des méthodes, telles que les systèmes de génération multilingue, pour les défis qui peuvent dépasser le centre d'intérêt prédominant de la recherche sur le texte anglais pour atteindre un public plus large et mondial. Nous nous concentrons également sur l'efficacité et l'évolutivité de divers mécanismes de récupération qui sont importants non seulement pour un entraînement rapide mais aussi pour un déploiement pratique des modèles neuronaux. Enfin, nous nous concentrons sur un grand nombre de tâches — de la simplification de phrases au dialogue, en passant par la réponse à des questions, la génération de Wikipedia et la vérification des faits — comme applications possibles de notre travail. Aujourd'hui, la majorité de ces applications restent à la frontière de la recherche. Nous espérons que le développement d'ensembles de données d'entraînement et d'évaluation, ainsi que la recherche sur les méthodologies de modélisation, inciteront d'autres chercheurs à se joindre à nous pour améliorer l'accessibilité de la technologie de génération pour un usage quotidien.

DEUX MOITIÉS

Cette thèse étudie deux défis centraux dans les systèmes de génération de texte : la génération au-delà de l'anglais et l'incorporation de connaissances. Ces éléments sont essentiels à la création de modèles génératifs de langage naturel capables de gérer des objectifs d'écriture profondément complexes, semblables à ceux des humains, et d'être utiles aux personnes du monde entier. Imaginez une expérience transparente où des fils de discussion Messenger incroyablement longs dans plusieurs langues différentes pourraient être instantanément résumés pour les personnes essayant de rattraper la conversation, ou des personnes travaillant sur des travaux écrits tels que des articles encyclopédiques pourraient immédiatement accéder à des brèves organisées d'informations connexes et commencer avec un article entièrement écrit comme squelette à éditer. Ces expériences ne semblent pas très éloignées de la technologie que nous considérons déjà comme acquise et que nous utilisons tous les jours. Cependant, elles représentent également des avancées fondamentales que nous, en tant que communauté de recherche en génération de texte, devons encore faire — ainsi que des expériences qui pourraient devenir possibles en intégrant les diverses avancées faites dans cette thèse.

La première moitié de cette thèse étudie les problèmes de génération de texte à texte et se concentre sur les avancées de la génération multilingue. Nous développons une méthode basée sur l'exploration qui peut créer des ensembles de données d'entraînement supervisées pour une variété de langues différentes, appliquée à la tâche de simplification des phrases. Par la suite, nous proposons une méthode permettant d'utiliser le pré-entraînement multilingue et les encastrements inter-lingues pour créer un modèle multilingue capable de décoder dans plus de 20 langues différentes, à partir de la même entrée AMR anglaise structurée. Prises ensemble, ces méthodes sont applicables à une variété d'autres tâches et représentent des approches complémentaires à la génération multilingue — la création automatique de jeux de données et la représentation efficace de plusieurs langues dans un seul modèle.

La seconde moitié de cette thèse relève le défi de l'incorporation des connaissances. Chaque jour, les humains apprennent de nouvelles informations et effectuent des actions qui synthétisent et intègrent des connaissances provenant de diverses sources. Imaginez, par exemple, que vous décidiez si vous voulez voir le dernier film qui est sorti — vous pouvez lire quelques critiques et un article de presse sans spoiler pour décider si vous voulez le voir ou non. Ces tâches nous paraissent simples, mais sont complexes pour les machines car, fondamentalement, elles nécessitent une découverte et une utilisation efficaces de l'information. Nous proposons une variété d'approches et d'architectures qui permettent à la fois de récupérer des informations à partir de grandes sources de connaissances externes et de les incorporer efficacement dans des modèles standard de séquence à séquence. De plus, nous adaptons continuellement ces méthodes pour traiter des sources de connaissances de plus en plus grandes, depuis les documents individuels nécessaires pour répondre à des questions jusqu'aux données du web ouvert nécessaires pour écrire des articles entiers sur Wikipédia.

Nous nous concentrons sur une mise à l'échelle efficace et des composants architecturaux modulaires qui peuvent s'étendre à une variété de tâches différentes.

CADRE DE LA THÈSE

- CHAPITRE 2 introduit la section sur la génération de texte sans récupération. Elle commence par une discussion sur les objectifs généraux des systèmes de génération de langage naturel. Nous motivons les deux chapitres suivants sur le texte-à-texte et la représentation du sens-à-texte. En particulier, nous nous concentrons sur le défi de la génération *multilingue*, ou des systèmes automatisés qui peuvent produire du langage naturel dans des langues autres que l'anglais.

- CHAPITRE 3 Nous décrivons les systèmes de génération de *texte-à-texte* qui prennent en entrée des connaissances textuelles en langue naturelle sous la forme d'une phrase et produisent une sortie en langue naturelle. Nous motivons l'importance et les défis de telles tâches, avant de plonger dans les détails de notre système de simplification de phrases multilingues. Nous montrons que notre système peut atteindre l'état de l'art en matière de simplification de phrases en anglais, mais qu'il peut également être appliqué à la simplification de phrases en français et en espagnol.

- CHAPITRE 4 va au-delà des entrées en langage naturel pour s'intéresser aux systèmes de génération, et décrit des systèmes capables de générer à partir d'entrées de connaissances structurées. Nous nous concentrons sur la tâche de génération de *représentation de sens-à-texte*, et plus particulièrement sur les représentations de sens abstraites. Nous proposons une tâche de génération AMR-texte multilingue et démontrons qu'en combinant diverses avancées, il est possible de créer un modèle qui génère du texte en 21 langues à partir d'AMR.

- CHAPITRE 5 introduit la section sur la génération de texte avec *retrieval*. Il commence par une discussion sur les raisons pour lesquelles les systèmes de génération ont besoin d'accéder au *knowledge* pour gérer des tâches de complexité croissante. Nous motivons les quatre chapitres suivants qui développent des modèles pour diverses tâches, mais à des échelles de plus en plus grandes de recherche d'information.

- CHAPITRE 6 Nous commençons par décrire l'état actuel d'une grande partie des travaux sur la génération basée sur les connaissances, qui se concentrent sur la lecture de courts paragraphes du texte d'entrée comme preuve. Dans ce chapitre, nous décrivons un système qui ingère un document entier et nous nous concentrons sur une tâche de vérification des faits qui génère un dossier structuré de questions-réponses pour aider les gens à comprendre si un texte sur Internet est vrai ou faux. Pour générer ce dossier de questions-réponses, les modèles ingèrent des preuves à partir d'un seul document trouvé sur le Web.

- CHAPITRE 7 étend l'accès aux connaissances à plusieurs documents, récupérés sur Wikipédia. Il se concentre sur la récupération de contenu pertinent à partir d'un sous-ensemble de Wikipédia pour créer des systèmes de dialogue génératifs de type chit-chat capables de discuter de n'importe quel sujet avec l'utilisateur. Nous commençons par nous concentrer sur l'efficacité de l'extraction, car la consultation d'un plus grand nombre de documents exige rapidité et précision. De plus, nous étendons notre système pour utiliser de multiples sources de connaissances dans différentes modalités.

- CHAPITRE 8 Nous étendons le besoin de systèmes basés sur la recherche au-delà de Wikipédia, au web ouvert. Nous décrivons la tâche d'écrire un article complet sur Wikipédia, qui nécessite de rechercher sur Internet de grandes quantités d'informations, de réduire le contenu aux connaissances les plus pertinentes, puis d'écrire un article complet et bien structuré. Nous nous concentrons particulièrement sur le cas des biographies, afin de remédier au biais de représentation des femmes sur Wikipédia.

- CHAPITRE 9 Nous examinons enfin les tâches à l'échelle du Web qui prennent comme entrée non pas le langage naturel, mais une entrée structurée sous la forme d'un graphe de connaissances. Contrairement au texte, les entrées structurées telles que les tableaux ou les graphes ont des schémas prédéfinis qui facilitent leur interprétation, leur interrogation et leur utilisation pour réduire la redondance. Cependant, ces graphes de connaissances structurés n'existent pas nécessairement. Nous décrivons un système de génération de graphes de connaissances à partir de textes Web et l'appliquons aux tâches de réponse à des questions de longue durée et de génération d'articles Wikipédia.

- CHAPITRE 10 Nous concluons dans le chapitre 10 en résumant les principales contributions de recherche de cette thèse et réfléchissons aux futures pistes de recherche.

Acknowledgments

I thank my advisors, Claire Gardent, Antoine Bordes, and Chloé Braud for their guidance, patience, and encouragement of scientific pursuit and personal development. I thank my thesis committee and reviewers, André Martins, Benoit Crabbé, and Natalie Schluter for their generous time. I thank my husband for his continued support each and every day.

Contents

ABSTRACT ix

RÉSUMÉ xi

SOMMAIRE xiii
  Introduction . . . xiii
  Deux Moitiés . . . xv
  Cadre de la thèse . . . xvi

ACKNOWLEDGMENTS xix

CONTENTS xxi

LIST OF FIGURES xxii

LIST OF TABLES xxiv

I PROLOGUE 1

1 INTRODUCTION 3
  1.1 Thesis Outline . . . 4
  1.2 Publications . . . 6

II TEXT GENERATION WITHOUT RETRIEVAL 7

2 TEXT GENERATION 9
  2.1 The Need for Text Generation . . . 9
  2.2 Structure of this Section . . . 10

3 TEXT-TO-TEXT GENERATION 11
  3.1 Text-to-Text Generation . . . 11
  3.2 Multilingual Sentence Simplification . . . 14

4 MEANING REPRESENTATION-TO-TEXT GENERATION 27
  4.1 Structured Input to Text Generation . . . 27
  4.2 Abstract Meaning Representations . . . 27
  4.3 Multilingual AMR-to-Text Generation . . . 29

III TEXT GENERATION WITH RETRIEVAL 43

5 RETRIEVAL FOR KNOWLEDGE-BASED TEXT GENERATION 45
  5.1 The Need for Knowledge . . . 45
  5.2 Challenges . . . 47
  5.3 Structure of this Section . . . 47

6 KNOWLEDGE FROM A SINGLE DOCUMENT 49
  6.1 Motivation: Document-Level Knowledge . . . 49

6.2 Fact Checking as a Knowledge-Based Text Generation Task . . . 50
6.3 Generating Fact Checking Briefs . . . 50

7 SCALING KNOWLEDGE ACCESS TO MULTIPLE DOCUMENTS IN WIKIPEDIA 63
7.1 Motivation: Knowledge from Multiple Documents . . . 63
7.2 Dialogue as a Knowledge-Based Text Generation Task . . . 63
7.3 Augmenting Transformers with KNN-Based Composite Memory . . . 64

8 SCALING KNOWLEDGE ACCESS TO THE OPEN WEB 85
8.1 Motivation: Knowledge from the Open Web . . . 85
8.2 Wikipedia Article Writing as a Knowledge-Based Text Generation Task . . . 85
8.3 Generating Biographies for Marginalized Groups on Wikipedia . . . 87

9 KNOWLEDGE ON THE WEB, IN STRUCTURED FORM 101
9.1 Motivation: Knowledge in Structured Form . . . 101
9.2 Long-form Question Answering as a Knowledge-Based Text Generation Task . . . 102
9.3 Using Local Knowledge Graph Construction to Scale Seq2Seq Models to Multi-Document Inputs . . . 103

IV EPILOGUE 117

CONCLUSION 119

BIBLIOGRAPHY 121

List of Figures

3.1 Sentence simplification Models for Any Language without Simplification Data 14
3.2 Model Ablations for Multilingual Sentence Simplification 24

4.1 Example AMR Representation of Sentence 28
4.2 Generating into Multiple Languages from English AMR 31
4.3 One-to-Many Architecture for Multilingual AMR-to-Text Generation 32
4.4 Attention Alignment When Decoding in French and German From the Same Input AMR 38
4.5 Relationship between BLEU Score and Token Overlap for All 21 Languages 39
4.6 BLEU Difference Training on Close v. Far Languages Within One Family 40

6.1 Fact Checking Briefs 51
6.2 Three Types of Briefs 51
6.3 Examples of QABriefs in QABriefDataset 52
6.4 Question and Answer Types 56
6.5 QABriefer Model 56
6.6 Accuracy and Time Taken to Fact Check by Different Groups 57
6.7 Usefulness of Briefs 58
6.8 Overconfidence When Given a QABrief 61

7.1 KIF Module overview 66
7.2 Human Evaluation Results on Both Datasets 74
7.3 Human Evaluation on the Unseen Test Set of Wizard of Wikipedia 74
7.4 Conversation Between Human and KIF-Augmented Transformer on Wizard of Wikipedia 75
7.5 Examples of Top-3 Fetched Training Utterances and Fetched Knowledge When Responding 76
7.6 Ablations on Wizard of Wikipedia 78

8.1 Model Architecture for Wikipedia Biography Generation 90
8.2 Example Generated Wikipedia Section 95
8.3 Human Evaluation of Generation of Wikipedia Biographies 96

9.1 Multi-Document Input to Linearized Graph 104
9.2 Steps of Graph Construction 106
9.3 Graph Attribute Embeddings 106
9.4 Model Architecture to read Text and Linearized Graph Input 107
9.5 Graph Statistics From the ELI5 Training Set 112
9.6 Graph Statistics From the ELI5 Training Set, Part 2 112
9.7 Interpretable Attention of Seq2Seq Models on a Subgraph When Answering a Question in ELI5 114

List of Tables

3.1 Statistics of Mined Paraphrase Training Corpora Compared to Standard WikiLarge Labeled for Simplification 17
3.2 Unsupervised and Supervised Sentence Simplification for English 21
3.3 Examples of Generated Simplifications 22
3.4 Unsupervised Sentence Simplification in French and Spanish . . 22
3.5 Human Evaluation Results 25

4.1 Results on 21 Languages in Europarl 34
4.2 Results on Gold AMR from LDC2015E86 35
4.3 Human Evaluation Results 36
4.4 Example Paraphrases generated by our multilingual model . . . 37
4.5 Human Evaluation of Our Approach Compared to the Hybrid English AMR to Text + Machine Translation Baseline . . 38
4.6 Performance When Training With Increasingly More Languages 38

6.1 Statistics of QABriefDataset 56
6.2 Question Generation Models 60
6.3 Question Answering Models 60

7.1 Results on the Wizard of Wikipedia Dataset 72
7.2 Results on the Engaging ImageChat Dataset 73
7.3 Effect of Fetched Information on Generated Utterances 79
7.4 Using Multiple KIF Modules on Multiple Sources 81
7.5 Important Features for KNN Search 81
7.6 Multi-hop With KIF to Retrieve Information With Multiple Fetch Steps 82

8.1 Example Section Headings and Occupations in our Evaluation Dataset 92
8.2 Breakdown and Statistics of Biographies in our Evaluation Dataset 93
8.3 Full Results on Biography Generation on Our Constructed Evaluation Dataset 95
8.4 Ablations of Types of Queries for the Retrieval Module and Generation Section by Section 95
8.5 ROUGE-L Performance Broken Down by Sub-Categories 96
8.6 Statistics of Biographies on Wikipedia 96
8.7 Improved Performance when Finetuning on Biographical Articles About Women 96

9.1 Answer Generation on ELI5 Using Seq2Seq Models Receiving the Question and a Support Document to Produce the Answer 111

9.2 Lead Paragraph Generation on WikiSum CommonCrawl Using Seq2Seq Models Receiving the Title and a Support Document to Produce the Lead Paragraph 111
9.3 Ablations on the ELI5 Validation Set 113
9.4 Importance of Web Search Relevance on Validation for ELI5 . . . 113

Part I

PROLOGUE

1

Introduction

"You can make anything by writing."
—C.S. Lewis

▸ PREAMBLE. Every day we write — from sending your mother a quick text to drafting a scientific article such as this thesis. The writing we do often goes hand-in-hand with automated assistance. For example, modern instant messaging software often suggests what word to write next, emails can be started with an autocomposer, and essays are improved with machine-suggested edits. These technologies are powered by years of research on *text generation*, a natural language processing field with the goal of automatically producing fluent, human-readable natural language. At a small scale, text generation systems can generate individual words or sentences, but have wide-reaching applications beyond that. For instance, systems for summarization, dialogue, and even the writing of entire Wikipedia articles are grounded in foundational text generation technology.

Producing fluent, accurate, and useful natural language faces numerous challenges. Recent advances in text generation, principally leveraging training neural network architectures on large datasets, have significantly improved the surface-level readability of machine-generated text. However, current systems necessitate improvement along numerous axes, including generation beyond English and writing increasingly longer texts. While the field has seen rapid progress, much research focus has been directed towards the English language, where large-scale training and evaluation datasets for various tasks are readily available. Nevertheless, applications from autocorrect to autocomposition of text should be available universally. After all, by population, the majority of the world does not write in English. In this work, we create text generation systems for various tasks with the capability of incorporating languages beyond English, either as algorithms that easily extend to new languages or multilingual models encompassing up to 20 languages in one model.

Beyond our work in multilingual text generation, we focus on a critical piece of generation systems: *knowledge*. A pre-requisite to writing well is knowing what to write. This concept of knowledge is incredibly important in text generation systems. For example, automatically writing an entire Wikipedia article requires extensive research on that article topic. The instinct to research is often intuitive — decades ago people would have gone to a library, replaced now by the information available on the World

Wide Web. However, for automated systems, the question is not only what knowledge to use to generate text, but also how to *retrieve* that knowledge and best utilize it to achieve the intended communication goal.

We face the challenge of retrieval-based text generation. We present several techniques for identifying relevant knowledge at different scales: from local knowledge available in a paragraph to sifting through Wikipedia, and finally identifying the needle-in-the-haystack on the scale of the full web. We describe neural network architectures that can perform large-scale retrieval efficiently, utilizing pre-computation and caching mechanisms. Beyond how to retrieve knowledge, we further investigate the *form* the knowledge should take — from natural language such as Wikipedia articles or text on the web to structured inputs in the form of knowledge graphs. Finally, we utilize these architectures in novel, much more challenging tasks that push the boundaries of where text generation models work well today: tasks that necessitate knowledge but also require models to produce long, structured natural language output, such as answering complex questions or writing full Wikipedia articles.

► UTILITY OF TEXT GENERATION SYSTEMS.

A theme throughout this work is the *applicability* of generation systems not only for research, but for general use. As artificial intelligence systems continue to have greater capabilities and functionality as a result of research advances, what once were research methodologies will have broader impacts on people's everyday lives. In this thesis, we develop a variety of fundamental neural architectures, datasets, and evaluation methodologies that are broadly applicable to different tasks, but also focus on specific areas where they could be applied. Thus, we consider methods, such as multilingual generation systems, for challenges which can push beyond the predominant research focus of English text to a wider, global audience. We focus further on the efficiency and scalability of various retrieval mechanisms which are important for not only fast training but also practical deployment of neural models. Finally, we focus on a vast number of tasks — from sentence simplification to dialog to question answering to Wikipedia generation to fact checking — as possible applications of our work. Today, the majority of these applications remain at the research frontier. We hope that the development of training and evaluation datasets, along with research in modeling methodologies, excites other researchers to join us in improving the accessibility of generation technology for daily use.

1.1 THESIS OUTLINE

► CHAPTER 2 introduces the section on text generation without retrieval. It begins with a discussion of the broad goals of natural language generation systems. We motivate the subsequent two chapters on text-to-text and meaning-representation-to-text. In particular, we focus on the challenge of *multilingual* generation, or automated systems that can produce natural language in languages beyond English.

- CHAPTER 3 describes *text-to-text* generation systems that take as input natural language textual knowledge in the form of a sentence and produce a natural language output. We motivate the importance and challenges of such tasks, before diving into detail on our system for multilingual sentence simplification. We show that our system can achieve state-of-the-art on English sentence simplification, but can also be applied to simplify French and Spanish sentences.

- CHAPTER 4 moves beyond natural language inputs to generation systems, and describes systems that can generate from structured knowledge inputs. We focus on the task of *meaning-representation-to-text* generation, specifically abstract meaning representations. We propose a multilingual AMR-to-Text generation task and demonstrate that by combining various advances, it is possible to create one model that generates text 21 languages from AMR.

- CHAPTER 5 introduces the section on text generation with *retrieval*. It begins with a discussion of why generation systems require access to *knowledge* to handle tasks of increasing complexity. We motivate the subsequent four chapters that develop models for various tasks, but at larger and larger scales of information retrieval.

- CHAPTER 6 opens by describing the current state of much work in knowledge-based generation, which focuses on reading short paragraphs of input text as evidence. In this chapter, we describe a system that ingests an entire document and focus on a fact checking task that generates a structured question-and-answer brief to help people understand whether text on the internet is true or false. To generate this question-and-answer brief, models ingest evidence from a single document found on the web.

- CHAPTER 7 extends knowledge access to multiple documents, retrieved from Wikipedia. It focuses on retrieving relevant content from a subset of Wikipedia to create generative chit-chat dialog systems that can discuss any topic with the user. We begin to focus on retrieval efficiency, as looking through a greater number of documents requires speed and accuracy. Further, we extend our system to utilize multiple sources of knowledge in different modalities.

- CHAPTER 8 scales the need for retrieval-based systems beyond Wikipedia to the open web. We describe the task of writing a full Wikipedia article, which requires searching on the internet for large quantities of information, narrowing down the content to the most relevant knowledge, and then writing a fully formed, well structured Wikipedia article. We particularly focus on the case of biographies, to address the bias in representation of women on Wikipedia.

- CHAPTER 9 finally considers web-scale tasks that take as input not natural language, but a structured input in the form of a knowledge graph. Unlike text, structured input such as tables or graphs have predefined schemas that make them easier to interpret, query, and utilize to reduce redundancy.

However, these structured knowledge graphs may not necessarily exist. We describe a system for generating knowledge graphs from web text and apply it to the tasks of long-form question answering and Wikipedia article generation.

- CHAPTER 10 We conclude in Chapter 10 by summarizing the main research contributions of this thesis and reflect on future avenues of research.

1.2 PUBLICATIONS

Portions of this thesis have appeared elsewhere in the following publications:

- CONFERENCE PUBLICATIONS
    - *Multilingual AMR to Text Generation*. Empirical Methods in Natural Language Processing, 2020. Angela Fan, Claire Gardent.
    - *Generating Fact Checking Briefs*. Empirical Methods in Natural Language Processing, 2020. Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, Sebastian Riedel
    - *Using Local Knowledge Graph Construction to Scale Seq2Seq Models to Multi-Document Inputs*. Empirical Methods in Natural Language Processing, 2019. Angela Fan, Claire Gardent, Chloe Braud, Antoine Bordes
    - *ELI5: Long Form Question Answering*. Association for Computational Linguistics 2019. Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, Michael Auli.

- JOURNAL PUBLICATIONS
    - *Augmenting Transformers with KNN-Based Composite Memory for Dialog*. Transactions of the Association for Computational Linguistics, 2020. Angela Fan, Claire Gardent, Chloe Braud, Antoine Bordes

- OTHER PREPRINTS AND SUBMITTED WORK
    - *Multilingual Unsupervised Sentence Simplification*. Louis Martin, Angela Fan, Eric de la Clergerie, Benoit Sagot, Antoine Bordes.
    - *Generating Biographies for Marginalized Groups on Wikipedia*. Angela Fan, Claire Gardent.

Part II

TEXT GENERATION WITHOUT RETRIEVAL

2

Text Generation

► SYNOPSIS We motivate the need for text generation systems, or automated mechanisms for writing natural language. We describe systems that ingest input data in various forms, and produce as output written, human readable text. We focus on input data in the form of natural language (text-to-text generation) and in the form of meaning representations (meaning-representation-to-text generation).

2.1 THE NEED FOR TEXT GENERATION

Rapid advancements in artificial intelligence have changed how we live our everyday lives. We can text people quickly using autosuggested content, we can communicate freely online using translation services, and we can book plane tickets through automated dialog applications. These conveniences that we experience everyday are powered by text generation technology[1]. For example, the autosuggest on your phone analyzes the text you have already written and provides suggestions for what you might want to type next by understanding the distribution of likely next words. This fundamental technology is language modeling, and underlies many of the advancements in neural text generation systems today. While generation technology already permeates our everyday lives, what if we want to work towards more and more complex applications? Imagine systems that can help users draft entire emails, summarize long message threads, make a complicated book more easy to read, or freely chit-chat with someone all day. Unlike the next-word autosuggest on your phone, these applications are at the frontier of text generation research today. In this series of chapters, we discuss the challenges facing the development of more complex generation systems, and provide several novel methods to create such advanced systems.

2.1.1 *The Challenge of Multilingual Generation*

> *"Dear Computer Scientists, 'Natural Language' is* not *a synonym for 'English'. That is all."*
> —Emily Bender

Our particular area of focus in the subsequent chapters of this section is on the challenge of multilingual text generation, or generation in languages beyond English. For a plethora of complex reasons, scientific progress in text generation has often focused on improving the performance of systems only in English. However, billions of people around the world speak languages

[1] The development of autocomplete features has a long history, and was actually first created to help those with disabilities type and produce text with greater efficiency. Afterwards, it was used in industries such as medicine, with complex and difficult-to-spell terminology. Now, such features are commonplace.

beyond English, and the vast majority of text generation applications are applicable to them as well. For example, sentence simplification systems can help children more easily understand complex written text regardless of language. In part, systems have been limited to English due to lack of data for training models in other languages, though a number of other factors contribute (such as the lack of high-quality evaluation datasets). Further, developing systems for individual languages would explode the number of systems that need to be created. Thus, we focus on methodologies that can create datasets for multiple different languages, as well as models that can produce generations in more than one language at a time.

2.2 STRUCTURE OF THIS SECTION

In the subsequent chapters, we describe methodology and applications of text generation systems that **first** take as input natural language, and **second** a structured form of meaning representation called an Abstract Meaning Representation. We provide brief background on text-to-text and meaning-representation-to-text as broad paradigms, and introduce specific challenges in each area. We then present methods for specific applications of generation systems in each of these paradigms, with a particular focus on multilingual generation.

3

Text-to-Text Generation

"What I try to do is write. I may write for two weeks 'the cat sat on the mat, that is that, not a rat.' And it might be just the most boring and awful stuff. But I try."
—Maya Angelou

▸ SYNOPSIS We begin with a chapter on *text-to-text* generation, a broad term for the set of tasks where systems read a textual input and produce a textual output. This chapter defines text-to-text generation and discuss its challenges, then focuses on a large-scale semi-supervised generation approach for the task of *sentence simplification*. We particularly focus on multilingual simplification, or being able to simplify text in languages other than English. We describe a generalized way to collect large-scale training data from the internet to train models in other languages, and then a controllable generation method to adjust sentence simplifications to user needs.

3.1 TEXT-TO-TEXT GENERATION

Text generation systems have the broad goal of producing understandable, fluent, and readable textual output in natural language. These systems often produce text based on ingesting input data. For example, machine translation models intake text in a source language, and translate the text, generating the translation in the target language. Generation systems can intake various different forms of input data, from freeform text to structured representations to images. In this section, we focus on *text-to-text* generation, where the input is natural language.

Recent progress in deep learning for natural language processing has been driven by the *Sequence-to-Sequence* architecture (Sutskever et al., 2014), most widely used in text-to-text generation tasks. These architectures use an encoder neural network to encode a sequence of input text, mapping the sequence to a vector representation, then employ a decoder neural network to generate the target text. Various neural architectures have been used for encoders and decoders, from Recurrent Neural Networks (Rumelhart et al., 1986; Hopfield, 1982) and the related Long Short-Term Memory models (Hochreiter and Schmidhuber, 1997) to Convolutional Neural Networks (Dauphin et al., 2017; Gehring et al., 2017), and most recently Transformers (Vaswani et al., 2017). These sequence-to-sequence architectures connect the encoder and decoder networks using an attention mechanism (Bahdanau et al., 2014) that enables the decoder network to focus on a subset of the input sequence when decoding.[1]

[1] We refer the reader to Rush (2018) for a detailed explanation of state-of-the-art Transformer sequence-to-sequence architectures.

3.1.1 *Archetypal Applications*

Text-to-text generation has a variety of widely-used, important applications, which have contributed to the development of various subfields of natural language generation focusing on each of these tasks. We briefly detail a select subset as examples, which provide an overview of the tasks we consider in this thesis.

- MACHINE TRANSLATION The goal of machine translation is to seamlessly convert text from one language into another. These models encode a sequence of text in one language, and decode the sequence in another language. Machine translation models have paramount importance, from enabling people to travel to different countries more easily to allowing people to read content online in another language.[2]

- SENTENCE SIMPLIFICATION Sentence simplification has the goal of reducing the complexity of a sentence while retaining the original meaning. Such simplification systems contribute to greater readability and accessibility of textual content, which has meaningful applications for digital literacy, making it easier for children and non-native speakers to read content, and aiding those with cognitive disabilities.

- ABSTRACT MEANING REPRESENTATION TO TEXT Abstract Meaning Representations can be broadly thought of as semantic representation languages, or a way of representing content in text that is abstracted from the syntactic representation. These systems are studied to produce natural language from structured forms, which has applicability to verbalizing information in databases, tables, or other representations that are not strictly natural language in nature.

- SUMMARIZATION Summarization systems have the goal of briefly presenting the main points of a longer work. These systems encode a long sequence of text and generate a significantly shortened, condensed version that retains the important information. Summarization has practical applicability in helping people digest information efficiently, from reading newspaper articles to emails to long message threads.

- QUESTION ANSWERING Broadly, question answering systems seek to help users find answers to their questions, from asking Siri what the weather is like to one day being able to explain a complex concept to answer questions such as *why is the sky blue?*. Most systems read user questions and search for answers in a larger base of knowledge — from a single paragraph to the entire web, then produce answers.

- DIALOG There are a diverse array of possible formulations of dialog tasks, from helping users accomplish specific objectives (booking an airline ticket) to simply chatting with a user to keep them company[3]. Dialog models often receive various inputs, from the conversation history to pluggable knowledge sources, such as a database of when the next Air France flight will depart to

[2] The early Georgetown-IBM experiment, which translated a set of Russian sentences into English in 1954, was one of the earliest recorded research efforts in machine translation. Researchers on the project expressed optimism that the problem of automatic machine translation would be solved in a few years. While readily available online systems such as Google Translate exist and translation is one of the most recognizable commercial applications of neural text generation, it remains a challenging task.

[3] In particular, such systems have been featured in quite a number of popular films, such as `https://en.wikipedia.org/wiki/Her_(film)`

Wikipedia articles. The goal of most generative dialog systems is to produce natural sounding chitchat that will engage users.

- WIKIPEDIA ARTICLE WRITING Finally, as text generation technology continues to improve, more and more challenging tasks have become popular. Recent work has focused on generative systems that can write not single sentences, but entire Wikipedia articles. These systems necessitate advanced capabilities in digesting large quantities of information as a basis to write the article, as well as novel methods to write multiple paragraphs that are coherent and flow together seamlessly. Long-form generative systems such as these would have strong applicability in various domains, from Wikipedia articles and beyond.

3.1.2 *Challenges*

While text-to-text generation has seen great progress, a number of challenges face the field. We discuss several challenges below, and describe in the next section how we address many of these for the task of sentence simplification.

- RELIANCE ON LARGE DATASETS

Recent architectural (Vaswani et al., 2017) and methodological advances, particularly the development of pretraining-and-finetuning regimes (Devlin et al., 2019) have substantially increased the surface-level fluency of generation systems. However, these advances necessitate the existence of large, clean corpora for both pretraining and finetuning. While possible for a small handful of languages[4] and tasks, this limitation is an obstacle to the development of specialized systems for more languages. For example, creating a system for sentence simplification still requires[5] data for supervised finetuning on that specific task.

> *"The hard things to measure are often the most important things to measure."*
> —Stacey Barr

- MULTI-FACETED EVALUATION

Writing is a complex task, and measuring the quality of the written word is complex and nuanced. Most generation tasks are fundamentally open-ended, because the ability to say something in multiple different ways is inherent in natural language. This inherent variability in generation tasks, linked to the paraphrasing power of natural language, exists across almost all tasks. However, certain tasks can be more difficult to evaluate than others. If we take sentence simplification as an example — there are multiple ways to simplify a sentence, multiple types of simplification that could be applied, and so on, which adds task-dependent variability. Thus, whether a certain simplification is adequate or not can be difficult to quantify. Further, generation systems not only have to generate text, but must write fluent and accurate natural language. The grammar and sentence structure must be clear, and the generated language must be factually accurate. The evaluation of fluency and factuality is challenging, particularly when it must also be applied across multiple languages.

[4] A limitation we discuss in greater detail in the subsequent thought.

[5] We note a number of exciting advances in zero-shot learning for text generation, though these methods remain far from the performance of systems trained on supervised data for specific tasks.

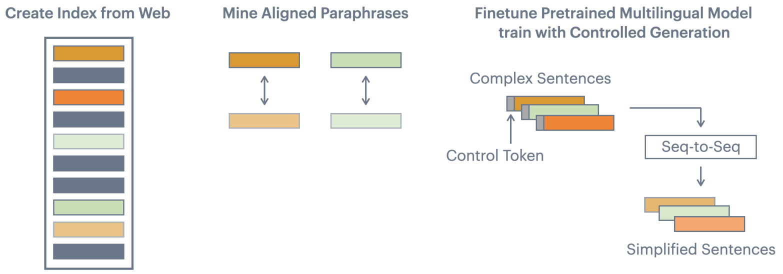


FIGURE 3.1: **Sentence Simplification Models for Any Language without Simplification Data**. Sentences from the web are used to create a large scale index that allows mining millions of paraphrases. Subsequently, we finetune pretrained models augmented with controllable mechanisms on the paraphrase corpora to achieve sentence simplification models in any language.

3.2 MULTILINGUAL SENTENCE SIMPLIFICATION

In this section, we focus on the text-to-text generation task of *Sentence Simplification*. While simplification has a wide array of applications, many existing methods rely on large corpora of aligned complex and simple sentences, restricting the systems to English. We describe our unsupervised, multilingual sentence simplification method, which leverages large-scale mining in multiple languages to build simplification systems for languages beyond English. We describe challenges of evaluation, which are exacerbated when working on generation in a multilingual setting, and how we approach the evaluation of multilingual simplification systems.

3.2.1 *Introduction*

"I didn't have time to write a short letter, so I wrote a long one instead."
—Mark Twain

Sentence simplification is the task of reducing the lexical and syntactic complexity of a sentence, while retaining most of its original meaning. The broad goal of simplification systems is to improve readability and understanding. Simplification has a variety of important societal applications, for example increasing accessibility for those with cognitive disabilities such as aphasia (Carroll et al., 1998), dyslexia (Rello et al., 2013), and autism (Evans et al., 2014), or for non-native speakers (Paetzold and Specia, 2016b) and children with reading difficulties (Gala et al., 2020). Research has mostly focused on English simplification, where source texts and their associated simplified texts exist and can be automatically aligned, such as English Wikipedia and Simple English Wikipedia (Zhang and Lapata, 2017), or the NEWSELA simplified news corpus (Xu et al., 2015). However, such data is limited in terms of size, domain (encyclopedic or news), and difficult to find in other languages.

We focus on mining paraphrases to train simplification systems in multiple languages without *labeled*[6] simplification data. Using the controllable text generation mechanisms proposed in Martin et al. (2020), we leverage this paraphrase data to train simplification models that control attributes such as length, lexical and syntactic complexity.

The method we describe is *unsupervised* as it does not require labeled parallel simplifications, though we mine using supervised embeddings (LASER embeddings trained with machine translation data). The lack of labeled simplification data has been a considerable bottleneck in data-driven sentence simplification research, which we propose to remove in this work.[7]

Simplifications are a special type of paraphrases where the target sentence is easier to read and understand than the source sentence. In this work,

[6] We use the term *labeled simplifications* to refer to parallel datasets where source texts were manually simplified by humans.

[7] Previous publications have also used the term *unsupervised simplification* to describe works that do not make use of any labeled parallel simplification datasets while leveraging supervised components such as constituency parsers and knowledge bases (Kumar et al., 2020), external synonymy lexicons (Surya et al., 2019), and databases of simplified synonyms (Zhao et al., 2020b).

we mine paraphrases instead of only mining the subset of paraphrases that correspond to simplifications. This removes the need to decide in advance the type of simplifications that should be mined. We show that a model trained on a dataset mined with fewer assumptions and adjusted at test time reaches better experimental results.

These automatically created corpora benefit from large quantities of data in various languages available online. We apply this technique to mine English, French, and Spanish paraphrase training data. On English, we show that the quality of the models trained with mined unlabeled data is similar to that of models trained with automatically aligned English Wikipedia and Simple English Wikipedia. Subsequently, we use multilingual pretraining with BART and mBART to further incorporate unsupervised training in our sentence simplification models (Lewis et al., 2020a). By using pretrained models and finetuning on our mined paraphrase datasets, we are able to achieve state-of-the-art results with no labeled simplification data in multiple languages.

3.2.2 *Related work*

▸ SENTENCE SIMPLIFICATION IN MULTIPLE LANGUAGES Data-driven methods have been predominant in English Sentence Simplification in the recent years, requiring large supervised training corpora of complex-simple aligned sentences (Wubben et al., 2012; Xu et al., 2016; Zhang and Lapata, 2017; Zhao et al., 2018c; Martin et al., 2020). Methods have relied on using the Wikipedia edit history (Botha et al., 2018), or more notably on English Wikipedia and Simple English Wikipedia with automatic alignment of sentences from similar articles (Zhu et al., 2010; Coster and Kauchak, 2011; Woodsend and Lapata, 2011; Kauchak, 2013; Zhang and Lapata, 2017). Using Wikipedia and automatic alignments proved to have various flaws compared to professional simplifications of the NEWSELA corpus (Xu et al., 2015). Such professional datasets are however rare, and often come with restrictive licenses that hinder reproducibility and widespread usage.

Multiple efforts explored simplification in other languages such as Brazilian Portuguese (Aluísio et al., 2008), Spanish (Saggion et al., 2015; Štajner et al., 2015), Italian (Brunato et al., 2015; Tonelli et al., 2016), Japanese (Goto et al., 2015; Kajiwara and Komachi, 2018; Katsuta and Yamamoto, 2019), and French (Gala et al., 2020), but the lack of a large labeled parallel corpora has been problematic.[8] In this work, we show that a method trained on automatically mined large corpora in three languages can reach state-of-the-art results in each language.

▸ UNSUPERVISED SIMPLIFICATION When labeled parallel simplification data is unavailable or unused, sentence simplification systems rely on unsupervised simplification techniques, often based on methods from machine translation. The prevailing approach is to train unsupervised models by splitting a monolingual corpora into disjoint sets of complex and simple sentences using readability metrics. Kajiwara and Komachi (2018) train statistical machine translation models on unsupervised alignment of English Wikipedia,

[8] We refer the reader to more detail in the survey by Alva-Manchego et al. (2020b).

based on word embeddings. Other methods train on disjoint complex-simple sets without sentence-level alignment using auto-encoders (Surya et al., 2019; Zhao et al., 2020b), unsupervised statistical machine translation for Japanese (Katsuta and Yamamoto, 2019), and back-translation in Spanish and Italian (Aprosio et al., 2019). More recently, (Kumar et al., 2020) have proposed to iteratively edit the constistuency parse tree of a complicated sentence to create unsupervised simplifications using a tailored scoring function. The performance of such unsupervised methods are often below their supervised counterparts. We remove the need for separating monolingual corpora in two disjoint sets and instead mine paraphrases directly to train models with state-of-the-art performance.

▸ MINING Previous work on the unsupervised creation of corpora from text has focused on mining parallel data for machine translation systems[9]. Works leveraged document retrieval (Munteanu and Marcu, 2005), language models (Koehn et al., 2018, 2019), and embedding space alignment (Artetxe and Schwenk, 2019) to create large corpora (Tiedemann, 2012; Schwenk et al., 2019). We focus on paraphrasing for sentence simplifications, which presents new challenges. Unlike machine translation, where the same sentence should be identified in two languages, we develop a method to identify varied paraphrases of sentences. Mining in translation has leveraged heuristics such as similar length, but paraphrases, and most specifically simplifications, have a wide array of surface forms, including multiple sentences, different vocabulary usage, and removal of content from more complex sentences.

Previous work in unsupervised paraphrasing has aligned sentences from various corpora (Barzilay and Lee, 2003) with a variety of different objective functions (Liu et al., 2019a). Bilingual pivoting relied on MT datasets to create large databases of word-level paraphrases (Pavlick et al., 2015), lexical simplifications (Pavlick and Callison-Burch, 2016; Kriz et al., 2018), or sentence-level paraphrase corpora (Wieting and Gimpel, 2018). This has not been applied to multiple languages or the task of sentence simplification. We further only use raw monolingual data to create our paraphrase corpus instead of relying on parallel MT datasets.

3.2.3 *Method*

We describe how to create simplification models in multiple languages without labeled data. We mine a large quantity of paraphrases from Common Crawl using LASER (Artetxe and Schwenk, 2019), a multilingual text embedding system, and `faiss` (Johnson et al., 2019), a library for fast and efficient nearest neighbor search. We then scale our paraphrase mining approach to multiple languages to create automatically aligned paraphrase corpora in each language. Finally, we leverage controllable generation mechanisms and unsupervised pretraining to train high quality simplification models on paraphrase corpora.

Mining Paraphrases in Many Languages

[9] The related works section of Schwenk et al. (2019) provides a detailed history of efforts in mining.

	Type	# Sequence Pairs	# Avg. Tokens per Sequence
WIKILARGE (English)	Labeled Parallel Simplifications	296,402	original: 21.7 simple: 16.0
NEWSELA (English)	Labeled Parallel Simplifications	94,206	original: 23.4 simple: 14.2
English	Mined	1,194,945	22.3
French	Mined	1,360,422	18.7
Spanish	Mined	996,609	22.8

TABLE 3.1: Statistics of mined paraphrase training corpora compared to standard WIKILARGE labeled for simplification.

- WHY MINE PARAPHRASES NOT SIMPLIFICATIONS? Progress in creating simplification models in multiple languages has been hindered by lack of labeled training data, i.e. pairs of complex sentences matched with their simplified forms. Moreover, simplifying a sentence can generally be achieved in multiple ways, which might depend on the target audience. For instance, a simplified sentence can be shorter than its original counterpart, be split into multiple sentences, use less complex vocabulary, include less detail, and so on. Simplification guidelines are therefore not uniquely defined, outlined by the stark differences between simplification datasets for English (Alva-Manchego et al., 2020a). Even if heuristics could be used to mine simplification corpora, these issues might prevent large-scale mining in multiple languages.[10]

- TRAINING SIMPLIFICATION MODELS ON PARAPHRASES To create models that can simplify after being trained on paraphrases, we leverage advancements in controllable text generation by using ACCESS (Martin et al., 2019) to control length, amount of paraphrasing, lexical complexity and syntactic complexity. ACCESS augments training data with *control markers* that are appended to the input side of supervised training data. During training time, the model learns to associate these control markers with properties of the output text, such as length. Thus, at inference time, the model can dynamically adjust the output form based on swaps of the control marker. We can then create a variety of different simplifications at inference time using the chosen controls.

- SEQUENCE EXTRACTION Simplification consists of multiple rewriting operations, some of which span over multiple sentences (e.g. sentence splitting or sentence fusion). To allow these types of operations to be represented in our data, we extract sequences of multiple sentences from documents (up to 300 characters). These sequences are further filtered to remove noisy text with too much punctuation, and low language model probability. In the following, these series of multiple consecutive sentences are termed *sequences*.

  We extract these sequences from CCNET (Wenzek et al., 2019). CCNET is an extraction of Common Crawl[11] (an open source snapshot of the web) that has been split into different languages using `fasttext` language identification (Joulin et al., 2017) and various language modeling filtering techniques to identify high quality, clean sentences. For English and French, we extract 1 billion sequences from CCNET. For Spanish we extract 650 millions sequences, the maximum for this language in CCNET after filtering

---

[10] We demonstrate subsequently in this chapter that mining simplifications does not achieve better results than more general paraphrase mining.

[11] `https://commoncrawl.org`

out noisy text.

► Creating a Sequence Index Using Embeddings To automatically mine our paraphrase corpora, we first compute n-dimensional sentence embeddings for each of our sequences using the LASER toolkit (Artetxe and Schwenk, 2019). LASER is a multilingual sentence embedding that maps sentences of similar meaning to the same location in the embedding space. We use `faiss` to create an index with all these sentence embeddings. Fully unsupervised sentence embeddings could also be used, even though they are often available only in English.

► Mining Paraphrases For each language, after the billion-scale index is created, we use those same 1 billion sequences as queries to identify potential paraphrases in the index. Each sequence is queried against the index and returns a set of top-k nearest neighbor sequences according to the semantic LASER embedding space using L2 distance[12]. We experimented with a range of k and ultimately set $k = 8$. These nearest neighbors are candidate paraphrases of the query sequence. We apply additional filters to remove poor quality alignments where the sequences are not paraphrases: for example when they are almost identical, when they are contained in one another, or when they were extracted from two consecutive and overlapping sliding windows of the same original document. Table 3.1 reports statistics of the mined corpora in English, French and Spanish.

Simplifying with ACCESS

We produce simplifications from models trained on paraphrases using ACCESS (Martin et al., 2020).

► Training with Control Tokens At training time, the model is provided with control tokens that give oracle information on the target sequence, such as the amount of compression of the target sequence relative to the source sequence (length control). For example, when the target sequence is 80% of the length of the original sequence, we provide the <NumChars_80%> control token. At inference time we can then control the generation by selecting a given target control value. We use the controls from Martin et al. (2020): number of characters ratio, Levenshtein similarity, aggregated word frequency ratio, and dependency tree depth ratio[13]. We describe these below:

- Number of Characters Ratio: character length ratio between source sentence and target sentence (compression level). This control token accounts for sentence compression, and content deletion. Previous work showed that simplicity is best correlated with length-based metrics, and especially in terms of number of characters. The number of characters indeed accounts for the lengths of words which is itself correlated to lexical complexity.
- Levenshtein Similarity: calculates the normalized character-level Levenshtein similarity between source and target. It quantifies the amount

[12] Based on empirical studies, we found L2 distance to work slightly better than similar distance measures. The original LASER work used cosine distance.

[13] This dependency tree was computed using the `spaCy` library.

of modification operated on the source sentence (through paraphrasing, adding and deleting content).

- Aggregated Word Frequency Ratio: as a proxy to lexical complexity, we compute a sentence-level measure by taking the third-quartile of log-ranks (inverse frequency order) of all words in a sentence. We subsequently divide the WordRank of the target by that of the source to get a ratio.
- Dependency Tree Depth Ratio: maximum depth of the dependency tree of the source divided by that of the target. This control token is designed to approximate syntactic complexity. Deeper dependency trees indicate dependencies that span longer and possibly more intricate sentences.

► SELECTING CONTROL VALUES AT INFERENCE Sentence simplification often depends on the context and target audience (Martin et al., 2020). Shorter sentences are more adapted to people with cognitive disabilities, while using more frequent words are useful to second language learners. Evaluation datasets reflect this plurality with different simplification typologies (Alva-Manchego et al., 2020a). Simplification systems, then, must adapt to different conditions. We follow the same approach as in Kumar et al. (2020) where for each evaluation dataset, a set of operation-specific weights of the proposed unsupervised simplification model is chosen. We adapt our model automatically by selecting the 4 control hyper-parameters of ACCESS with SARI on each validation set, similar to how (Surya et al., 2019) use SARI to select their best unsupervised simplification model. These 4 control hyper-parameters are intuitive and easy to interpret: when no validation set is available, they can also be set using prior knowledge on the task and still lead to solid performance[14].

The next step is to apply heuristics to only keep the subset of these paraphrases that are actual simplifications. To this end, we apply several heuristics to identify simplifications based on various possible types of rewriting a complex sentence into a simpler more readable one. For example, a long sentence can be split into several shorter sentences, or a its lexical complexity can be reduced by using simpler words.

More specifically, we create a simplicity score composed of two features.

The first feature assesses lexical simplification by comparing the 90% quantile of log ranks according to a frequency table of content words. We use the frequency table provided with FASTTEXT embeddings (Mikolov et al., 2018). Word frequencies have been found to be a strong indicator of lexical complexity (Paetzold and Specia, 2016a). Let w_i denote the words in the original o_{seq} and candidate simplified s_{seq} sequences, we compute the lexical score as:

$$s_1(o_{seq}, s_{seq}) = \frac{q_{90\%, w_i \in s_{seq}}\{w_i\}}{q_{90\%, w_i \in o_{seq}}\{w_i\}}$$

The second feature is a coarse proxy to structural simplification that compares the maximum length of a sentence in the complex sequence to

---

[14] Note, at an extreme, the various controls can be used to produce degenerate simplifications. For example, a long sentence could be controlled to simplify into one or two words. Of course, this would no longer preserve the meaning. Thus, as a starting point, we set the control tokens based on metrics computed on the validation set. This can also be computed on datasets for other languages, if a validation set is not present.

the maximum length of a sentence in the simple sequence because sentence-length has been shown to be one of the best indicator of overall simplicity (Martin et al., 2018). Let s_i be the sentences composing the original o_{seq} and simplified s_{seq} sequences, the structural score is:

$$s_2(o_{seq}, s_{seq}) = \frac{max_{s_i \in s_{seq}}\{len(s_i)\}}{max_{s_i \in s_{seq}}\{len(s_i)\}}$$

We then cast these features to a $[0, 1]$ range using a quantile transformer $qt()$ from scikit-learn (Pedregosa et al., 2011). We finally average them to get our final score:

$$s = \frac{qt(s_1)) + qt(s_2)}{2}$$

We only keep pairs where the average of those two features is below a certain threshold.

Leveraging Unsupervised Pretraining

Unsupervised pretraining has demonstrated large improvements on generative tasks, by using sequence-to-sequence models as denoising auto-encoders on large quantities of data (Lewis et al., 2020a; Liu et al., 2020b), training with noising functions such as span-based masking or shuffling sentence order[15]. We leverage these pretrained models to further improve our unsupervised approach to text simplification. For English, we finetune the pretrained generative model BART (Lewis et al., 2020a) on our newly created mined training corpora. BART is a pretrained sequence-to-sequence model that can be seen as a generalization of other recent pretrained models such as BERT (Devlin et al., 2019). For non-English, we use its multilingual generalization MBART (Liu et al., 2020b), which was pretrained on 25 languages.

3.2.4 *Experimental Setting*

We assess the performance of our approach on three languages: English, French, and Spanish. In all our experiments, we report scores on the test sets averaged over 5 random seeds with 95% confidence intervals.

Baselines

We report the results of various published systems as well as several additional baselines.

- **Identity:** The entire original sequence is kept unchanged and used as the simplification.
- **Truncation:** The original sequence is truncated to the first 80% words. It is a strong baseline in practice, as measured by standard simplification metrics.
- **Pivot:** We use machine translation to provide a baseline for languages for which no simplification corpus is available. The source non-English sentence is translated to English, simplified with our best supervised English simplification system, and then translated back into the source

[15] We refer the reader to a comprehensive survey on pretraining objectives and architectures here in the upcoming preprint *AMMUS: A Survey of Transformer-based Pretrained Models in Natural Language Processing*, found at `https://arxiv.org/abs/2108.05542`

	Data	ASSET		TURKCORPUS		NEWSELA	
Previous Work		SARI ↑	FKGL ↓	SARI ↑	FKGL ↓	SARI ↑	FKGL ↓
PBMT-R	PWKP (Wikipedia)	34.63	8.85	38.04	8.85	26.32	8.20
EditNTS	WIKILARGE/ NEWSELA	34.95	8.38	37.66	8.38	39.30	3.90
Dress-LS	WIKILARGE/ NEWSELA	36.59	7.66	36.97	7.66	38.00	4.90
DMASS-DCSS	WIKILARGE	38.67	7.73	39.92	7.73	—	—
ACCESS †	WIKILARGE	40.13	7.29	41.38	7.29	—	—
BTRLTS	UNSUPERVISED	33.95	7.59	33.09	8.39	37.22	3.80
UNTS	UNSUPERVISED	35.19	7.60	36.29	7.60	—	—
RM+EX+LS+RO	UNSUPERVISED	36.67	7.33	37.27	7.33	38.33	2.98
Baselines and Gold Reference							
Identity Baseline	—	20.73	10.02	26.29	10.02	12.24	8.82
Truncate Baseline	—	29.85	7.91	33.10	7.91	25.49	6.68
Gold Reference	—	44.87±0.36	6.49±0.15	40.04±0.30	8.77±0.08	—	—
Supervised Systems (This Work)							
Seq2Seq	WIKILARGE	32.71±1.55	8.62±0.34	35.79±0.89	8.63±0.34	22.23±1.99	8.00±0.26
MUSS (BART+ACCESS)	WIKILARGE	**43.63**±0.71	6.25±0.42	**42.62**±0.27	6.98±0.95	40.00±0.63	3.51±0.53
MUSS (BART+ACCESS)	WIKILARGE + MINED	**44.15**±0.56	6.05±0.51	**42.53**±0.36	7.60±1.06	39.08±1.80	3.96±1.71
MUSS (BART+ACCESS)	NEWSELA	42.91±0.58	6.91±0.42	41.53±0.36	7.39±0.42	**42.59**±1.00	2.74±0.98
MUSS (BART+ACCESS)	NEWSELA + MINED	41.36±0.48	6.96±0.26	40.01±0.51	8.26±0.36	**41.17**±0.95	2.70±1.00
Unsupervised Systems (This Work)							
Seq2Seq	MINED	38.03±0.63	9.41±0.07	38.06±0.47	9.43±0.07	30.36±0.71	8.85±0.13
MUSS (MBART+ACCESS)	MINED	41.11±0.70	7.18±0.21	39.40±0.54	8.65±0.40	34.76±0.96	5.44±0.25
MUSS (BART+ACCESS)	MINED	42.65±0.23	8.23±0.62	40.85±0.15	8.79±0.30	38.09±0.59	5.12±0.47

TABLE 3.2: **Unsupervised and Supervised Sentence Simplification for English.** We display SARI and FKGL on ASSET, TURKCORPUS and NEWSELA test sets for English. Best system SARI scores within confidence intervals are in bold.
† We compare to system outputs from the original ACCESS system (Martin et al., 2019).

language. For French and Spanish translation, we use CCMATRIX (Schwenk et al., 2019) to train Transformer models with LayerDrop (Fan et al., 2020a). We use the BART+ACCESS supervised model trained on MINED + WIKILARGE as the English simplification model. While pivoting creates potential errors, recent improvements of translation systems on high resource languages make this a strong baseline.

- **Gold Reference:** We report gold reference scores for ASSET and TURKCORPUS as multiple references are available. We compute reference scores in a leave-one-out scenario where each reference is evaluated against all others.[16]. The scores are then averaged over all references.
- **mBART v. BART:** Constructing simplification systems beyond English must utilize pretrained methods in multiple languages, either language-specific pretraining or multilingual pretrained models such as mBART. For English systems, they could also use either English BART or the multilingual mBART. We compare performance with both.

Evaluation Metrics

We evaluate with the standard metrics, SARI and FKGL.

- **SARI** Sentence simplification is commonly evaluated with SARI (Xu et al., 2016), which compares model generated simplifications with the source sequence and gold references. It averages F1 scores for addition, keep, and deletion operations. We compute SARI with the EASSE[17]

---

[16] To avoid creating a discrepancy in terms of number of references between the gold reference scores, where we leave one reference out, and when we evaluate the models with all references, we compensate by duplicating one of the other references at random so that the total number of references is unchanged.

[17] `https://github.com/feralvam/easse`

Table 3.3: **Examples of Generated Simplifications.** We show simplifications generated by our best unsupervised model: BART+ACCESS trained on mined data only. Bold highlights differences between original and simplified.

Original	**History** Landsberg prison, which is **in** the **town's western outskirts,** was **completed** in 1910.
Simplified	**The** Landsberg prison, which is **near** the **town,** was **built** in 1910.
Original	The name "hornet" is used for this and related species **primarily** because **of their habit of making aerial** nests **(similar to** the true hornets) rather than **subterranean nests.**
Simplified	The name "hornet" is used for this and related species because **they make** nests **in** the **air (like the** true hornets) rather than **in the ground.**
Original	Nocturnes is **an orchestral composition in three movements** by the French composer Claude Debussy.
Simplified	Nocturnes is **a piece of music for orchestra** by the French composer Claude Debussy.

Table 3.4: **Unsupervised Sentence Simplification in French and Spanish.** We display SARI on ALECTOR (French) and Newsela (Spanish) test sets. Best SARI scores within confidence intervals are in bold.
†mBART+ACCESS model.

Baselines	**Data**	**ALECTOR** (French) SARI ↑	**Newsela** (Spanish) SARI ↑
Identity	—	26.16	16.99
Truncate	—	33.44	27.34
Pivot	—	33.48$_{\pm 0.37}$	**36.19**$_{\pm 0.34}$
Unsupervised Systems (This Work)			
Seq2Seq	Mined	39.25$_{\pm 0.64}$	34.53$_{\pm 0.28}$
MUSS†	Mined	**41.73**$_{\pm 0.67}$	**35.67**$_{\pm 0.46}$

simplification evaluation suite (Alva-Manchego et al., 2019)[18].

- **FKGL** We report readability scores using the Flesch-Kincaid Grade Level (FKGL) (Kincaid et al., 1975), a linear combination of sentence lengths and word lengths. FKGL was designed to be used on English texts only, we do not report it on French and Spanish.

Training Data

For all languages we use the mined data described in Table 3.1 as training data. We show that training with additional labeled simplification data leads to even better performance in English. We use the two labeled datasets **WikiLarge** (Zhang and Lapata, 2017) and **Newsela** (Xu et al., 2015). WikiLarge is composed of 296k simplification pairs automatically aligned from English Wikipedia and Simple English Wikipedia. Newsela is a collection of news articles with professional simplifications, aligned into 94k simplification pairs by Zhang and Lapata (2017).

Evaluation Data

► English We evaluate our English models on **ASSET** (Alva-Manchego et al., 2020a), **TurkCorpus** (Xu et al., 2016) and **Newsela** (Xu et al., 2015). TurkCorpus and ASSET were created using the same 2000 valid and 359 test source sentences. TurkCorpus contains 8 reference simplifications per source sentence and ASSET contains 10 references per source. ASSET is a generalization of TurkCorpus with a more varied set of rewriting operations, and considered simpler by human judges (Alva-Manchego et al., 2020a). For

[18] We use the latest version of SARI implemented in `EASSE` (Alva-Manchego et al., 2019) which fixes bugs and inconsistencies from the traditional implementation. We thus recompute scores from previous systems that we compare to, by using the system predictions provided by the respective authors available in `EASSE`.

NEWSELA, we evaluate on the split from (Zhang and Lapata, 2017), which includes 1129 validation and 1077 test sentence pairs.

- FRENCH For French, we use the **ALECTOR** dataset (Gala et al., 2020) for evaluation. ALECTOR is a collection of literary (tales, stories) and scientific (documentary) texts along with their manual document-level simplified versions. These documents were extracted from material available to French primary school pupils. We split the dataset in 450 validation and 416 test sentence pairs.

- SPANISH For Spanish we use the **Spanish part of NEWSELA** (Xu et al., 2015). We use the alignments from (Aprosio et al., 2019), composed of 2794 validation and 2795 test sentence pairs.

3.2.5 *Results*

We now assess the quality of our mined data and the improvements brought up by unsupervised pretraining for simplifying English, French, and Spanish.

English Simplification

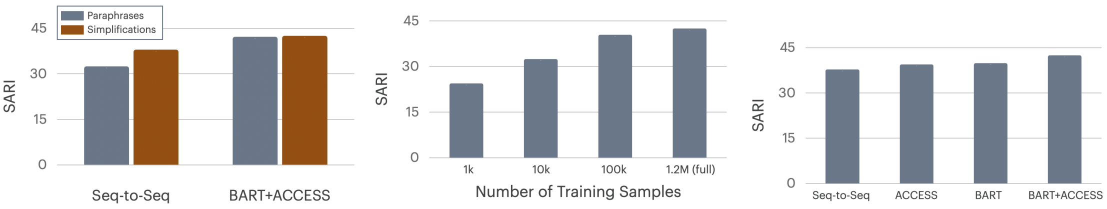
We compare models trained on our mined corpus (see Table 3.1) with models trained on the labeled parallel datasets WIKILARGE and NEWSELA. We also compare to other state-of-the-art supervised models: PBMT-R (Wubben et al., 2012), Dress-LS (Zhang and Lapata, 2017), DMASS-DCSS (Zhao et al., 2018c), ACCESS (Martin et al., 2020); and the unsupervised models: UNTS (Surya et al., 2019), BTRLTS (Zhao et al., 2020b), and RM+EX+LS+RO (Kumar et al., 2020). We present our results in Table 3.2.

- SEQ2SEQ MODELS ON MINED DATA When training a Transformer sequence-to-sequence model (Seq2Seq) on WIKILARGE compared to the mined corpus, models trained on the mined data perform better across all test sets. It is surprising that a model trained solely on paraphrases achieves such good results on simplification benchmarks. Previous works have shown that simplification models suffer from not making enough modifications to the source sentence and found that forcing models to rewrite the input was beneficial (Wubben et al., 2012; Martin et al., 2020). This could explain the good performance of a model trained at paraphrasing.

- ADDING BART AND ACCESS We use our mined data to finetune BART and add the simplification-based generative control from ACCESS to achieve state-of-the-art results.

With no labeled simplification data, we achieve a +2.52 SARI improvement over the previous state of the art on ASSET. We achieve strong results on TURKCORPUS with 40.85 SARI with no supervision and closely match the previous supervised state of the art (41.38 SARI, best seed selected on the validation set by the authors). Similarly, on NEWSELA we closely match the best supervised results (39.30) with our unsupervised model (38.09).

When incorporating labeled data from WIKILARGE and NEWSELA, we obtain state-of-the-art results on all datasets. In our experiments, using



(a) **Simplifications vs. Paraphrases** (b) **Large-Scale Mining** (c) **BART and ACCESS**

FIGURE 3.2: **Ablations** We display averaged SARI scores on the English ASSET test set with 95% confidence intervals (5 runs). (a) Models trained on mined simplifications or mined paraphrases, (b) BART+ACCESS trained on varying amounts of mined data, (c) Models trained with or without BART and/or ACCESS.

labeled data along with mined data does not always help compared to training only with labeled data, especially with the Newsela training set. Newsela is already a high quality dataset focused on the specific domain of news articles. It might not benefit from additional lesser quality mined data.

► Examples of Simplifications Various examples from our unsupervised system are shown in Table 3.3. Examining the simplifications, we see reduced sentence length, sentence splitting, and simpler vocabulary usage. For example, the words *in the town's western outskirts* is changed into *near the town* and *aerial nests* is simplified into *nests in the air*.

French and Spanish Simplification

Our unsupervised approach to simplification can be applied to many language. As for English, we first create a corpus of paraphrases composed of 1.4 million sequence pairs in French and 1.0 million sequence pairs in Spanish (Table 3.1). We evaluate the quality of our mined corpus in Table 3.4. Unlike English, where labeled parallel training data has been created using Simple English Wikipedia, no such datasets exist for French or Spanish. We compare to several baselines, namely the identity, truncation and pivot baselines.

► Using Seq2Seq Models on Mined data Compared to our baselines, training a Transformer sequence-to-sequence model on our mined data achieves stronger results in French and stronger results in Spanish except for the pivot baseline.

► Adding mBART and ACCESS To incorporate multilingual pretraining, we use mBART. mBART was trained on 25 languages compared to only English for BART. Similar to what we observed in English, we achieve the best results by combining mBART+ACCESS, and training on mined data. It outperforms our strongest baseline by +8.25 SARI in French. In Spanish it matches the pivot baseline performance. As shown in the English results in Table 3.2, mBART has a small loss in performance of 1.54 SARI compared to its monolingual English counterpart BART, probably due to the fact that it handles 25 languages instead of one. Using monolingual BART trained for French or Spanish would perform even better. The pivot baseline also uses a supervised English simplification model (BART+ACCESS on Mined + WikiLarge), compared to our unsupervised Spanish model.

	ASSET (English)			ALECTOR (French)			NEWSELA (Spanish)		
	Adequacy	Fluency	Simplicity	Ade.	Flu.	Simp.	Ade.	Flu.	Simp.
ACCESS (Martin et al., 2020)	3.1	3.5	1.4	—	—	—	—	—	—
Pivot baseline	—	—	—	1.8	0.5	0.3	0.3	**3.5**	2.2
Gold Reference	**3.4**	3.8	1.8	**3.5**	**3.9**	**1.7**	2.2†	3.4	1.3†
(M)BART+ACCESS (unsup.)	3.2	3.8	1.9	2.9	3.5	1.2	**2.3**	**3.5**	**2.6**
BART+ACCESS (sup.)	3.2	**3.9**	**2.2**	—	—	—	—	—	—

TABLE 3.5: **Human Evaluation** We display human ratings of adequacy, fluency and simplicity for previous work ACCESS, pivot baseline, reference human simplifications, our best unsupervised systems (BART+ACCESS for English, MBART+ACCESS for other languages), and our best supervised model for English. Scores are averaged over 50 ratings per system with 95% confidence intervals. †Low ratings of the gold reference in Spanish NEWSELA is due to automatic alignment errors.

Human Evaluation

To validate the quality of our models, we conduct a human evaluation in all languages according to adequacy, fluency, and simplicity and report the results in Table 3.5.

▸ HUMAN RATINGS For human evaluation, we recruit volunteer native speakers for each language (5 in English, 2 in French, and 2 in Spanish). We evaluate three linguistic aspects on a 5 point Likert scale (0-4): adequacy (*is the meaning preserved?*), fluency (*is the simplification fluent?*) and simplicity (*is the simplification actually simpler?*). For each system and each language, 50 simplifications are annotated and each simplification is rated once only by a single annotator.

Table 3.5 displays the average ratings. Human judgments confirm that our unsupervised and supervised BART+ACCESS models are more fluent and produce simpler outputs than previous state-of-the-art (Martin et al., 2019). They are deemed as fluent and simpler than the human simplifications from ASSET test set, which indicates our model is able to reach a high level of simplicity thanks to the control mechanism. In French and Spanish, our unsupervised model MBART+ACCESS performs better or similar in all aspects than the strong supervised pivot baseline which has been trained on labeled English simplifications. In Spanish the gold reference surprisingly obtains poor human ratings, which we found to be caused by errors in the automatic alignments of source sentences with simplified sentences of the same article.

▸ ANNOTATOR AGREEMENT We rate a subset of 50 simplifications twice in order to compute an inter-annotator agreement. The Cohen's Kappa is 0.21 for adequacy, 0.27 for fluency, and 0.22 for simplicity. This inter-annotator agreement highlights the fact that the notion of simplicity is subjective and that annotators do not always agree.

Ablations

▸ MINING SIMPLIFICATIONS VS. PARAPHRASES In this work, we mined paraphrases to train simplification models. We also compared to directly mining simplifications using simplification heuristics. To mine a simplification dataset, we followed the same paraphrase mining procedure of querying 1 billion sequences on an index of 1 billion sequences. Out of the resulting paraphrases, we kept only pairs that either contained sentence splits, reduced sequence length, or simpler vocabulary. We removed the paraphrase constraint that enforced sentences to be different enough as we found these were no longer necessary. We tuned these heuristics to optimize SARI on the validation set. The resulting dataset has 2.7 million simplification pairs. In Figure 3.2(a), we show that sequence-to-sequence models trained on paraphrases achieve better performance. A similar trend exists with BART and ACCESS, justifying the simpler approach of mining paraphrases instead of simplifications.

- HOW MUCH MINED DATA DO YOU NEED? We investigate the importance of a scalable mining approach that can create million-sized training corpora for sentence simplification. In Figure 3.2(b), we analyze the performance of training our best model on English on different amounts of mined data. By increasing the number of mined pairs, SARI drastically improves, indicating that efficient mining at scale is critical to performance. Unlike human-created training sets, unsupervised mining allows for large datasets in multiple languages.

- IMPROVEMENTS FROM PRETRAINING AND CONTROL We compare the respective influence of pretraining BART and controllable generation ACCESS in Figure 3.2(c). While both BART and ACCESS bring improvement over standard sequence-to-sequence, they work best in combination. Unlike previous approaches to text simplification, we use pretraining to train our simplification systems. We find that the main qualitative improvement from pretraining is increased fluency and meaning preservation. While models trained on mined data see several million sentences, pretraining methods are typically trained on billions. Combining pretraining with controllable simplification enhances simplification performance by flexibly adjusting the type of simplification.

3.2.6 *Conclusion*

We propose a sentence simplification approach that does not rely on labeled parallel simplification data thanks to controllable generation, pretraining and large scale mining of paraphrases from the web. This approach is language agnostic and matches or outperforms previous state-of-the-art results, even from supervised systems that use labeled simplification data, on three languages: English, French, and Spanish. In future work, we plan to investigate how to scale this approach to more languages and types of simplifications.

4

Meaning Representation-to-Text Generation

> *"There is no such thing as information overload. There is only bad design."*
> —Edward Tufte

▸ Synopsis The challenge of text-to-text generation methods is to produce natural language output based on information contained in textual input. However, a wide variety of information exists in various *structured* forms — such as tables. In this chapter, we focus on structured inputs for text generation tasks and discuss its motivations and unique challenges, then focus on a specific type of structured input: abstract meaning representations.

4.1 STRUCTURED INPUT TO TEXT GENERATION

Numerous applications of natural language generation systems utilize textual input to produce textual output. However, structured data is plentiful around us. For example, information does not exist only in a free-text form on Wikipedia, but is also stored in knowledge bases such as Wikidata and DBPedia. Knowledge is often more succinctly represented in the form of tables or graphs. These forms of structured information can be verbalized as natural language, which is often referred to as *Structured Input-to-Text Generation*. In this chapter, we focus on a particular form of structured input, an *Abstract Meaning Representation*. We detail the definition of Abstract Meaning Representations (AMRs), their construction, and their motivation in the subsequent sections.

4.2 ABSTRACT MEANING REPRESENTATIONS

Abstract Meaning Representations (AMRs) are a type of semantic meaning representation, first introduced in Banarescu et al. (2013). At a high level, AMRs represent sentences as single-root, directed acyclic graphs. This formalism lends AMRs three main advantages as semantic representations — they are easy to read and have a standard methodology for evaluation. Overall, AMRs are used to represent sentences in a form that can abstract away from morphological and syntactic variability.

▸ Definition AMRs are defined with a unique root, corresponding to the top node of the tree structure. Each node in the graph has a variable associated with it, labeled with a concept. Each edge represents a relationship between two nodes. For example, the sentence *the puppy is eating a bone* is represented in AMR schema in Figure 4.1.

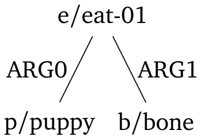


FIGURE 4.1: Example AMR representation of the sentence *The puppy is eating a bone*, where *p/puppy* indicates that *p* is an instance of *puppy*.

► **INFORMATION LOSS IN AMRS** We note that AMRs are not meant to fully represent all information contained within a natural language sentence. For example, AMR schema drops aspects of plurality, articles, and tense. To illustrate, the following phrases: *A puppy eating, The puppy eats, puppies ate, the puppies will eat* are all mapped to p/puppy e/eat-01. The removal of information is driven by various motivations. Tense, for instance, is traditionally removed as English verbal tense does not generalize across languages, and AMR is designed to be a crosslingual representation.

While this information loss from the original natural language into structured information is a limitation, it is also an advantage. This advantage stems from the fact that AMRs project variable forms of the same sentence onto a consistent, simple representation. For example, the following four very different sentence structures:

- The man described the mission as a disaster.
- The man's description of the mission: disaster.
- As the man described it, the mission was a disaster.
- The man described the mission as disastrous.

map to the same, simple, high level AMR representation[1]:

d/describe -01
:ARG0 (m/man)
:ARG1 (m2/mission)
:ARG2 (d/disaster)

This mapping distils the core conceptual information from many variable sentences into one concise representation of fundamental semantic meaning. This allows AMRs to differentiate semantic meaning from the surface form of the sentence.

► **AMR ANNOTATION**

Since AMRs were initially proposed, a variety of data has been annotated into gold-standard AMRs. For example, the Linguistic Data Consortium (LDC) has released several annotated AMR datasets.[2] Various research efforts have also centered around automatic AMR Parsing, with the goal of representing sentences as their AMR forms.[3] Based on this interest in AMR annotation and AMR Parsing, various corpora — in English and beyond — exist in AMR form, as a basis for AMR-to-Text generation tasks.

4.2.1 *Challenges*

While AMRs are an attractive, high-level representation of semantic meaning, there are challenges in generation of natural language from AMR. We describe two major challenges facing AMR-to-Text Generation systems before discussing our solutions in the next section.

[1] We refer the reader to the excellent tutorial from Nathan Schneider, Tim O'Gorman, and Jeffrey Flanigan: `https://github.com/nschneid/amr-tutorial`, from which this example is borrowed

[2] `https://catalog.ldc.upenn.edu/LDC2020T02`

[3] We refer the reader to Pust et al. (2015) for an example of recent work in AMR parsing.

- BRIDGING STRUCTURE AND STRING

  Unlike Text-to-Text generation tasks, where the input and output are both natural language, Structured Input-to-Text tasks face the fundamental challenge of interpreting the structure. This broadly challenges all structured input systems, and is not a specific limitation of AMR. However, systems hoping to handle structured information must face the gap between structure and string. For example, ingesting structured input could necessitate the usage of specialized encoder neural networks adapted to read graphs or trees, rather than standard Seq-to-Seq models.

- UNDERSPECIFICATION

  Further, structured representations such as AMRs often lack information. As discussed earlier in this chapter, AMRs in particular do not capture information about tense and plurality. Contrast this with a task such as sentence simplification, where the entirety of the context required to simplify a sentence is (usually) fully contained in the original complex sentence. In comparison, AMR-to-Text generation systems must not only generate natural language based on the structured input, but also *fill in* the missing information. From this perspective, AMR-to-Text generation is a fundamentally different task than the Text-to-Text models discussed in the previous chapter.

4.3 MULTILINGUAL AMR-TO-TEXT GENERATION

In this section, we propose a system for multilingual AMR-to-Text Generation, verbalizing a standard English AMR into 21 different languages. AMR is meant to be a less English-centric representation of semantic meaning and general enough to be verbalized into multiple different languages — the capability of which we investigate in this section. As this is the first instantiation of multilingual AMR generation, we describe how we created the training and evaluation data for such a task in addition to the multilingual model for the task. We end with an analysis of the performance of this system for various languages and interesting phenomena observed in AMR-to-Text generation, as well as an extensive human evaluation.

4.3.1 *Introduction*

Generating text from structured data has a variety of applications in natural language processing. Tasks such as decoding from tables (Lebret et al., 2016; Sha et al., 2018), question answering from knowledge bases (Fan et al., 2019a), and generation from Resource Description Framework (RDF) Triples (Gardent et al., 2017), knowledge graphs (Marcheggiani and Perez-Beltrachini, 2018) and linguistic meaning representations (Konstas et al., 2017) face similar challenges: interpreting structured input and writing fluent output. We focus on generating from graph structures in the form of Abstract Meaning Representations (AMR) (Banarescu et al., 2013). Previous work has largely focused on generating from AMR into English, but we propose a multilingual approach that can decode into twenty one different languages.

Compared to multilingual translation, decoding from structured input has distinct challenges. Translation models take natural language input and must faithfully decode into natural language output. However, as shown in Zhao et al. (2020a), bridging the gap between structured input and linear output is a difficult task. In addition, in structured input such as graphs, the input is usually semantically under-specified. For example, in AMRs, function words are missing and tense and number are not given. Thus, generation from structured input must bridge the gap between (i) structure and string and (ii) underspecified input and fully specified output. Multilinguality brings a third challenge — that of generating in languages that have varied morphological and word order properties.

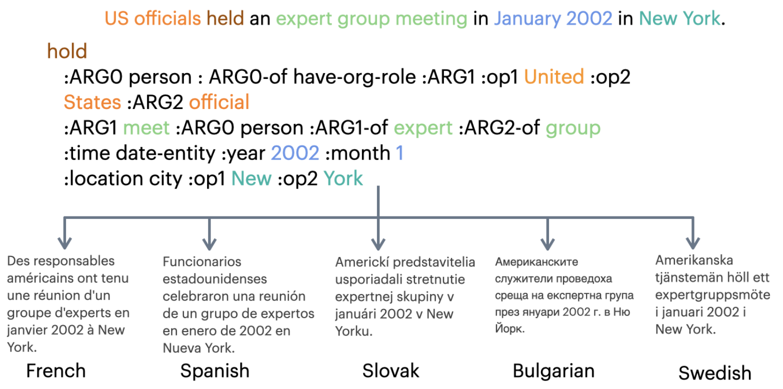
Annotating natural language with AMR is a complex task and training datasets only exist for English,[4] so previous work on AMR-to-text generation has overwhelmingly focused on English. We create training data for multilingual AMR-to-Text models, by taking the EUROPARL multilingual corpus and automatically annotating the English data with AMRs using the `jamr` semantic parser. We then use the English AMRs as the input for all generation tasks. To improve quality, we leverage recent advances in natural language processing such as cross-lingual embeddings, pretraining and multilingual learning. Cross-lingual embeddings have shown striking improvements on a range of cross-lingual natural language understanding tasks (Devlin et al., 2019; Conneau et al., 2019; Wu and Dredze, 2019; Pires et al., 2019). Other work has shown that the pre-training and fine-tuning approaches also help improve generation performance (Dong et al., 2019; Song et al., 2019; Lawrence et al., 2019; Rothe et al., 2019). Finally, multilingual models, where a single model is trained to translate from multiple source languages into multiple target languages, are achieving increasingly better results in machine translation (Johnson et al., 2017; Firat et al., 2017; Aharoni et al., 2019; Arivazhagan et al., 2019).

By combining these techniques, we demonstrate that fluent generation is possible for multilingual AMR-to-Text models. We use automatic and human evaluation to assess performance on (1) EUROPARL data with silver English-centric AMR as input and the 21 EUROPARL languages as target and (2) on LDC2015E86 data with gold English-centric AMR as input and English, Spanish, Italian, and German as target. Our results demonstrate, for the first time, that it is possible to generate from AMRs into multiple languages. We show that multilingual models have strong performance compared to single-language baselines and produce fluent output, based on the judgments of native speakers. We further investigate how factors, such as differences in the size of the training data, differences in language word order and morphological properties, and differences in the set of languages used for training many-to-one models, impact results. We will make code and models available, to aid research in multilingual AMR-to-Text Natural Language Generation.

4.3.2 *Related Work*

---

[4] AMR datasets from the LDC can be found at https://amr.isi.edu/download.html



US officials held an expert group meeting in January 2002 in New York.

```
hold
  :ARG0 person : ARG0-of have-org-role :ARG1 :op1 United :op2
  States :ARG2 official
  :ARG1 meet :ARG0 person :ARG1-of expert :ARG2-of group
  :time date-entity :year 2002 :month 1
  :location city :op1 New :op2 York
```

French: Des responsables américains ont tenu une réunion d'un groupe d'experts en janvier 2002 à New York.

Spanish: Funcionarios estadounidenses celebraron una reunión de un grupo de expertos en enero de 2002 en Nueva York.

Slovak: Americkí predstavitelia usporiadali stretnutie expertnej skupiny v januári 2002 v New Yorku.

Bulgarian: Американските служители проведоха среща на експертна група през януари 2002 г. в Ню Йорк.

Swedish: Amerikanska tjänstemän höll ett expertgruppsmöte i januari 2002 i New York.

FIGURE 4.2: **Generating into Multiple Languages from English AMR.** We depict our proposed multilingual AMR to text generation task, displaying a single English AMR as input with multiple different possible natural language outputs. The same system would be able to produce text in multiple languages.

▸ AMR-TO-TEXT GENERATION. Initial work on AMR-to-text generation adapted methods from statistical machine translation (MT) (Pourdamghani et al., 2016), grammar-based generation (Mille et al., 2017), tree-to-string transducers (Flanigan et al., 2016), and inverted semantic parsing (Lampouras and Vlachos, 2017). Neural approaches explored sequence-to-sequence models where the AMR is linearized (Konstas et al., 2017) or modeled with a graph encoder (Marcheggiani and Perez-Beltrachini, 2018; Damonte and Cohen, 2019; Ribeiro et al., 2019; Song et al., 2018; Zhu et al., 2019). As professionally-annotated AMR datasets are in English, all this work focuses on English.

One exception is the work of Sobrevilla Cabezudo et al. (2019) which uses automatic translation to translate the English text of the LDC AMR data into Brazilian Portuguese and align English with the Portuguese translation to create Portuguese-centric AMRs. However, this work focuses only on one language. In contrast, we consider generation into twenty one languages. We use very different methods and generate from English-centric AMRs, not target-language AMRs.

▸ MULTILINGUAL MR-TO-TEXT GENERATION. While work on AMR-to-Text generation has mostly focused on generation into English, the Multilingual Surface Realization shared tasks (Mille et al., 2018, 2019) have made parallel MR/Text datasets available for 11 languages. Two tracks are proposed: a shallow track where the input is an unordered, lemmatized dependency tree and a deep track where the dependency tree edges are labelled with semantic rather than syntactic relations and where function words have been removed.

The participants approaches to this multilingual generation task use gold training data and mostly focus on the shallow track where the input is an unordered lemmatized dependency tree and the generation task reduces to linearization and morphological realization. The models proposed are pipelines that model each of these subtasks and separate models are trained for each target language (Kovács et al., 2019; Yu et al., 2019; Shimorina and Gardent, 2019a,b; Castro Ferreira and Krahmer, 2019). In this work, we focus instead on more abstract, deeper, input (AMRs) and propose end-to-end, multilingual models for all target languages.

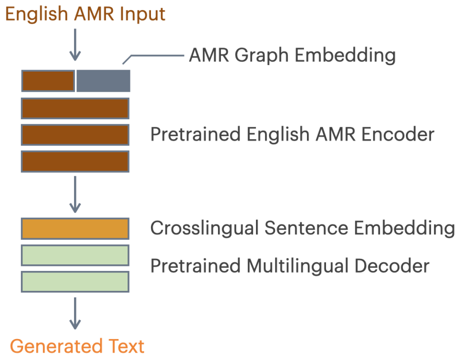

English AMR Input
AMR Graph Embedding
Pretrained English AMR Encoder
Crosslingual Sentence Embedding
Pretrained Multilingual Decoder
Generated Text

FIGURE 4.3: **One-to-Many Architecture for Multilingual AMR-to-Text Generation.** The English-centric AMR input is linearized and modeled with graph embeddings with a pre-trained Transformer Encoder. Text is generated with a pre-trained Transformer Decoder initialized with cross-lingual embeddings.

4.3.3 *Method*

To generate from AMRs, we use neural sequence to sequence models that model the input AMR with a Transformer Encoder and generate natural language with a Transformer Decoder. For all languages, the input is an English-centric AMR that was derived automatically using the `jamr` semantic parser from English text. We pre-train both the AMR encoder and the multilingual decoder and we leverage crosslingual embeddings.

Encoding English AMR

The AMR graph is first linearized into a sequence of tokens as shown in Figure 4.2 after preprocessing following (Konstas et al., 2017) (see Section 4.1 for a detailed description). Rather than model the graph structure directly, following Fan et al. (2019a), we model the graph using a *graph embedding*. The graph embedding provides additional information to the Transformer Encoder by encoding the depth of each node in the rooted graph and the subgraph each node belongs to. Concretely, each token has a word and position embedding, and additionally an indicator of depth calculated from the root and an indicator of which subtree the node belongs to (with all subtrees stemming from the root). These additional embeddings are concatenated to the word and position embeddings. Such information allows the Transformer Encoder to capture some graph structure information, while still modeling a sequence. This is depicted in Figure 4.3.

To create a one-to-many multilingual model, we model a *language embedding* on the encoder side to allow the decoder to distinguish which language to generate into. This technique has been previously used in multilingual translation (Arivazhagan et al., 2019). The English AMR begins with a token that indicates the decoder side language.

To improve the quality of the encoder, we incorporate large-scale pretraining on millions of sequences of AMR by adopting the generative pretraining approach proposed in Lewis et al. (2020a). This pretraining incorporates various noise operations, such as masking (Devlin et al., 2019), span masking (Fan et al., 2019a), and shuffling. Previous work has shown that pretraining is effective for providing neural models with additional information about the structure of natural language and improving model quality (Dong et al., 2019; Song et al., 2019; Lawrence et al., 2019). As models increase in size, smaller training datasets (such as human-annotated AMR) are often not large enough to fully train these models. The entire encoder is pretrained on silver AMRs, as shown in Figure 4.3.

Multilingual Decoding from AMR

The Transformer Decoder attends to the encoded English AMR, a graph of concepts and relations, and generates text into many different languages with varied word order and morphology.

As displayed in Figure 4.3, we use both language model pretraining and crosslingual embeddings to improve decoder quality. Monolingual data from various languages is used to pretrain each language model. Further, we incorporate crosslingual embeddings. These embeddings aim to learn universal representations that encode sentences into shared embedding spaces. Various recent work in crosslingual embeddings (Conneau and Lample, 2019) show strong performance on other multilingual tasks, such as XNLI (Conneau et al., 2018), XTREME (Hu et al., 2020), and MLQA (Lewis et al., 2019). We use the embeddings from XLM (Conneau and Lample, 2019) to initialize the multilingual embeddings of our decoder.

Model Training

To train our one-to-many multilingual AMR-to-text generation model, we use pairs of English AMR and text in multiple different languages. The English AMR does not need to be aligned to sentences in multiple languages. Instead, we create one AMR-to-text corpus for each language and concatenate all of them for training a multilingual model. During the training process, the pretrained AMR encoder and pretrained crosslingual decoder are finetuned on our multilingual AMR-to-text training corpus.

4.3.4 *Experimental Setting*

We describe the various sources of data used to create multilingual AMR-to-text generation models and describe the implementation and evaluation.

Data

- PRETRAINING. For encoder pretraining on silver AMR, we take thirty million sentences from the English portion of CCNET[5] (Wenzek et al., 2019), a cleaned version of Common Crawl (an open source version of the web). We use `jamr`[6] to parse English sentences into AMR. For multilingual decoder pretraining, we take thirty million sentences from each language split of CCNET.

- MULTILINGUAL DATA. We use EUROPARL, an aligned corpus of European Union parliamentary debates. Each language in EUROPARL is aligned to English. We study the twenty one languages available in EUROPARL: Bulgarian, Czech, Danish, Dutch, English, German, Greek, Spanish, Estonian, Finnish, French, Hungarian, Italian, Lithuanian, Latvian, Polish, Portuguese, Romanian, Slovak, Slovenian, and Swedish. The earliest releases in EUROPARL

[5] `https://github.com/facebookresearch/cc_net`

[6] `https://github.com/jflanigan/jamr`

Model Amount of Data	**en** 8.2M	**da** 1.9M	**de** 1.9M	**el** 1.2M	**es** 1.9M	**fi** 1.9M	**fr** 2M	**it** 1.9M	**nl** 1.9M	**pt** 1.9M	**sv** 1.9M
Machine Translation	—	—	**17.8**	—	24.9	—	**20.7**	18.6	19.4	21.0	19.2
English AMR-XX	**34.2**	21.3	16.9	14.2	24.3	12.9	20.5	**19.1**	18.8	20.4	18.6
Multilingual	32.5	21.2	17.0	13.8	24.2	12.4	19.7	17.8	18.5	20.5	18.7
+ Graph Embedding	32.9	21.4	17.0	14.0	24.3	12.5	19.9	18.0	18.6	20.7	18.9
+ Crosslingual Embedding	33.0	21.7	17.3	14.4	24.7	12.9	19.9	18.5	19.0	21.0	19.0
+ Encoder Pretraining	33.4	21.7	17.3	14.5	24.9	13.0	20.2	18.7	19.1	21.0	19.1
+ Decoder Pretraining	33.8	**21.9**	17.5	**14.6**	**25.1**	**13.4**	20.3	18.9	**19.4**	**21.2**	**19.5**

Model Amount Data	**bg** 400K	**cs** 650K	**et** 650K	**hu** 620K	**lt** 630K	**lv** 640K	**pl** 630K	**ro** 400K	**sl** 640K	**sk** 620K
English AMR-XX	33.8	27.5	18.9	23.1	23.9	25.4	23.4	30.6	30.1	28.7
Multilingual	34.6	28.4	19.1	23.8	24.4	26.9	23.4	31.5	30.6	29.7
+ Graph Embedding	34.7	28.5	19.3	23.9	24.5	27.0	23.6	31.5	20.7	29.9
+ Crosslingual Embedding	35.0	28.9	19.7	24.3	24.8	27.4	24.0	31.7	30.8	30.1
+ Encoder Pretraining	35.2	29.0	19.8	24.5	25.0	27.5	24.1	31.9	31.0	30.2
+ Decoder Pretraining	**35.7**	**29.5**	**21.2**	**24.7**	**25.5**	**27.9**	**24.4**	**32.1**	**31.4**	**30.6**

TABLE 4.1: **Results on 21 Languages in Europarl**. The English-XX baseline (generation into a single language) combines all modeling improvements. When training on multiple seeds, the standard deviation is around 0.1 to 0.3 BLEU, making the difference between the multilingual baseline and the addition of our modeling improvements statistically significant.

were prepared with a fixed common testing set across all languages, but later releases in ten new languages do not have a validation or test set. Thus, for the languages where the standard split is applicable, we report results on the common testing set, splitting it in half for validation and testing. For languages where there is no evaluation set, we take a part of the training set and reserve it for validation and another portion for testing. We use `jamr` to parse the English text of the Europarl corpus into AMRs. This creates a corpus of automatically created silver English AMRs aligned with sentences in twenty one European languages.

- GOLD AMR. We also evaluate our models (trained on silver AMRs) on gold AMR where available. For this, we use the CROSSLINGUAL AMR dataset from Damonte and Cohen (2018).[7] The corpus was constructed by having professional translators translate the English text of the LDC2015E86 test set into Spanish, Italian, German, and Chinese. We only evaluate on languages where we have training data from EUROPARL (i.e. we do not include Chinese as it is not in EUROPARL).

- PREPROCESSING. All data remains untokenized and cased. For AMR, we follow Konstas et al. (2017) in processing the `jamr` output into a simpler form. We remove variable names and instance-of relation (/) before every concept. However, we do not anonymize entities or dates, as improvements in modeling have allowed for better representations of rare words such as entities. We learn a sentencepiece model with 32K operations to split the English AMR into subword units. On the decoder side, we apply the sentencepiece model and vocabulary of XLM (Conneau and Lample, 2019). We choose to use the existing XLM sentencepiece and vocabulary so that the XLM cross-lingual embeddings can be used to initialize our models. For the encoder, we do not use existing vocabularies, as they do not capture the AMR graph structure.

[7] The data will be available in a subsequent release from LDC. To evaluate on this data before then, please contact Damonte and Cohen (2018).

Models

We implement our models in `fairseq-py` (Ott et al., 2019). We use large Transformer (Vaswani et al., 2017) sequence-to-sequence models and train all models for 50 epochs with LayerDrop (Fan et al., 2020a), which takes around 2 days. We initialize all weights with the pretrained models. When combining crosslingual word embeddings and encoder and decoder pretraining, we initialize all weights with pretraining, then use crosslingual word embeddings. We do not perform extensive hyperparameter search, but experimented with various learning rate values to maintain stable training with pretrained initialization. To generate, we decode with beam search with beam size 5. Our pretrained models are available for download.[8]

4.3.5 *Monolingual and Translation Baselines*

We compare our multilingual models both to monolingual models (one model trained for each language) and to a hybrid NLG/MT baseline. For the latter, we first generate with the AMR-to-English model and then translate the generation output to the target language using MT. Our translation models are Transformer Big models trained with LayerDrop (Fan et al., 2020a) for 100k updates on public benchmark data from WMT where available and supplemented with mined data from the `ccMatrix` project (Schwenk et al., 2019). We trained translation models for languages where large quantities of aligned bitext data are readily available, and cover a variety of languages.

Model	**en**	**es**	**it**	**de**
Konstas et al. (2017)	22.0	—	—	—
Song et al. (2018)	23.3	—	—	—
Cao et al. (2019)	23.5	—	—	—
Damonte et al. (2019)	24.4	—	—	—
Guo et al. (2019)	25.7	—	—	—
Ribeiro et al. (2019)	24.3	—	—	—
Zhu et al. (2019)	**29.7**	—	—	—
Machine Translation	—	21.6	19.6	**15.7**
English-XX Seq2Seq	25.2	21.1	**19.8**	14.9
Multilingual Seq2Seq	24.2	21.0	19.0	14.7
+ Graph Attribute	24.5	21.0	19.2	14.8
+ Crosslingual Embed	24.6	21.3	19.4	15.1
+ AMR Enc Pretrain	24.7	21.5	19.6	15.1
+ Multiling Dec Pretrain	24.9	**21.7**	**19.8**	15.3
+ Finetune on Gold AMR	26.3	—	—	—

TABLE 4.2: **Results on Gold AMR from LDC2015E86**. We display the BLEU scores on the test set with Gold AMR input, for four supported languages: English, Spanish, Italian, and German. We compare to various published systems as well as our created machine translation baseline.

Evaluation

We evaluate with detokenized BLEU using `sacrebleu` (Post, 2018). We conduct human evaluation by asking native speakers to evaluate word order, morphology, semantic faithfulness (with respect to the reference) and paraphrasing (how much the generation differs from the reference) on a 3 point scale. The evaluation was done online. For each language, evaluators annotated 25 test set sentences with high BLEU score and 25 sentences with low BLEU score. We removed sentences that were shorter than 5 words.

---

[8] `https://github.com/facebookresearch/m-amr2text`

Language	Number of Evaluators	Morphology	Word Order	Semantic Accuracy	Good Paraphrases	Std Dev Morphology	Std Dev Word Order
English	7	2.9	2.9	2.4	84%	0.06	0.04
Danish	2	2.9	2.9	2.3	88%	0.09	0.04
German	4	3.0	2.9	2.2	75%	0.02	0.06
Greek	5	2.9	2.9	2.2	75%	0.06	0.04
Spanish	10	2.9	2.9	2.2	81%	0.09	0.07
Finnish	2	2.9	3.0	2.1	69%	0.01	0.00
French	7	3.0	3.0	2.3	81%	0.02	0.03
Italian	5	3.0	3.0	2.3	82%	0.04	0.05
Dutch	7	2.9	2.9	1.9	60%	0.06	0.06
Portuguese	7	2.9	2.9	2.4	83%	0.08	0.06
Swedish	5	2.9	2.9	2.3	84%	0.04	0.08
Bulgarian	6	2.8	2.8	2.0	67%	0.07	0.11
Czech	3	2.9	2.8	2.3	79%	0.05	0.11
Estonian	1	2.9	2.9	2.2	78%	—	—
Hungarian	5	2.6	2.5	2.1	70%	0.14	0.23
Latvian	3	2.8	2.7	2.1	74%	0.07	0.16
Polish	2	2.8	2.9	1.6	54%	0.10	0.04
Romanian	10	2.7	2.7	1.9	68%	0.22	0.23

TABLE 4.3: **Human Evaluation**. Native speakers assess fifty sentences on a scale of 1 to 3, with 3 the highest score. Good Paraphrases are sentences with high scores (2 or 3) for both Semantic Accuracy and Paraphrasing.

As it is difficult to ensure high quality annotations for 21 languages using crowdsourcing, we relied on colleagues by reaching out on NLP and Linguistics mailing lists. As a result, the number of evaluators per language varies (cf. Table 4.3).

We evaluate multilingual AMR-to-Text generation models in 21 languages. We conduct an ablation study which demonstrates the improvements in modeling performance induced by incorporating graph embeddings, cross lingual embeddings, and pretraining. Finally, we analyze model performance with respect to several linguistic attributes (word order, morphology, paraphrasing, semantic faithfulness) using both automatic metrics and human evaluation.

Multilingual AMR-to-Text Generation

► MONOLINGUAL VS. MULTILINGUAL MODELS. We compare English-XX baselines trained to generate from AMR into a single language with multilingual models. We note that as the English-XX models specializes for each language, they have less to model with the same parameter capacity. Results are shown in Table 4.1. Overall, multilingual models perform well — on 18 of the 21 languages, the performance measured by BLEU is stronger than the monolingual baseline.

One advantage of multilingual AMR-to-Text generation is increased quantities of AMR on the encoder side. This is particularly helpful when the size of the training data is low. For instance, Estonian (*et*) sees a 2.3 BLEU point improvement from multilingual modeling. Conversely, languages such as English, Swedish and French benefit less from multilingual modeling, most likely because there is sufficient data for those languages already.[9] More generally, there is a marked difference between languages for which the training data is large and those for which the training data is smaller. When the training data is large (1.9 to 2M training instances, top part of Table 4.1), the average improvement is +0.36 BLEU (Min:-0.2, Max:+0.9) whereas for

[9] For many languages, such as Slavic languages, it is because the EU expanded to include these countries later on. Thus there is less European Parliamentary proceeding data.

English	**Generation**	This point will certainly be the subject of subsequent further debates in the council.
	Reference	This is a point that will undoubtedly be discussed later in the Council.
French	**Generation**	Je ne suis pas favorable à des exceptions à cette règle.
	Reference	A mon avis, il n'est pas bon de faire des exceptions à cette règle.
Swedish	**Generation**	Därför röstade vi inte för detta betänkande.
	Reference	Vi har därför inte röstat för detta betänkande.

TABLE 4.4: **Example Paraphrases** generated by our multilingual model.

We display sentences in English, French, and Swedish where the generated text is a valid paraphrase of the reference.

languages with smaller training data (400 to 620K training instances, bottom part of Table 4.1), the average improvement is +1.75 (Min:+1, Max:+2.3). These trends are similar to observations on other tasks — namely that pretraining is most helpful when there is not sufficient training data in the task itself to train strong representations.

4.3.6 *Results*

- PERFORMANCE ON GOLD ENGLISH AMR. We evaluate our models trained on silver AMR on the CROSSLINGUAL AMR dataset from Damonte and Cohen (2018) where the input is a gold English-centric AMR and the output is available in three European languages: Spanish, French, and Italian. The results are shown in Table 4.2. Similar to the trends seen when generating from silver AMR, we find that multilingual models have strong performance. BLEU scores are lower than on EUROPARL as the models are tested out of domain (training on parliamentary debates but testing on newswire and forum data domains).

  On English LDC data, we compare to existing work. Even though it is trained on silver AMRs and out of domain, non-LDC data, the multilingual model compares well with previous work (see Table 4.2). When finetuned on the LDC2015E86 train set, our model improves on English by over 1 BLEU point, outperforming all previous work except Zhu et al. (2019). This work directly models the graph structure of AMR with structure aware attention to improve Transformer architectures — this is orthogonal to our main aim of multilingual generation and can be incorporated in future work.

- IMPACT OF MODELING IMPROVEMENTS. For the multilingual model, we display the effect of incrementally adding additional modeling improvements (cf. Table 4.1). Each improvement is essentially universally helpful across all considered languages, though some have a greater improvement on performance than others.

- COMPARISON TO THE HYBRID NLG/MT BASELINE. Compared to the NLG/MT baseline, our multilingual models provide comparable results while providing an arguably simpler approach (end-to-end rather than pipeline) and training on much lower quantities of parallel data — on German and French (very high resource languages with millions of examples of training data), there is slightly stronger performance. On other languages we compare to, the translation models perform a bit worse.

  We further conduct a human evaluation study on Spanish, Italian, and German. We ask evaluators to assess the morphology, word order, and

TABLE 4.5: **Human Evaluation of our approach compared to the Hybrid English AMR to Text + Machine Translation baseline using Gold AMR from LDC2015E86.** Two native speakers per language assess fifty sentences each on a scale of 1 to 3, with 3 being the highest score.

Evaluation	Morphology	Word Order	Semantic Accuracy
Spanish			
Machine Translation	2.9	2.7	2.0
Multilingual AMR to Text	2.8	2.9	2.1
Italian			
Machine Translation	3.0	2.9	2.2
Multilingual AMR to Text	2.9	3.0	2.1
German			
Machine Translation	2.8	2.9	2.0
Multilingual AMR to Text	3.0	3.0	2.2

TABLE 4.6: **Performance when training with increasingly more languages.** Training one multilingual AMR-to-Text model with languages in the related language family improves performance.

Model	es	fr	it	pt	ro
One Language	25.2	20.3	18.9	22.2	32.1
Romance Family	25.5	20.5	19.3	22.5	32.5
All Languages	25.3	20.5	19.3	22.4	32.2
		da	**de**	**nl**	**sv**
One Language		21.3	17.0	18.5	18.7
Germanic Family		21.8	21.9	19.6	19.3
All Languages		21.9	17.5	19.4	19.5

semantic accuracy of our Multilingual AMR to Text system compared to this hybrid English AMR to Text + Machine Translation baseline. We show in Table 4.5 that the two models score very similarly in human evaluation, indicating the strength of this fully multilingual system in producing fluent output.

Analysis of Multilingual Generation

A core challenge for multilinguality is that languages differ with respect to word order and morphology, so models must learn this per language. We use automatic and human evaluation to investigate how these differences affect performance.

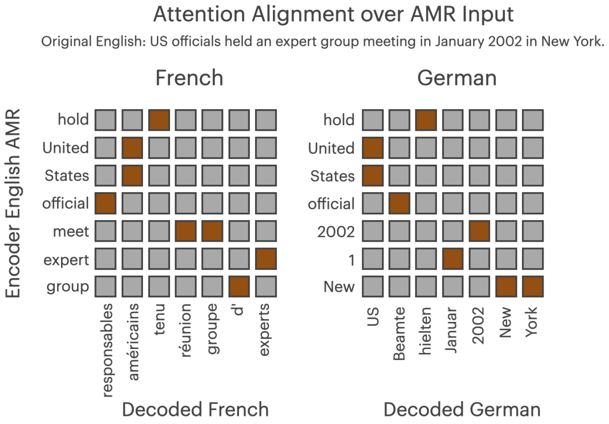


FIGURE 4.4: **Attention alignment** when decoding in French and German from the same input AMR. Based on the English AMR input, the model learns an attention alignment to produce different natural language outputs.

- MORPHOLOGY. Instead of operating on words, our models use sentencepiece (Wu et al., 2016), a data-driven approach to break words into subwords. As shown in Wu and Dredze (2019), in transfer-based approaches to natural language understanding tasks, the proportion of subwords shared between

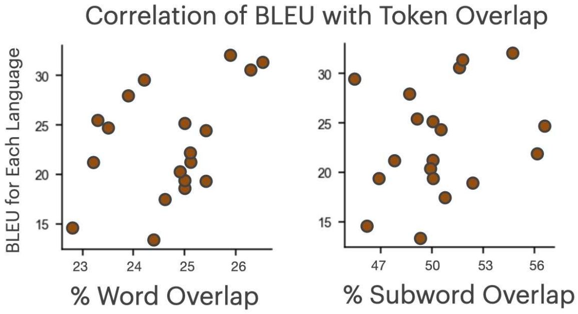


Figure 4.5: **Relationship between BLEU Score and Token Overlap** for all 21 languages. Correlation coefficient between word overlap and BLEU is 0.42, and coefficient between subword overlap and BLEU is 0.26.

the source and the transfer language impacts performance. We therefore explore the relation between the proportion of subwords and words shared between the AMR and the output vocabulary. Figure 4.5 displays this relationship, with weak positive correlation for both word and subword overlap.

We further assess morphology by asking human evaluators to grade the morphology of sentences (*Is the morphology correct? Are agreement constraints e.g., verb/subject, noun/adjective respected?*) on a scale from 1 to 3 with 3 being the highest score. As Table 4.3 shows, there is not much difference in performance between languages even though there is a marked difference in terms of agreement constraints between e.g., Finnish and English. Between annotators, agreement was high — the standard deviation across was low, with the exception of Romanian, Hungarian, and Spanish (as shown in Table 4.3). This demonstrates the surprisingly high ability of multilingual models to generalize across languages.

- WORD ORDER. To assess the impact of varied word orders by language, we ask human evaluators to judge if the word order is natural. As shown in Table 4.3, for all languages except Latvian and Romanian, the score is very high (close to 3) indicating that the model learns to decode into multiple languages even though word order differs. The agreement between annotators was high, with low standard deviation (see Table 4.3). Further, the attention pattern between the encoder English AMR and the decoder clearly reflects the word order of the various languages. This is illustrated in Figure 4.4, where the activation pattern mirrors the word order difference between French (1) and German (2).

(1) *ont tenu (une réunion de groupe)*$_{\text{OBJ}}$ *(en Janvier 2020)*$_{\text{TIME}}$ *(à New York)*$_{\text{LOC}}$

(2) *hielten (im Januar 2020)*$_{\text{TIME}}$ *(in New York)*$_{\text{LOC}}$ *(eine Gruppestreffen)*$_{\text{OBJ}}$

- TRAINING ON RELATED LANGUAGES. Multilingual models have the potential to benefit from similarities between languages. Languages of the same family often have shared morphological characteristics and vocabulary. First, we analyze the performance of training on languages within a family. Table 4.6 displays that a model trained on languages within a family has the strongest performance.

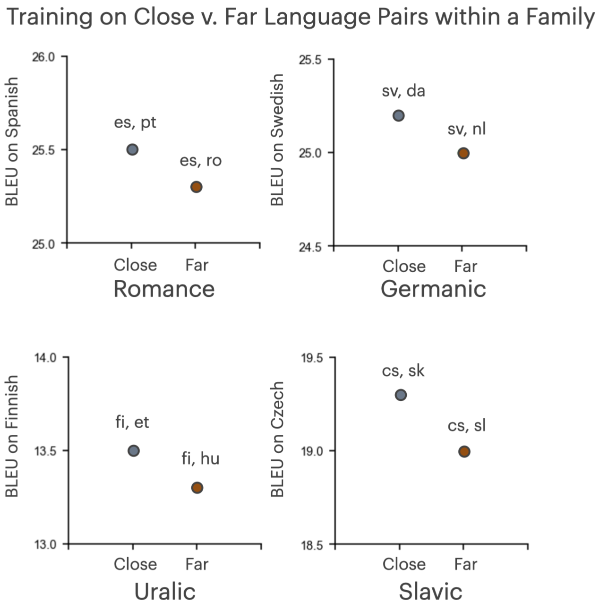


FIGURE 4.6: **BLEU Difference Training on Close v. Far Languages Within One Family**. Training on a close pair consistently improves performance compared to training on a far pair, even within a language family.

Second, we analyze languages within the same family. For four families: Romance, Germanic, Uralic, and Slavic, we create multilingual models trained on pairs. One pair is for the most related languages within that family (e.g. Spanish and Portuguese) and another pair is for the farthest languages within that family (e.g. Spanish and Romanian). We determine which pairs are close and far from Ahmad et al. (2019). Results in Figure 4.6 display that training on pairs of closely related languages has better performance than pairs of less closely related languages, even within a family. Multilingual models could pick up on similarities between languages to improve performance.

► SEMANTIC ACCURACY AND PARAPHRASING. We ask human evaluators to grade the faithfulness of the hypothesis compared to the reference on a scale of 1 to 3. As shown in Table 4.3, the overall semantic accuracy is very high (note a score of 2 indicates *minor differences*). We also asked annotators to evaluate how different the generated sentence was from the reference. When coupled with the semantic accuracy score, this allows us to evaluate generation of true paraphrases i.e., sentences with the same meaning as the reference but different surface form. In Table 4.3, *Good Paraphrases* indicates the percentage of cases that scored highly (2 or 3) with respect to both semantic adequacy and paraphrasing. A large majority of generated sentences are labeled as valid paraphrases by native speakers, indicating (i) that despite underspecified input, the written sentence retains the meaning of the reference and (ii) that this underspecification allows for the generation of paraphrases. This also suggests that BLEU scores only partially reflect model performance as good paraphrases typically differ from the reference and are likely to get lower BLEU score even though they may be semantically accurate. Table 4.4 shows some examples illustrating the paraphrasing potential of the approach.

4.3.7 *Conclusion*

We investigate the generation of text not from natural language but from structured meaning representations such as AMR. Such tasks are a microcosm of other structured generation tasks, such as decoding from RDF or knowledge graphs. We focus on the development not of individual systems specialized for specific languages, but on multilingual models that can generate in over 20 different languages from the same AMR input.

▸ CONCLUSION

Despite recent advancements, many challenges face current text generation systems — even sentence-level systems that generate from short input texts. In this section, we focus on the challenge of multilingual generation, or development of systems that extend to or encompass to languages beyond English.

First, we focus on developing approaches that could be replicated for other languages. The vast majority of text generation methods require supervised training (or at least, finetuning), making extensions to other languages nontrivial as supervised corpora may not exist. We demonstrate the applicability of large-scale mining methods to the automatic creation of training corpora in other languages, and show that this creates state-of-the-art sentence simplification systems for those languages. Based on human evaluation and automatic metrics, these techniques show great potential for other text generation tasks.

Second, we focus on an approach that creates multilingual models, or models that can encompass a large number of languages in one system. Creating one model rather than developing specialized systems is much easier for both research iteration and practical deployment. We demonstrate how to successfully create multilingual architectures that decode from structured input such as AMR, incorporating techniques such as pretraining, crosslingual embeddings, and graph embeddings to generate text in over 20 different languages. After extensive human evaluation, we find that the multilingual model we develop retains high quality even when handling so many languages, and is competitive with specialized single-language and translation-based baselines based on automatic metrics.

These techniques are broadly applicable to a wide variety of generation tasks, and together represent two general approaches to successfully creating generation models for languages beyond English.

Part III

TEXT GENERATION WITH RETRIEVAL

5

Retrieval for Knowledge-Based Text Generation

► SYNOPSIS We motivate the need for text generation models that can access and effectively utilize knowledge, and describe the challenges of developing such systems. In the subsequent chapters, we describe knowledge-based generation systems operating at ever-increasing scale: from knowledge in single documents, to identifying information in all of Wikipedia, and finally retrieving knowledge on the open web.

5.1 THE NEED FOR KNOWLEDGE

> *"Information is a source of learning. But unless it is organized, processed, and available to the right people in a format for decision making, it is a burden, not a benefit."*
> —William Pollard

In the previous chapters, we discussed generation of natural language with the constraint that the majority of the information required to produce the text was contained in the input. The task of sentence simplification, for instance, largely focuses on understanding a complex input sentence and producing a shorter, simpler output sentence. However, natural language generation is applicable to a much broader set of tasks. For example, generation systems can be used to answer questions, interact with people in the form of dialogs, write Wikipedia articles, and so on. These applications expose a fundamental challenge of generation systems: the need to *access knowledge*.

To illustrate, let us consider the challenge of creating a dialog system that users can freely interact with. A system that generates dialogs naturally needs to interact with the user, so must be able to write fluent, engaging responses and stimulate conversation. Fundamentally, *chatting* is completely unconstrained — the generation system would need to be able to generate text about any topic. A response to an utterance such as *I like Slinky Dog! He's sooo cute in the latest movie* would be immensely challenging without being able to reference some body of information, where a system might be able to understand that *Slinky Dog* is a character in Toy Story.

This inherent need for knowledge permeates many applications for natural language generation. Beyond dialog, tasks such as question answering, fact checking, writing a news article, writing a biography, and so on necessitate a knowledge source. For humans, doing such activities causes us to instinctively attempt to identify a nearby information source — from the libraries and encyclopedias we used as children to typing in a search engine in today's modern world. Machine learning models are no different. In the following chapters, we discuss how to create text generation systems that can access knowledge effectively.

5.1.1 *Structured and Unstructured Knowledge*

Information can be found in many different forms and from many different sources. Broadly, knowledge can come in two forms: structured and unstructured. We refer to *structured* knowledge as information that comes in a predefined schema. Examples include knowledge graphs, where information is often represented as nodes linked by edges that indicate specific relationships. Other examples include knowledge bases or other kinds of tabular data, where rows and columns have specific meanings. In contrast, *unstructured* knowledge is information that exists in a freeform fashion. Imagine, for example, reading a paragraph about the population of each country in the world, instead of an organized table.

In reality, information is often a mix of both structured and unstructured. Articles on Wikipedia, for example, are organized by categories, subcategories, and keywords, and within an article, there are section headings, subheadings, and tabular information. Further, Wikipedia information exists as the article itself, but can also be accessed in structured form on databases such as WikiData and DBPedia. In this work, we explore a wide variety of representations of knowledge, including blending both structured and unstructured information.

5.1.2 *Representing Knowledge in Neural Models*

How do neural text generation systems feasibly access knowledge and subsequently incorporate it? Various work has designed a variety of different methods, which we broadly categorize into two groups. We provide a brief overview here.[1]

First, work has focused on identifying the relevant knowledge required to complete a task, and incorporating that as input into standard neural architectures. Much of this relates to the field of information retrieval. For example, to answer a question in a question answering task, a model may identify a relevant paragraph, then provide that paragraph as input to a Sequence-to-Sequence architecture. Broadly, these approaches generally operate by encoding text into a vector representation of either sparse or dense features, then identifying other relevant text based on what is closest in representation space. Often, this can be done using nearest neighbor methods. In this work, we develop multiple different retrieval mechanisms for the identification of relevant information, as well as more sophisticated techniques for incorporating retrieved knowledge into neural architectures.

Second, a large body of work has focused on *architectures* that can comprehend various different structured information sources. Examples include encoding knowledge graphs or the tabular structure of databases, either with specialized models such as Graph Neural Networks or linearization methods to incorporate structured information in standard Sequence-to-Sequence

[1] We refer the reader to Yu et al. (2020) for a detailed survey of neural methods that incorporate knowledge. While this represents much of the focus of current research in knowledge representation in text, we refer the reader to Mitra and Craswell (2017) that covers a much longer history of information retrieval and knowledge incorporation in machine learning pre-neural methods.

models. In this work, we explore both structured and unstructured knowledge, and consider how to best represent structured knowledge graphs.

5.2 CHALLENGES

A number of challenges face knowledge-based text generation systems.

► SCALE We start by describing the challenge of scale, which is succinctly summarized as *there is too much knowledge in the world*. New facts and findings appear every day, and the scale of the internet continues to grow, with more and more data being produced at an exponential rate. One of the biggest challenges facing knowledge-based systems is the need to handle this scale. Only a few years ago, question answering systems focused on identifying relevant information in short paragraphs — in this work, we focus on finding the right information not at the paragraph level, but in all of Wikipedia and even open web search. This scale of information retrieval necessitates developing techniques that are fast and scalable.

► RELEVANCE A second challenge is relevance, that of identifying not just information that is related, but actually relevant to completing the task for the text generation system. Fetching information that is related, but not useful for the task at hand would not allow the generation system to perform the task most effectively. This is immensely complex: a simple keyword search for *Slinky Dog* could return millions of hits, but might not return a list of other characters in Toy Story.

Beyond this, relevant knowledge is very nuanced. For example, factual information changes over time, so knowledge needs to be accurate for the time in question. Information on the internet from various sources can disagree, and models must face the difficulty of representing multiple perspectives and resolving contradiction. Even if multiple pieces of information are relevant, they may not have the same quality. For instance, certain sources of knowledge are often more trustworthy than others. Thus, not only do models need to identify the relevant content, but must consider a myriad of nuances of representing that identified knowledge.

► MODALITY Finally, knowledge can exist in multiple modalities, beyond natural language. We have discussed tabular data or graphical structures of representing knowledge, but information can also be contained in images (such as a photo of a line chart). In this work, we primarily focus on sources of knowledge in the form of natural language, though we do describe general techniques that can identify relevant content in multiple modalities.

5.3 STRUCTURE OF THIS SECTION

In the subsequent chapters, we describe methodology and applications of knowledge-based text generation at an increasing scale of needed retrieval. Current methods, heavily focused on question answering applications, focus on identifying the correct sentences or individual paragraphs to generate answers. We first begin with a question generation and question answering generation task that requires models to internalize evidence at the level of entire documents. Next, we expand beyond retrieving the relevant document, and scale to retrieving multiple pieces of information across all of Wikipedia, in an open-domain dialogue task that searches through hundreds of thousands of sentences. Subsequently, we demonstrate the capability of retrieval-based generation methods to retrieve multiple relevant documents from the searches on the internet: truly scaling text-generation to web-scale knowledge. Finally, we end by considering web-scale information retrieval not by digesting free text, but by creating a *structured* representation of that knowledge in the form of a knowledge graph.

6

Knowledge from a Single Document

"Truth is singular. Its 'versions' are mistruths."
—David Mitchell

▸ SYNOPSIS We begin with a description of knowledge-based text generation systems that take as input a single document. We focus on a fact checking task, where we generate structured *briefs* to aid human fact checkers. These briefs are structured as a series of questions and answers, where a search engine is used to retrieve the single most relevant document to generate an answer.

6.1 MOTIVATION: DOCUMENT-LEVEL KNOWLEDGE

The incorporation of knowledge in text generation systems has a long history.[1] Most predominantly thought of, at least in recent years, has been the task of question answering — focusing on identifying the answer to a question of any domain. Much of the development of question answering methods leveraged *extractive* models, rather than generative ones. Instead of writing the textual answer, various benchmarks focused on identifying on span extraction, or copying a portion of natural language as the answer. Recent work, particularly with the shift to a strong pretraining and subsequent finetuning regime, has converted these benchmarks to be instead generative ones, and generative question answering has become a popular approach to question answering tasks.

Common question answering benchmarks focus on identification of an answer based on a paragraph of input information. Examples include the popular Stanford Question Answering Dataset (Rajpurkar et al., 2016), Conversational Question Answering Dataset (Reddy et al., 2018), and HotpotQA (Yang et al., 2018). In these datasets, the provided input knowledge is sufficient for answering the question.[2] However, in a realistic setting, users with questions most likely do not know which specific paragraph would contain the answer to their question. Thus, expanding beyond paragraph-level knowledge to create systems that can read much more input information is a very applicable challenge. In this chapter, we focus on a generative question answering system that leverages a search engine to identify the most relevant single web article, then uses that document as evidence to generate an answer. We describe this system in the context of the fact checking domain, which we motivate in the next section.

[1] We refer the reader to an excellent reading list: `https://github.com/wyu97/KENLG-Reading`

[2] We note a variety of question answering datasets now include popular *no answer* or *insufficient information* options, to reflect these challenges.

6.2 FACT CHECKING AS A KNOWLEDGE-BASED TEXT GENERATION TASK

When a user reads a piece of information, how do they know if they can trust it? The scale of knowledge available freely at one's fingertips with a simple query to Google is unprecedented. However, this ease of discoverability is largely double-sided. Misinformation is widely prevalent online — from simply outdated information such as the president of the United States now being Joseph Biden and no longer Barack Obama, to more insidious content that has serious implications on people's health and safety. In the last couple of decades, the need for *fact checking* has risen, or the creation of institutions that professionally attempt to verify claims online. For example, most presidential debates in the United States in recent years have included live fact checkers, tasked with verification challenges. Beyond professionals, though, the ability to fact check content accurately has strong implications for the development of digital literacy and safe consumption of content online. Thus, we focus on fact checking as an important area where the every day individual might want to use artificial intelligence systems to aid them.

But where do text generation models fit into all this? In this chapter, we focus on automatically creating systems to work hand-in-hand with every day people, to help users identify misinformation online. Most often, people who want to fact check information turn to search engines for verification, but must spend a significant amount of time before reaching their answer. A myriad of reasons affect this amount of time required, but one of the root causes is lack of background knowledge. For example, an individual reading the morning news about a proposed bill in Congress may not know the rough contents of that bill, or which political figure proposed it. Thus, we propose using text generation models to automatically write useful background context and provide it to the user. We frame this as a multi-step generation system that first generates questions, and then generates answers to those questions. We describe a dataset, model, and various metrics to measure the success of such a system in the subsequent section.

6.3 GENERATING FACT CHECKING BRIEFS

6.3.1 *Introduction*

Fact checking is a challenging task. It requires deep knowledge of a claim's topic and domain, as well as an understanding of the intricacies of misinformation itself. Checking a single claim can take professional fact checkers 15 minutes to one day (Hassan et al., 2015). Volunteers on the other hand are not considered accurate enough; with access to a search engine, Roitero et al. (2020) report crowdsourced fact check accuracies of around 58%. This result corroborates earlier reports[3] by fact checking websites which attempted to engage volunteers, but reported success only for claim detection, which is considered a much simpler task (Konstantinovskiy et al., 2018). This is problematic, both from the perspective of using crowdsourced fact checking to combat misinformation and from the perspective of helping individuals fact check themselves.

[3] `http://mediashift.org/2010/11/crowdsourced-fact-checking-what-we-learned-from-truthsquad320/`, `http://fullfact.org/blog/2018/may/crowdsourced-factchecking/`

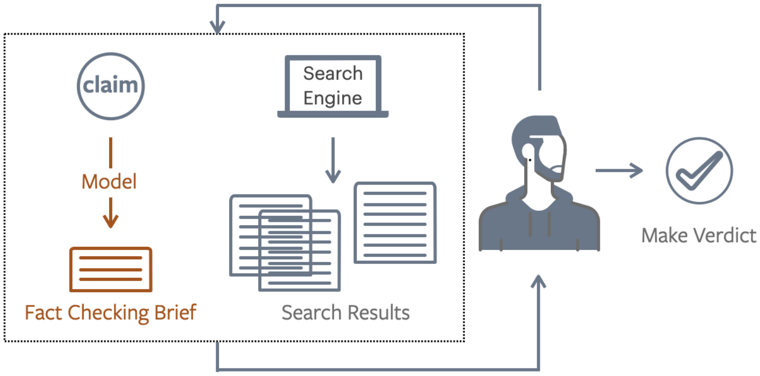
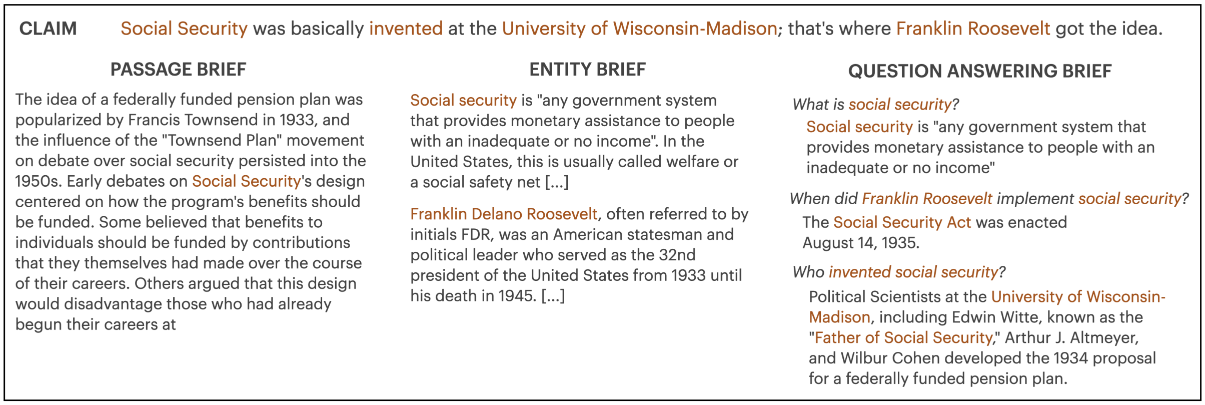


FIGURE 6.1: **Fact Checking Briefs.** Before conducting a fact check, we propose generating *briefs* to provide information about the claim. We show they make fact checking more accurate and efficient.



CLAIM Social Security was basically invented at the University of Wisconsin-Madison; that's where Franklin Roosevelt got the idea.

PASSAGE BRIEF

The idea of a federally funded pension plan was popularized by Francis Townsend in 1933, and the influence of the "Townsend Plan" movement on debate over social security persisted into the 1950s. Early debates on Social Security's design centered on how the program's benefits should be funded. Some believed that benefits to individuals should be funded by contributions that they themselves had made over the course of their careers. Others argued that this design would disadvantage those who had already begun their careers at

ENTITY BRIEF

Social security is "any government system that provides monetary assistance to people with an inadequate or no income". In the United States, this is usually called welfare or a social safety net [...]

Franklin Delano Roosevelt, often referred to by initials FDR, was an American statesman and political leader who served as the 32nd president of the United States from 1933 until his death in 1945. [...]

QUESTION ANSWERING BRIEF

What is social security?

Social security is "any government system that provides monetary assistance to people with an inadequate or no income"

When did Franklin Roosevelt implement social security?

The Social Security Act was enacted August 14, 1935.

Who invented social security?

Political Scientists at the University of Wisconsin-Madison, including Edwin Witte, known as the "Father of Social Security," Arthur J. Altmeyer, and Wilbur Cohen developed the 1934 proposal for a federally funded pension plan.

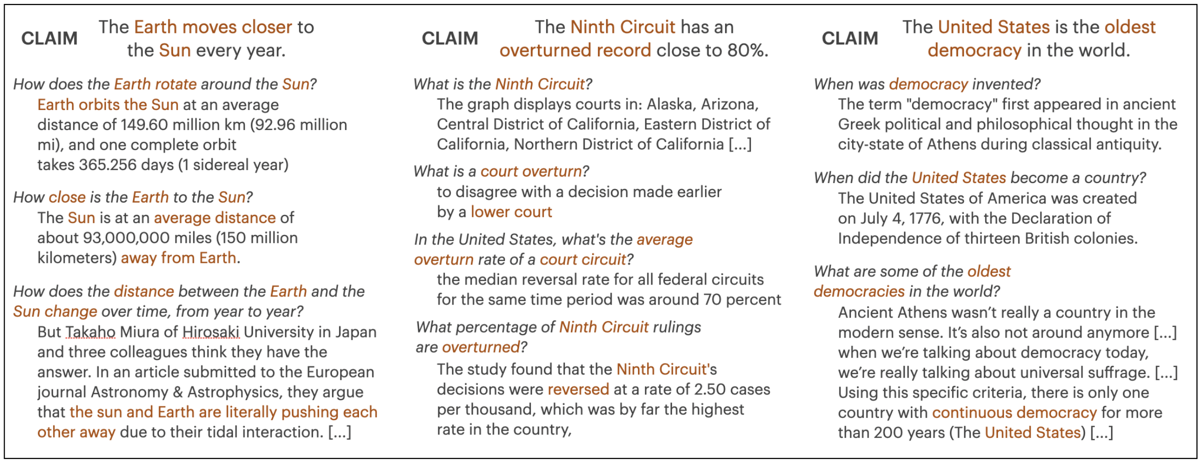
FIGURE 6.2: **Three Types of Briefs: (1) Passage Briefs**, based on information retrieval applied to the claim, **(2) Entity Briefs**, using entity linking to identify information about each entity, and **(3) Question Answering Briefs**, which condition on the claim to generate questions, then answer questions using open domain question answering

One path for scaling fact checking could be through full automation, taking a claim as input and producing a verdict (Vlachos and Riedel, 2014). Existing work has framed fact checking as classification, often supported by evidence (Wang, 2017; Thorne et al., 2018; Augenstein et al., 2019). However, due to the limitations of existing automated solutions, practitioners prefer solutions that improve efficiency in reaching a verdict, instead of approaches to the complete process (Graves, 2018).

In this work, we propose *briefs* to increase the accuracy and efficiency of fact checking (Figure 6.1). By generating fact checking briefs, our models aim to provide evidence a human fact checker would find useful. We investigate several approaches, including returning Wikipedia passages that relate to the claim, and an entity linking approach that shows information about mentioned entities. Crucially, we introduce QABriefs — a set of relevant questions and their answers (see Figure 6.2).

To learn how to produce QABriefs and create training data, we use crowdsourcing to gather such briefs based on existing fact checks. We create QABRIEFDATASET, a collection of about 10,000 QABriefs with roughly 3 question and answer pairs each. We introduce QABRIEFER, a novel model that performs structured generation via claim-conditioned question generation and open domain question answering. Each question is used to identify evidence using a search engine. Finally, a pretrained question answering model is finetuned to generate answers and produce the full brief.

In experiments with crowdworkers, QABriefs improve accuracy by 10% compared to using only a search bar while reducing the time a fact check



CLAIM The Earth moves closer to the Sun every year.

How does the Earth rotate around the Sun?
Earth orbits the Sun at an average distance of 149.60 million km (92.96 million mi), and one complete orbit takes 365.256 days (1 sidereal year)

How close is the Earth to the Sun?
The Sun is at an average distance of about 93,000,000 miles (150 million kilometers) away from Earth.

How does the distance between the Earth and the Sun change over time, from year to year?
But Takaho Miura of Hirosaki University in Japan and three colleagues think they have the answer. In an article submitted to the European journal Astronomy & Astrophysics, they argue that the sun and Earth are literally pushing each other away due to their tidal interaction. [...]

CLAIM The Ninth Circuit has an overturned record close to 80%.

What is the Ninth Circuit?
The graph displays courts in: Alaska, Arizona, Central District of California, Eastern District of California, Northern District of California [...]

What is a court overturn?
to disagree with a decision made earlier by a lower court

In the United States, what's the average overturn rate of a court circuit?
the median reversal rate for all federal circuits for the same time period was around 70 percent

What percentage of Ninth Circuit rulings are overturned?
The study found that the Ninth Circuit's decisions were reversed at a rate of 2.50 cases per thousand, which was by far the highest rate in the country,

CLAIM The United States is the oldest democracy in the world.

When was democracy invented?
The term "democracy" first appeared in ancient Greek political and philosophical thought in the city-state of Athens during classical antiquity.

When did the United States become a country?
The United States of America was created on July 4, 1776, with the Declaration of Independence of thirteen British colonies.

What are some of the oldest democracies in the world?
Ancient Athens wasn't really a country in the modern sense. It's also not around anymore [...] when we're talking about democracy today, we're really talking about universal suffrage. [...] Using this specific criteria, there is only one country with continuous democracy for more than 200 years (The United States) [...]

FIGURE 6.3: **Examples of QABriefs in QABRIEFDATASET**

. We depict three different claims and display the annotated questions and answers from the dataset.

takes. For volunteer fact checkers, accuracy is improved by 4% and the process is 20% faster compared to using a search bar. Using QABriefs from human annotators leads to the largest improvement, followed by briefs generated by QABRIEFER and other proposed forms of briefs. This suggests that briefs are a promising avenue for improving crowdsourced fact checking. Further, QABRIEFDATASET can be used to develop models capable of answering challenging, real world questions.

6.3.2 *Related Work*

Previous work in NLP has focused on claim veracity. It has been treated as a classification problem (Wang, 2017), often using stance detection (Riedel et al., 2017). The FEVER Challenge (Thorne et al., 2018) proposed providing provenance for a decision along with classification, and various approaches developed combine information retrieval with stance detection or question answering (Li et al., 2018; Lee et al., 2018). Question generation and answering has been considered in the context of FEVER (Jobanputra, 2019) — the focus was on eliciting the right answer from a question answering system rather than improving the accuracy and efficiency of human fact checkers.

However, FEVER is based on modified Wikipedia sentences, not real world claims, which are arguably more difficult. To address this Hanselowski et al. (2019) considered the claims fact checked by the website Snopes, but used the reports accompanying them as evidence instead of finding the evidence directly. Popat et al. (2018) and Augenstein et al. (2019) used search engines, but without ensuring that they provide evidence supporting/refuting the claim instead of being related to it or that they were not fact checking reports. Finally, Kochkina et al. (2018) used responses on social media for rumour verification, but did not address evidence finding.

Various work studies how to improve the fact checking process. Analysis shows accuracy can improve by providing feedback (Hill, 2017), additional time (Bago et al., 2020), tooling (Karduni et al., 2019), or training (Zhang et al., 2018a). These works are complementary to ours — we provide support in the form of briefs. Studies emphasize that current solutions for fully automated fact checking face various challenges (Graves, 2018) that must be addressed with interdisciplinary research (Karduni, 2019). Developing tools to aid human-in-the-loop fact checking has received increasing attention, from NLP to human-computer interaction and psychology, often with positive results when tested with journalists (Miranda et al., 2019) and professionals (Lurie, 2019).

6.3.3 *Briefs for Fact Checking*

Fact checkers must comprehend each part of a claim, which requires gathering information about a wide range of concepts— a precise definition of a term, how a politician voted, or the exact contents of a bill. Such knowledge is available in many sources: knowledge bases, statistical reports, or on the internet. We introduce the notion of *briefs* to provide relevant information to fact checkers—as if *briefing* them before fact checking— and explore three possible forms: Passage Briefs, Entity Briefs, and Question Answering Briefs. We show how they can be constructed with modern NLP approaches.

Passage Briefs

To provide information before checking a claim, Passage Briefs consist of relevant passages retrieved from Wikipedia. For the claim in Figure 6.2, information about the history and implementation of social security in the United States is retrieved and presented as background for the fact checker. To generate Passage Briefs, we identify relevant Wikipedia passages for each claim. Based on the results by Lewis et al. (2020b) on open-Wikipedia tasks, we use the Dense Passage Retriever (DPR) (Karpukhin et al., 2020). This state of the art, pretrained retriever model learns a representation of questions and possible relevant paragraphs.[4] In our case, we provide the claim as input instead of the question, rank the outputs, and select the top ranked passage. We limit to 500 tokens for readability. Initial experiments suggested web-based Passage Briefs returned poor results for most claims, as it relied on finding a single passage addressing the entire claim, so we keep the Passage Brief focused on Wikipedia. Further, DPR is trained on Wikipedia, and we found the best performance within this domain.

Entity Briefs

Passage briefs provide information from a single passage, but claims are complex and often require multiple pieces of information from different sources. Thus we propose entity briefs that focus on each entity referenced in the claim.

Entities in each claim are identified with BLINK (Wu et al., 2019), a model trained on Wikipedia data that links each entity to its nearest Wikipedia page. BLINK combines a bi-encoder (Urbanek et al., 2019; Humeau et al., 2019) that identifies candidates with a cross-encoder that models the interaction between mention context and entity descriptions. For each entity, we retrieve its Wikipedia page and provide the first paragraph in the brief. In Figure 6.2, *Franklin Roosevelt* is an entity, and the brief communicates he is *an American politician who served as the 32nd president of the United States [...].* However,

[4] The Dense Passage Retriever focuses on learning dense vector representations using pairs of questions and answers, without relying on additional pretraining methods. It is based on the BERT architecture. At inference time, passages are encoded with the DPR passage encoder, and then the `faiss` library is used to conduct the final similarity search.

unlike Passage Briefs, if several entities are identified, information from multiple pages is displayed in an Entity Brief.

Question Answering Briefs

Entity briefs provide information about entities mentioned in the claim, but not necessarily the evidence needed for the claim in question. For this reason we propose QABriefs, which decompose fact checking into a set of questions and answers. E.g. the claim in Figure 6.2 could be split into understanding what social security is, identifying who invented the concept, and finally where Franklin Roosevelt got the idea. Each step can be written into a question — *What is social security? Who invented social security?* — that is then answered. The decomposition into question-answer pairs is likely to be better amenable to the current generation of information retrieval systems, which typically assume simpler information needs, e.g. most QA datasets have questions about single factoids. Unfortunately, there are no existing datasets or models available to create QABriefs. Next, we describe how we create a dataset (Section 6.3.4) and a model (Section 6.3.5) to produce QABriefs.

6.3.4 *QABrief Dataset*

To train and evaluate models to generate QABriefs, we collect a dataset of questions based on claims, together with answers to those questions found on the open web. Crucially, annotators first read the article from a fact checking website that describes how the claim was checked, and then decompose the process into questions, for which answers are provided. The claims for the dataset are sourced from existing fact checking datasets, specifically DATACOMMONS[5] and MULTIFC (Augenstein et al., 2019). The annotator instructions are in the Appendix and examples are shown in Figure 6.3.

While numerous question generation and answering datasets exist, none of them focuses on using questions and answers to combat misinformation. QABRIEFDATASET focuses on this real world problem, with each question grounded in a claim that was actually fact checked. Further, existing datasets are quite different from our usecase — for example, many datasets are based on Wikipedia, but fact checkers find evidence from other sources. Many datasets have short answer spans, but our questions are complex, so require longer answers.

Question Generation

Crowdworkers are asked to read a *claim* and its corresponding *fact checking article*,[6] which details the investigative process used to perform the fact check. After reading the article, crowdworkers write *questions* to reconstruct the process taken by professional fact checkers. For each claim, crowdworkers write two to five questions that are at least five words long and standalone. For instance, the question *why did he do that* is invalid, as it is not clear what *he* or *that* is. We discourage questions with yes/no answers and discourage questions about the same claim from overlapping more than five words.

[5] `https://datacommons.org/factcheck`

[6] For our running example, the reference article is: `https://www.politifact.com/factchecks/2016/dec/16/russ-feingold/was-social-security-basically-invented-university-/`

After the questions are collected, a *question validation* phase is conducted. A separate group of crowdworkers reviews the quality of the questions and flags those that are redundant and/or otherwise poor quality. For example, questions such as *What evidence is there that [claim] is true?* are rejected. Other instances of questions rejected at this phase include nonsensical questions and questions that simply rephrase the claim. Any questions that do not pass this review are re-annotated. Finally, a *question clarity* phase is conducted — crowdworkers read the questions and edit those that are unclear or underspecified. For example, questions may need to have a year added to them to accurately verify a statistic. Further, additional questions can be added if crowdworkers feel the existing questions are not sufficient. This can lead to more than five questions per claim. Spelling errors are highlighted and crowdworkers are encouraged to correct them.

Question Answering

After each claim is annotated with multiple questions, we proceed to collect the answers to them. To answer questions, crowdworkers are given the claim; the source of the claim (for example, the entity who said the quote being checked); and the question. Crowdworkers enter a *query* into a *search engine* to find information on the web. The search is restricted from accessing fact checking domains, to prevent the answer from being trivially found on a fact checker's website. The query does not need to be identical to the question, and is often rephrased to find better search results. After reading the returned results, crowdworkers can provide one of three possible answer types:

- *Extractive* — the encouraged option, crowdworkers copy paste up to 250 words as an answer. We focus on extractive answers, as identifying such an answer is more straightforward compared to writing an answer.
- *Abstractive* — if the answer is present in an image or graph, crowdworkers write an abstractive answer of at least 20 words.
- *No Answer* — if no answer can be found, crowdworkers write an explanation of at least 20 words to describe why there is no answer.

Next, *validation* is conducted. The questions are complex, so we do not assume the answer is known. Crowdworkers instead flag answers that seem incorrect. For example, if the answer to *How many people live in California* is *three billion*, this would be flagged and re-annotated. A last step is conducted for answers that are *No Answer*. To verify that answers cannot be found, a second group of crowdworkers tries to find an answer. If an answer is found, the *No Answer* annotation is discarded.

QABrief Dataset Statistics

In summary, QABRIEFDATASET includes 6,897 claims and 21,168 questions paired with their answers. We use 500 claims as a validation set and 500 claims as a test set. The validation and test sets include around 1400 questions and answers each.

Table 6.1: **Statistics of QABriefDataset.** We display the sizes of the Train, Valid, and Test data splits as well as the average number of questions in a brief and length of questions and answers.

Train	Number of Claims	5,897
	Number of QA Pairs	18,281
Valid	Number of Claims	500
	Number of QA Pairs	1,431
Test	Number of Claims	500
	Number of QA Pairs	1,456
Avg Number Questions/Claim		3.16
Avg Number Words in Questions		10.54
Avg Number Words in Answers		43.56

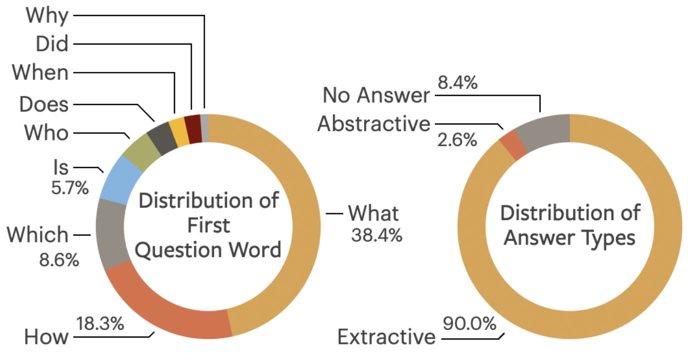


Figure 6.4: **Question and Answer Types.** The collected dataset has a variety of question types. For answer types, the vast majority are extractive, with a few abstractive and no answer options.

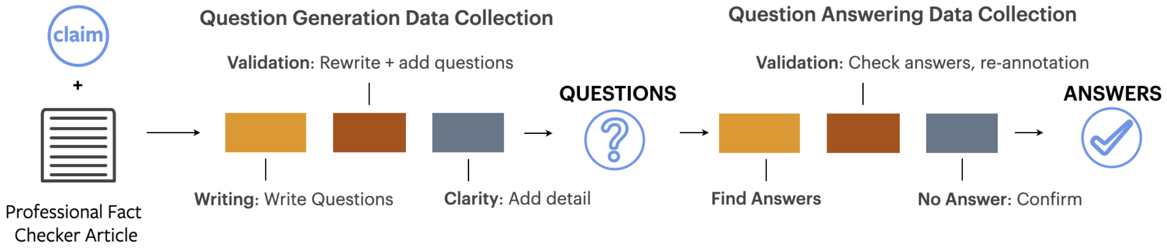

claim
+
Professional Fact Checker Article
Question Generation Data Collection
Validation: Rewrite + add questions
Writing: Write Questions
Clarity: Add detail
QUESTIONS
Question Answering Data Collection
Validation: Check answers, re-annotation
Find Answers
No Answer: Confirm
ANSWERS

Figure 6.5: **QABriefer Model.** First, BART is finetuned to conduct claim-conditioned question generation and generates a sequence of questions that decompose a fact check. Second, we use an information retrieval system and a second finetuned BART model to extract long-form answers to each question.

We examine the types of questions to analyze the diversity. Table 6.1 shows that each claim on average requires around 3 questions to cover the different parts of the claim, and questions contain 10.5 words on average. The questions are quite diverse, as seen in Figure 6.4 (left), though the majority begin with *What, How, Which* question words. There are few *Why* questions, indicating a focus on verifying factual information, rather than causality.

The answers obtained have mainly extractive annotations, though a small portion of abstractive and no answer options exist (see Figure 6.4, right). Answers are around 43.5 words long (Table 6.1), though abstractive answers are generally shorter as crowdworkers must fully write them.

We examined a subset of 50 claims where we conducted multiple data collection trials with the same claim to understand the agreement rate between workers. We found that for the question annotation step, about half of the questions provided by different people on the same claim were very similar and could be considered paraphrases. For example, the questions *Who invented social security* and *Who was the invetor of social security*. For the answer annotation step, the identified answers varied in length but were

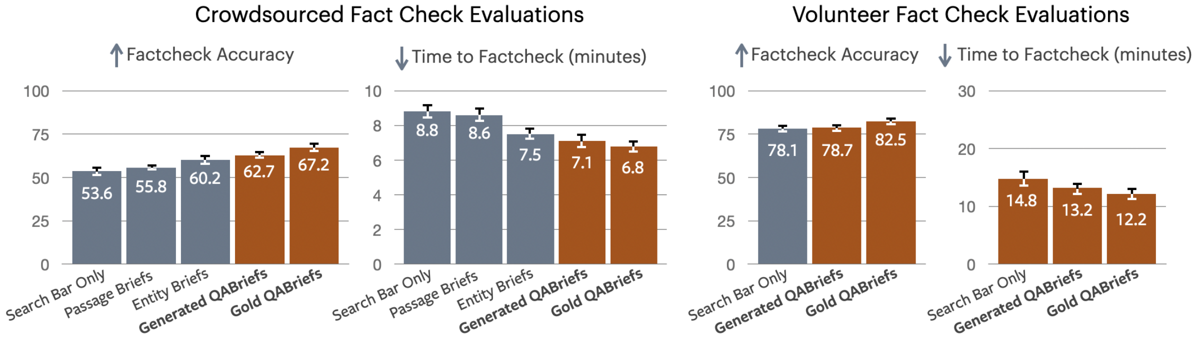


FIGURE 6.6: **Accuracy and Time Taken to fact check** by Crowdworkers (left) and Volunteer fact checkers (right). Briefs of various forms, but particularly QABriefs, increase fact checking accuracy and efficiency.

often paraphrases — some crowdworkers tended to select only the specific span that answered the question (e.g. an entity name), while others chose several sentences to capture the context.

6.3.5 *QABrief Model*

The automatic generation of QABriefs presents numerous modeling challenges. Generating such a brief is a hierarchical process: writing the questions, and then conditioned upon the questions, searching the web and writing the answers. While many question answering datasets exist, questions in QABRIEFDATASET are grounded on real claims that were fact checked. The diversity of the claims renders reusing questions across claims unlikely to work, thus precluding the use of retrieve-and-rank approaches (Rao and Daumé III, 2018). Unlike previous question generation models (Du et al., 2017; Duan et al., 2017; Tang et al., 2017; Zhao et al., 2018d) that generate based on an answer, we treat question generation closer to structured planning — laying out the format for the entire brief.

In contrast to most question answering datasets, the length of the answers in QABRIEFDATASET are long-form (Fan et al., 2019b). For example, the average answer in SQuAD (Rajpurkar et al., 2016) is four words long, while the average answer in QABRIEFDATASET is forty. Further, datasets such as SQuAD, Natural Questions (Kwiatkowski et al., 2019), and HotpotQA (Yang et al., 2018) are built from Wikipedia, while QABriefs uses the web.

In this section, we describe QABRIEFER (see Figure 6.5). For each claim, the question generation model is used to generate multiple questions. For each question, an evidence document is retrieved using a search engine. We take the top search hit as the evidence and retrieve the text from CommonCrawl.[7] Finally, the generated question and retrieved evidence document is provided to the question answering model to generate an answer.

Question Generation

The first step of QABRIEFER is to create the questions that will form the structure of the brief. To create models that can take a claim as input and generate a sequence of questions as output, we use sequence-to-sequence (Sutskever et al., 2014) models. As QABRIEFDATASET is not large enough to train the language model needed for question generation, we leverage advances in pretraining and use QABRIEFDATASET to adapt it to the task at hand. We use BART (Lewis et al., 2020a), a denoising autoencoder that uses various noise

[7] `http://commoncrawl.org/`

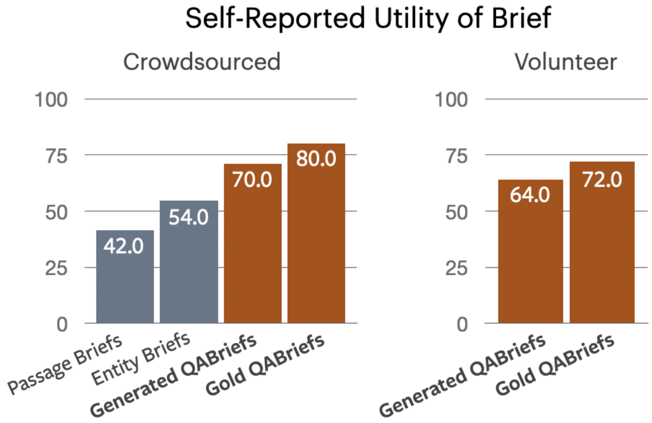


FIGURE 6.7: **Usefulness of Briefs** reported by Crowdsourced and Volunteer Fact Checkers. Overall, people report that briefs are very useful, even if they are generated.

functions and trains to recreate the input. In adapting BART for question generation based on claims, we explore three options: generating all questions based only on the claim, generating all questions based on the claim and the source of the claim (usually an entity), and generating questions one at a time. To write questions one at a time, the model conditions on the previous questions as well as the claim and source, and needs to predict the subsequent question or an *end of questions* token.

Question Answering

Given the question-based structure for QABriefs, the second part of the hierarchical process is to identify answers. Models take as input the question and evidence document that annotators indicated to contain the answer, and produce an answer. As QABRIEFDATASET does not have enough data to train a question answering model from scratch, we use BART finetuned on Natural Questions. and subsequently finetune it further on QABRIEFDATASET. As the dataset contains extractive and abstractive answers as well as questions where the model must provide an explanation to justify no answer, we use an abstractive approach with a generative model; abstractive models have shown strong performance on various question answering tasks (Lewis and Fan, 2018; Dong et al., 2019; Radford et al.; Raffel et al., 2019; Lewis et al., 2020b).

6.3.6 *Experimental Setup*

Our main question is whether briefs can increase the accuracy and efficiency of fact checking. We focus on human evaluation with both crowdworkers and volunteers fact checking claims.

Human Evaluation

▸ METRICS We evaluate the *accuracy* of a fact check by comparing the verdict from our human evaluators with professionals. The professional fact checking labels are obtained from the DATACOMMONS dataset. We measure the *time* taken to fact check from when the task is loaded to when the verdict and explanation is submitted.

- CROWDSOURCED EVALUATORS Crowdworkers on Mechanical Turk are presented with the 500 test set claims and instructed to use a search bar to decide if the claim is true, false, or in the middle. They then write at least 20 words justifying their verdict. We indicate that if a claim is *mostly true* it should be labeled as true, and *mostly false* should be false. We discourage the middle option and suggest it should be used only if a verdict cannot be made, to prevent it from being the default. Previous work has shown that fine-grained labels, such as *sometimes true*, *half true*, *mostly true* are difficult to calibrate even with professional fact checkers (Lim, 2018), so we opt for a more simpler scale. The search bar queries the open web, but is restricted from searching known fact checking domains. Evaluators either use only the search bar, or are provided with a brief to read before the fact check. The same claims are evaluated with all methods. We repeat the study three times to assess variance.

- VOLUNTEER EVALUATORS Crowdsourced evaluation is scalable, but crowdworkers may be less motivated to spend a large amount of time fact checking. Thus, we conduct a smaller scale study using graduate student volunteer evaluators, recruited by asking for those interested in the challenge of fact checking real claims themselves. Volunteers are presented with 100 claims rather than 500, but otherwise conduct the same task as crowdworkers. Volunteers compare the search-bar-only fact checking process with generated QABriefs and gold QABriefs. We do not evaluate Passage Briefs or Entity Briefs, as we found volunteer fact checking to be less scalable than crowdsourcing.

Automatic Evaluation of Model Quality

To evaluate the quality of question generation, following existing work (Duan et al., 2017), we use BLEU. To evaluate the quality of question answering, we use F1 score (Rajpurkar et al., 2016).

Model Details

We use `fairseq-py` (Ott et al., 2019) to train the QABRIEFER. We use the open-sourced BART model (Lewis et al., 2020a) and suggested finetuning hyperparameters, training for 10 epochs and taking the best epoch by validation loss. To generate, we use beam search with beam size 5. We tune the length penalty to decode such that written questions and answers approximately match the average length in the validation split. Exact training and generation commands, with further experimental details, can be found in the appendix.

6.3.7 *Results*

We show in human evaluations that fact checking efficiency and accuracy are improved with briefs.

Table 6.2: **Question Generation Models.** We display BLEU scores for question generation quality.

Model	**BLEU**
Claim ⇒ Qs	12.8
Claim + Source ⇒ Qs	13.2
Claim + Source + Prev Qs ⇒ Next Q	13.4

Table 6.3: **Question Answering Models.** We display BLEU scores for answer generation quality.

Model	**F1**
BART FT on QABriefDataset	30.5
BART FT on NQ + QABriefDataset	32.8

Briefs Increase Fact Checking Quality

We examine the accuracy of crowdsourced and volunteer fact checkers when presented —in addition to a search bar— with different types of briefs: Passage, Entity, and QABriefs. For QABriefs, we examine briefs generated by QABriefer and the *Gold* briefs annotated in QABriefDataset. We compare briefs against a *search bar only* baseline.

As shown in Figure 6.6 (left), when crowdworkers are presented with briefs, fact checking accuracy increases, even when taking into account variance in three repeated trials. The Passage Briefs are not more helpful in terms of accuracy compared to using the search bar alone, but Entity Briefs and QABriefs are both better than this baseline. Providing Gold rather than generated QABriefs performs best — suggesting modeling improvements could help bridge the gap. For crowdworkers, using briefs slightly reduces the time taken (from 8.8 minutes on average to around 7), but the overall time spent is low compared to professionals, who spend from 15 minutes to one day (Hassan et al., 2015).

For volunteer fact checkers (Figure 6.6, right), accuracy across all methods is higher compared to crowdworkers. Providing the Gold QABrief remains the best, though the gap is smaller than for crowdworkers. Providing the QABrief slightly decreases time taken to fact check. Note that the average volunteer spends twice the amount of time compared to a crowdworker, and this thoroughness probably contributes to higher accuracy, as well as the smaller improvement from providing briefs.

QABriefs are Preferred

Next, we further contrast QABriefs with Passage and Entity Briefs. We ask evaluators to consider if the brief made the fact check *easier* or provided useful *background context*. Crowdworkers rated QABriefs helpful twice as often as Passage Briefs (In Figure 6.7). When evaluators submit a fact check, they must write an explanation for their reasoning. Qualitatively examining these, we found many references to the QABrief. Evaluators noted that *based on [the QABrief], I searched for [X evidence]*. We hypothesize that the question-answer format may be easier to read, as it is naturally organized and possibly less redundant.

Generating QABriefs with QABriefer

Lastly, we assess the performance of our proposed QABriefer model. We display the BLEU scores for our proposed Question Generation models in

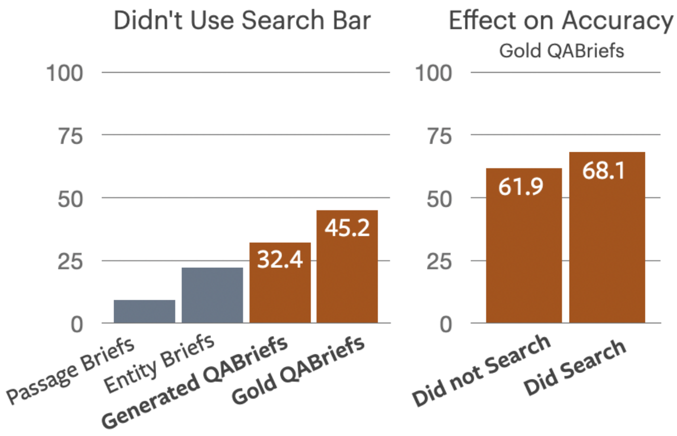


FIGURE 6.8: **Overconfidence** when given a QABrief. We find that when presented with a brief, many tend to use the search bar less, which can decrease the overall accuracy of their fact checking.

Table 6.2 and find that iteratively writing questions one by one is the best performing method. Further, providing information about the source of the claim (usually the entity who made the claim) provides better results. Question Answering results are shown in Table 6.3. We find that first fine-tuning on a large question answering dataset, Natural Questions (NQ), and further fine-tuning on QABRIEFDATASET provides the best results. Likely, this is because BART is a general purpose generative model, so fine-tuning for question answering first on a much larger dataset is useful.

6.3.8 *Discussion*

While our experiments show a generally positive impact of briefs for human fact checking, it is important to put them into a broader perspective.

- BRIEFS FOR PROFESSIONAL FACT CHECKERS Crowdworkers and professional fact checkers perform different tasks under very different circumstances. Professionals often investigate alternative interpretations and produce an explanation of their process in an article. They often have years of experience and must check a variety of claims. Consequently, we do not claim that briefs will make a difference in their work. Nevertheless, QABriefs can provide insights into the fact checking process. As the QABrief dataset was created using professional fact checking articles describing how a claim was checked, by decomposing a claim into multiple components, we can encourage a more structured fact checking process.

- BIASES INTRODUCED BY BRIEFS While briefs can increase accuracy, they can introduce biases. We found that providing a QABrief increased confidence — many submitted their fact check based on the brief alone, without the search bar. Figure 6.8 (left) displays that around 45% of crowdworkers did not use the search bar when given the Gold QABrief, even though accuracy without the search bar is reduced. Briefs aid accuracy and efficiency, but are not fully sufficient to produce a verdict.

- METRICS FOR FACT CHECKING We focus on improving fact checking accuracy, but we note that agreement amongst professionals is not 100% (Lim, 2018). Professionals often agree if part of a claim is true or false, but disagree on the importance (Lim, 2018) or pursue different directions for checking the claim (Marietta et al., 2015; Amazeen, 2016). Different fact checkers have different scales, which are not calibrated. Nevertheless, improving the accuracy of crowd sourced fact checkers is still reflective of agreement with professionals.

6.3.9 *Conclusion*

We propose the concept of fact checking briefs, to be read before performing a fact check. To create these briefs, models must generate both questions and answers. In particular, the task of answer generation requires utilizing the web to identify the most relevant evidence document, then identifying the portion of the document most relevant to producing an answer. We show in extensive empirical studies with crowdworkers and volunteers that QABriefs can improve accuracy and efficiency of fact checking.

7

Scaling Knowledge Access to Multiple Documents in Wikipedia

> *"Knowledge comes at a price ... Right? If you think so, think again ... All the knowledge is available on Wikipedia, for Free."*
> —Sandeep Sahajpal

► SYNOPSIS To create systems that can handle increasingly complex tasks, such as switching from topic to topic, text generation models must be able to access larger quantities of knowledge. We describe an open-domain dialog model capable of searching efficiently through hundreds of thousands of sentences of Wikipedia knowledge.

7.1 MOTIVATION: KNOWLEDGE FROM MULTIPLE DOCUMENTS

Moving from models using knowledge in single paragraphs to an entire document scales the input size of many Sequence-to-Sequence systems. Realistically, however, many tasks that require knowledge examine multiple difference sources. For example, users looking for information online often click through a few different search results. Writing this work requires ingesting a large body of different references, and synthesizing those different sources, consolidating that information, and producing one (not-so-succinct) thesis manuscript. Compared to digesting a single document, working with multiple sources of information has its own challenges. The scale of how many sources, how much to look through, and how efficient that search is, is of paramount importance. Beyond that, digesting multiple sources of information has a relevance challenge — models must determine which source is more relevant, particularly if certain tasks may require different amounts of knowledge. In this chapter, we propose a novel architecture capable of efficiently searching through hundreds of thousands of sentences of information, weighing the relevance of that content, and easily incorporating that knowledge in standard Sequence-to-Sequence architectures.

7.2 DIALOGUE AS A KNOWLEDGE-BASED TEXT GENERATION TASK

The ability to converse intelligently on any given topic requires knowledge. Most of us gather knowledge throughout our lives in an ambient manner — from watching television on the weekend so we can understand the conversation at lunch to speaking about or own experiences. However, many existing dialog systems rely on Sequence-to-Sequence methods that do not have explicit mechanisms for collecting this kind of knowledge. In this chapter, we focus on the task of open-domain dialog, where two individuals conduct open-ended chatty dialog given a generic topic.[1] Throughout the conversation, information is retrieved from Wikipedia to incorporate in the chit-chat, and knowledge is exchanged between the speakers. We describe an efficient, scalable methodology for a generative dialog system that can utilize multiple different sources of knowledge, in order to have knowledgeable conversations.

[1] Note that over the course of any conversation, the topic can naturally drift or broaden, or an entirely new topic of conversation could arise. Interestingly, the technique of the *non sequitur* has been proposed as a way for machine learning based chatbots to change the topic of conversation. As a point of reflection, one of the main challenges of most dialog datasets used for training these chatbots is the strong limitation of conversation length: most collected datasets contain only very short conversations, so topic drift or multiple topics in each conversation are not heavily encountered.

7.3 AUGMENTING TRANSFORMERS WITH KNN-BASED COMPOSITE MEMORY

In this section, we describe a general purpose architecture for (1) identifying information and (2) incorporating multiple sources of knowledge into standard Transformer models. The architecture leverages the scalability and differentiability of k nearest neighbors search to fetch information relevant for text generation tasks. To incorporate multiple sources of information and knowledge in multiple modalities, multiple KNN search modules can be added to the generation system. Each search module would retrieve information from a different knowledge source, and the final model would blend these together. We apply this architecture to two different chit-chat dialog datasets, to create open-domain, generative chatbots capable of discussing any topic, including images.

7.3.1 *Introduction*

Machine learning approaches to various tasks, such as game-playing or dialog, are often dependent on external information. This information can take multi-modal forms, including structured knowledge bases, free text, and images, and also comes in overwhelmingly large quantities. A pressing challenge is to create models that can identify which specific elements of multiple information sources are relevant in a particular context, and incorporate them into standard architectures on each task. In this work, we focus on human-machine dialog and how to efficiently retrieve external knowledge that is relevant to the dialog. We consider two scenarios and for each scenario, retrieve two types of knowledge: (i) knowledge about similar dialog contexts and (ii) external knowledge used to ground the conversation into real world information.

Knowledge about similar dialog contexts allows for a hybrid retrieval and generative approach to dialog where the system response is generated based not only on a representation of the current dialog context and of the relevant world knowledge, but also based on a response retrieved from a similar dialog context. The retrieved knowledge can be viewed as providing information about structure and dialog sentences, or utterances: which response is likely given a similar context?

External knowledge is also retrieved to improve the semantic content of the dialog model. In one scenario, Wizard of Wikipedia (Dinan et al., 2018), general topics are provided to crowdworkers, who are asked to have in-depth and specific conversations about these topics by referencing specific Wikipedia sentences as knowledge. In this scenario, external knowledge is retrieved from a pre-selected set of Wikipedia sentences associated with the current dialog topic. Retrieval aims to select the sentence that is most relevant at each step of the dialog and thereby to ground system responses in relevant world knowledge (e.g. by referring to Star Wars when talking about science fiction).

In the other scenario, Engaging ImageChat (Shuster et al., 2020), crowdworkers are provided with images and asked to have a conversation inspired by or about the image. In this case, the retrieved external knowledge is images and their associated dialogs. By retrieving images that are similar to the image being talked about, we aim to enrich system responses with knowledge about what is typically mentioned when describing similar images (e.g. when talking about an image with dogs, mentioning their breed).

Our work on incorporating different types and modalities of knowledge is related to methods that strive to add external memory, such as knowledge bases, to neural networks. Previous work has explored incorporating large external memories into neural network layers (Weston et al., 2015; Sukhbaatar et al., 2015, 2019b; Lample et al., 2019). Many existing approaches focus on using attention over the memory slots, which is computationally intensive and becomes less effective as the the size of the memory grows. In this work, we propose representing multiple sources of external information as fixed encodings and using K Nearest Neighbors search to fetch relevant information. KNN search is computationally efficient and scalable, and libraries like `faiss` (Johnson et al., 2019) allow KNN to be easily used on GPUs and integrated into neural networks. Further, the external memories are pre-encoded, so the information encoding is only computed once. As the external memories are kept fixed, they do not require any training to learn the memories along with the model. We can thus scale easily to larger memories by learning only the KNN-based read operation to identify relevant information from the memory.

Our core contribution proposes an efficient, KNN-based Information Fetching (*KIF*) module that can access relevant external knowledge, combine knowledge from different sources, and integrate this information into standard sequence to sequence architectures. We apply these flexible modules to two dialog datasets that challenge generative models to leverage external information to write coherent, on-topic responses. Both of our chosen tasks require models to leverage external information, such as information from Wikipedia or images, to engage in the conversation. We show that relevant information can be identified from hundreds of thousands of candidates in a multi-modal, multi-knowledge-source setting to improve the performance of generative dialog models. Further, the output of the KIF modules is interpretable as specific human-readable knowledge elements are selected, allowing users to better understand the information the generative model conditions upon when writing the subsequent utterance. On both datasets, we achieve state-of-the-art results compared to generative models and find there is no statistically significant difference in the interestingness or human preference of our model output compared to state-of-the-art retrieval models.

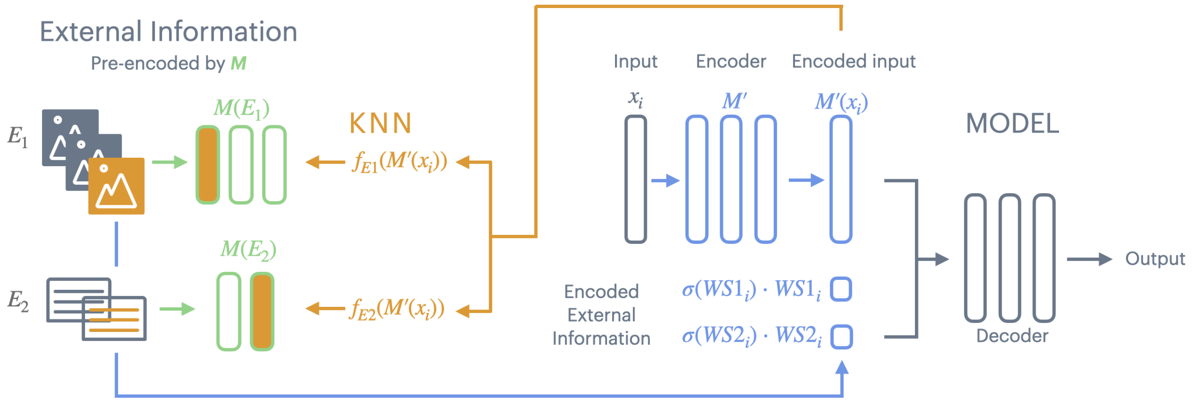


FIGURE 7.1: **KIF** modules fetch relevant information from multi-modal external knowledge. External knowledge sources E_1 and E_2 are pre-encoded by encoder M (green). In the model, input x_i is encoded by encoder M' (blue) to produce $M'(x_i)$. KIF modules (orange) operate on $M'(x_i)$ and identify the nearest neighbors encoded in $M(E_1)$ and $M(E_2)$ using KNN. Identified relevant elements from E_1 and E_2 are re-encoded by M' in a gating mechanism with a weighted sum (represented by $\sigma(\mathtt{WS1}_i) \cdot \mathtt{WS1}_i$, where `WS` stands for weighted sum), then concatenated to $M'(x_i)$. Full description with notation can be found in Section 7.3.3.

7.3.2 *Related Work*

We discuss related work on learning to incorporate external knowledge into neural networks and efficiently access relevant information. We then describe work in generative dialog that incorporates knowledge.

Incorporating External Knowledge

Augmenting neural networks with memory, or longer term components that can be accessed with read and write operations, has been explored in various proposed architectures. For example, Memory Networks (Weston et al., 2015; Sukhbaatar et al., 2015, 2019b) introduce attention mechanisms over large external memories. Neural cache models (Grave et al., 2017b) simplify these to access previous memories with a dot product. Previous work has also studied how to read and write into these memory architectures (Rae et al., 2016; Graves et al., 2014; Joulin and Mikolov, 2015). In contrast, we focus on how to read large memories.

Another line of research has focused on computational scalability for larger external memories to allow efficient access of information. For example, Chandar et al. (2016) propose a hierarchical memory network rather than a flat one and Rae et al. (2016) learn sparse operations to read and write. Lample et al. (2019) focus on learning memories of up to one million slots and how to efficiently access the slots using product keys. Khandelwal et al. (2019) use nearest neighbor operations to augment language models by performing retrieval at the token level — in contrast, we focus on multi-modal retrieval of multiple pieces of knowledge based on an entire dialog context. Beyond explicit memory representations, it may be possible to store information implicitly during training time by memorizing common patterns present in text (Petroni et al., 2019). We focus on learning to fetch relevant information from multiple explicit external multi-modal knowledge sources and integrate them into one network. Further, our work allows the retrieved information to be interpreted as each memory slot is an explicit fact that can be read as text, rather than a learned vector such as in Lample et al. (2019).

Work has also focused on computationally efficient softmax operations (Mnih and Hinton, 2009; Grave et al., 2017a; Chen et al., 2016). Many approximate softmax techniques use KNN-like operations to form clusters, and the overall softmax operation is constrained by the slow calculation of the exponential. Our usage of KNN benefits from efficient and scalable libraries such as `faiss` and `nmslib`.

Generative Dialog

We develop a general architecture for incorporating external information and apply it to the case of generative dialog models. Previous work in dialog has leveraged knowledge as necessary information to accomplish the task. For example, airline and restaurant booking tasks often use API calls to access information about reservation times and availability (Bordes et al., 2017). In contrast, our work focuses on how to incorporate unstructured knowledge, such as free text found on the web. Previous work has employed architectures that attend over the available knowledge and identify relevant pieces of information, which scales poorly with large quantities of information (Dinan et al., 2018; Qin et al., 2019; Lian et al., 2019). We replace the use of attention over external information with the output of a KNN module. Other work has investigated incorporating information retrieval in language modeling and question answering (Chen et al., 2017; Fan et al., 2019a; Seo et al., 2019; Guu et al., 2020), while we focus on dialog applications and flexibly incorporating knowledge from multiple, multi-modal sources.

On the modeling side, work has explored both generative (Serban et al., 2016b,a) and retrieval based models (Zhang et al., 2018b), which identify the best utterance from the training set to return as the dialog response. This often leverages self-attention or cross-attention mechanisms (Humeau et al., 2019). Further work has explored hybrid models, for example using the output of a retrieval model as input for a generative model (Dinan et al., 2018; Weston et al., 2018; Cai et al., 2019; Zhu et al., 2020). Some of this work has specialized to use both types of models to generate conversations in an ensemble (Song et al., 2016) or to specifically improve consistency (Song et al., 2020). We extend these approaches by augmenting generative models with retrieval-like operations based on KNN search, allowing dialog models to flexibly incorporate various sources of external knowledge at the same time and scale to large quantities of retrieval candidates.

7.3.3 *KNN-based Information Fetching Modules*

Broadly, the KNN-based Information Fetching (KIF) module assumes an encoder model M can access inputs $\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$. For example, $\boldsymbol{X}$ can be a collection of sentences, and x_i represents an individual sentence. In a setting without additional supporting information, the encoder will process an input x_i and produce the encoder output $M(x_i)$. If x_i is a sequence such as a sentence, then $M(x_i)$ is a representation of the variable size of the sequence length by the fixed size encoder M's hidden size. However, in many tasks, additional information is present, represented as $\boldsymbol{E} = \{e_1, e_2, \ldots, e_m\}$. We encode each element of $\boldsymbol{X}$ and $\boldsymbol{E}$ into a vector representation using the encoder. To identify the closest information in $\boldsymbol{E}$ that is relevant to x_i, our general approach will be to use K Nearest Neighbors by comparing the representation of x_i with the representation of each element in the set $\boldsymbol{E}$. K Nearest Neighbors is a fully differentiable operation (Plötz and Roth, 2018), so can be incorporated in a straightforward way into neural models. The most relevant information in $\boldsymbol{E}$ will then be available in the model. We display a KIF-Augmented model in Figure 7.1 and describe how the KIF module operates.

One challenge to overcome is that the representation of all elements of the knowledge source $\boldsymbol{E}$ are pre-computed and kept fixed, creating $M(\boldsymbol{E})$ — we do not backpropagate to affect the embeddings of the pre-encoded knowledge. In the early stages of training, the model receives large amounts of loss, which would affect the quality of the pre-encoded embeddings if we backpropagated to them. Further, encoding the fixed external knowledge once and re-using it allows for greater scalability. However, this lack of backpropagation can introduce a mismatch between the encoding of $\boldsymbol{E}$ and the encodings produced by a model that is training, as the training model has constantly changing representations because the weights are being learned. We use M to represent the original encoder model used to encode $\boldsymbol{E}$ and M' to represent the constantly training model that is encoding $\boldsymbol{X}$. The model must learn a function to align $M'(x_i)$ to the pre-encoded elements of the external memory $M(\boldsymbol{E})$.

To circumvent this misalignment, we learn a mapping operator $f_E(M'(x_i))$ that trains to map elements of the model's representation of $\boldsymbol{X}$, or $M'(\boldsymbol{X})$, into the additional information representation space $M(\boldsymbol{E})$. Concretely, $f_E(M'(x_i))$ is a multi-layer perceptron with ReLU nonlinearities. From the input elements of $\boldsymbol{X}$, $f_E(M'(x_i))$ learns representations of an output close to the corresponding projection of $\boldsymbol{X}$ into $\boldsymbol{E}$. This can be interpreted as learning a read operation on a fixed external memory. If there was no change to the encoding of the model compared to the pre-computed knowledge, then the ideal mapping operator would be the identity function (as M' would equal M). However, as the model changes significantly during the training process, the nonlinear mapping capability of $f_E(M'(x_i))$ is essential to be able to identify the correct knowledge $\boldsymbol{E}$ from the input $\boldsymbol{X}$.

Thus, a model augmented with KIF will incorporate external knowledge in the following manner. First, we find the k nearest elements to $f_E(M'(x_i))$ in $M(\boldsymbol{E})$, based on KNN search with inner product. Then, the relevant elements identified by KNN are re-encoded by M'. For example, if element e_j is retrieved by KIF, it would produce $M'(e_j)$. We use the optimized `faiss` library for KNN search, which can conduct billion-scale KNN efficiently on GPUs.

The KNN output for an element x_i is produced by using `faiss` to search for the k nearest representations to $f_E(M'(x_i))$ in $M(\boldsymbol{E})$. Note that as the encoders M and M' produce output representations of variable length (for example, in the case where x_i is a variable length sequence, such as a sentence), we average across the length dimension to produce a fixed-size representations r to conduct the KNN search.

$$r_{x_i} = \texttt{Avg}\big(f_E(M'(x_i))\big) \tag{7.1}$$

$$\boldsymbol{R}_E = \big\{\texttt{Avg}(M(e)) \mid e \in \boldsymbol{E}\big\} \tag{7.2}$$

$$\texttt{KNN}_{x_i} = \texttt{KNearest}\big(k, r_{x_i}, \boldsymbol{R}_E\big) \tag{7.3}$$

Then, the KIF module output for an element x_i is the set of all re-encoded representations of the KNN-retrieved knowledge:

$$\texttt{KIF}_{x_i} = \left\{M'(e) \mid e \in \texttt{KNN}_i\right\} \tag{7.4}$$

These elements are weighted by their normalized nearest neighbor scores and then summed. This is subsequently concatenated to $M'(x_i)$ to form the final encoder output:

$$[M'(x_i), \texttt{WeightedSum}(\texttt{KIF}_i)] \tag{7.5}$$

This can be easily extended to using multiple modules simultaneously. For instance, two sources of external information, $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$, can be combined by identifying the top candidates of each information source. The weighted sum of the KIF output on each information source is concatenated with the encoded input $M'(x_i)$. The KIF output dimensionality is the same size as the hidden size of $M'(x_i)$, so they can be directly concatenated.

Finally, different sources of information may not be required for every prediction and some information sources can be more important than others. To allow the model to make more fine-grained decisions about what information to use from what source, and how much of it, we add a gating mechanism using a sigmoid function around each weighted sum of KNN representations. $\texttt{KIF1}_i$ and $\texttt{KIF2}_i$ denote the KIF module from Equation 7.4 applied to $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ respectively.

$$\texttt{WS1}_i = \texttt{WeightedSum}(\texttt{KIF1}_i) \tag{7.6}$$

$$\texttt{WS2}_i = \texttt{WeightedSum}(\texttt{KIF2}_i) \tag{7.7}$$

which produces the final encoder output, a concatenation of $M'(x_i)$ with the output of multiple KIF modules:

$$\left[M'(x_i), \;\; \sigma(\texttt{WS1}_i)\cdot\texttt{WS1}_i, \;\; \sigma(\texttt{WS2}_i)\cdot\texttt{WS2}_i\right] \tag{7.8}$$

This concatenation represents the output of the encoder M' and can be used for various purposes, such as providing the encoder output to a decoder in a sequence to sequence model.

7.3.4 *Applying KIF to Dialog Tasks*

We describe how to apply KIF to the task of generative dialog, a setting where models must generate engaging and on-topic responses. We investigate dialog for two reasons: first, dialog agents must be able to consult relevant information to maintain the topic of the conversation. Second, retrieval-based agents have strong performance compared to generative ones, due to their ability to copy dialog utterances from the training set. Using KIF, we can incorporate the benefits of retrieval architectures into generative, knowledge-based models.

KIF for Generative Dialog

In dialog, x_i represents the text of the conversation i. A conversation consists of multiple back-and-forth *utterances* (or turns). For example, a conversation could consist of 4 turns: $x_i = [x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}]$ where $x_{i,4}$ is the direct utterance the model should respond to, and the earlier utterances are the *conversation context*.

Standard generative dialog models use a Transformer neural network as the encoder M and want to produce an output that is an appropriate response to the conversation. However, in many cases, the conversation history alone does not include all of the information required to produce an appropriate response. For example, if a model needs to chat about a specific movie, it can be helpful to provide the model with more information about that movie so a more interesting dialog response could be produced. To incorporate knowledge, models often concatenate a knowledge source $\boldsymbol{E}$ such as Wikipedia to x_i and use attention modules to identify the most relevant knowledge. However, this approach is computationally intensive when handling large quantities of information. Further, attention mechanisms have been found to operate poorly over long sequences, as the mechanism becomes blurry due to the softmax and struggles to make fine-grained decisions (Fan et al., 2018b). The same is true for hierarchical approaches, which lack scalability.

We augment Transformer sequence to sequence (seq2seq) networks on the encoder side with KIF to improve generative dialog models. We experiment on two dialog tasks, Wizard of Wikipedia (Dinan et al., 2018) and Engaging ImageChat (Shuster et al., 2020). In both datasets, models must leverage information external to the dialog history alone — in Wizard of Wikipedia, the chat requires access to knowledgeable facts and in Engaging ImageChat, discussion about a specific image. As models must process multiple inputs and ground responses in the knowledgeable facts or images, these tasks challenge existing seq2seq approaches.

Wizard of Wikipedia

The goal of the Wizard of Wikipedia dataset is to train knowledgeable agents that can chat in any domain. The dataset contains 1,365 various topics discussed in 18,430 dialogs in the training set, totalling 166,787 training utterances. Each topic is a general concept, such as *dogs* or *ice cream*, and is included as the first utterance of the conversation. The conversation is meant to be in-depth and detailed, so individual utterances must reference specific knowledge as a basis for the utterance. The knowledge takes the form of Wikipedia sentences. For example, the chat utterance *I love Toy Story! It was released in 1995* would reference the Wikipedia sentence *Toy Story is a 1995 American computer-animated buddy comedy [...]*. For each utterance, a set of sentences are identified by an information retrieval system, and the crowdworker selected one knowledge sentence as the basis for their utterance.

▸ KNOWLEDGE SOURCES. Our model for Wizard of Wikipedia has access to two sources of external information, $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$:

- $\boldsymbol{E}_1$ *is Wikipedia Knowledge* provided by the dataset as evidence to support knowledgeable chitchat (initially curated by the information retrieval system used in Dinan et al. (2018)). The scale of this KNN search is to filter through an average of 34 sentences. The KIF module uses dialog features to fetch relevant knowledge to condition upon to generate the subsequent utterance.

- ***E_2** is Training Utterances*. To incorporate the benefits of retrieval-based dialog models to the generative setting, we use KIF to identify relevant utterances from the training set and take their *responses* as input. If many conversations about dogs have already occurred, models should be able to take advantage of these human-written examples to improve their generations. For example, likely conversation could occur about the breed of the dog, daily routine with a pet, and similar topics. There are around 170K dialog utterances as inputs to KNN search. This can be interpreted as incorporating the benefits of retrieval models by identifying an utterance with similar structure as the text the model would like to generate. We do not allow the module to fetch the correct response of the current conversation context.

Access to these two sources of knowledge can be seen as learning a template and a topic separately. Sample templates can be identified from the training utterances, and topic-specific information learned by accessing the Wikipedia knowledge.

► ADDITIONAL KNN FEATURES. To better identify relevant training utterances from the large quantity available, we break down x_i into conversation sub-features for a more fine-grained match in the KNN search step. By conducting KNN on more features, we can achieve higher quality retrieval. We leverage the nature of dialog to decide these features.

We concatenate the encoding of the most recent dialog utterance (e.g. $x_{i,\text{last}}$) with the encoding of the dialog context from the current conversation and the turn number t, such that $M'(x_{i,\text{last}}), M'(x_{i,-\text{last}}), t$ is the representation used for KNN search. Concretely, if the model is trying to produce the 5th turn of the conversation, then $x_{i,\text{last}}$ is the most recent utterance from the dialog partner, $x_{i,-\text{last}}$ would be the last 3 turns of exchange, and t would be 4. Note that the turn number is represented as a standalone number. These are known to be salient conversation features. The most recent dialog utterance is the direct turn the model is responding to, and the dialog context may provide additional clues. The turn number is important, as earlier turns are often generic (e.g. *how are you doing today*) and later turns are more specific.

Engaging ImageChat

The goal of Engaging ImageChat is to create agents capable of chitchatting about images selected from the YFFC100M dataset (Thomee et al., 2016). The dataset contains 186,782 dialogs in the training set, each about a unique image, totalling 355,862 utterances. Agents are assigned one of 215 personalities (e.g. *sweet, caring, excited*) to increase engagingness. Previous work (Shuster et al., 2020, 2019) identified that both crowdworkers and models, when provided with personalities, produced more diverse, interesting responses, as evaluated by humans.

We use a Multi-Modal neural network designed to handle both image input and text input. Following Shuster et al. (2020), the images are encoded

TABLE 7.1: Results on the **Wizard of Wikipedia** dataset. We implement the Retrieve and Refine and Response Generation with MR approaches, all with Reddit Pre-Training, and evaluate them on Wizard of Wikipedia. The *Seen* test set consists of conversations on topics seen at training time, and the *Unseen* test set consists of conversations about new topics that were not in the training set.

Model	Test F1 (Seen)	Test F1 (Unseen)
Retrieval Baselines		
Retrieval Transformer MemNet (Dinan et al., 2018)	15.4	12.4
Generative Baselines		
2-Stage Generative MemNet (Dinan et al., 2018)	18.9	17.4
Generative Transformer MemNet (Dinan et al., 2018)	16.9	14.4
+ Reddit Pre-Training	17.6	16.3
Retrieve and Refine (Weston et al., 2018)	18.2	17.9
Response Generation with MR (Qin et al., 2019)	17.5	16.8
KIF-Augmented Transformer	**25.9**	**22.3**

using a pre-trained ResNeXt network (Xie et al., 2017). To extract the final image representation, we project the 2048-dimensional output of the image encoder to 512-dimensions using a deep multi-layer perceptron with ReLU activation units. The conversation history, which includes the one-word personality, is encoded with a Transformer encoder network. The image and conversation are integrated using the Multimodal-Sum-Combiner module proposed in Shuster et al. (2020).

► KNOWLEDGE SOURCES. Our model for Engaging ImageChat has access to two sources of external information, $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$:

- $\boldsymbol{E}_1$ *is Chat on Similar Images*. While there are over 180K different images in this dataset, many of the images are similar. For example, conversations associated with two pictures of dogs could be relevant to each other. The model is able to use KIF directly on the current image features to fetch from around 180K different images and return 6 turns of related chat for each fetched image. Fetching from $\boldsymbol{E}_1$ consists of identifying related image chats, or conversations on related topics.
- $\boldsymbol{E}_2$ *is Training Utterances*. Similar to the motivation for the previous dataset, we allow the model to identify training utterances that could be useful for responding in the current conversation. The scale of this fetching task is large: 350K dialog utterances. This could be interpreted as identifying utterances with similar structure to what the model would like to generate, and is complementary to the topic-based related image chats.

► ADDITIONAL KNN FEATURES. To identify relevant information from training utterances, we use the same dialog features as Wizard of Wikipedia in the KNN search step, with one modification: we add the *personality* provided by the dataset. We represent the personality feature as the personality word, such as *caring*, and embed it with the encoder M'. As utterances from speakers with the same personality are more likely to be similar, this feature improves the quality of the fetched information. For example, conversations with the *sweet* personality often include similar text such as *aww, that's wonderful*. We use two additional features for the KNN search: t, the turn

Model	Test F1
Retrieval Baselines	
Retrieval Transformer (Shuster et al., 2020)	9.8[2]
Generative Baselines	
Generative Transformer MemNet (Dinan et al., 2018)	7.1
+ Reddit Pre-Training	12.8
Retrieve and Refine(Weston et al., 2018)	13.6
Response Generation with MR (Qin et al., 2019)	13.2
KIF-Augmented Transformer	**14.4**

TABLE 7.2: Results on the **Engaging ImageChat** dataset. We implement the Generative Transformer Memory Network, Retrieve and Refine, and Response Generation with MR approaches, all with Reddit Pre-Training, and evaluate them on Engaging ImageChat.

number, and p, the personality. This feature is explicitly used in Shuster et al. (2020) to improve the engagingness and flow of the conversation. Similar to Wizard of Wikipedia, we represent the conversation turn t as a number. The Transformer model is used to encode text x_i and produce a representation of the text, then the turn number t and personality p are represented separately. As the personality is a word, we use the same Transformer to encode it. The concatenation of features used for KNN search is: $M'(x_{i,\text{last}}), M'(x_{i,-\text{last}}), t, p$.

7.3.5 *Experimental Setup*

Implementation Details

- PARAMETER SETTINGS. We use `parl.ai` (Miller et al., 2017) to implement our models.[3] The data for both datasets used is available for download from `parl.ai` as well. We use byte-pair encoding (Sennrich et al., 2016) to represent the text to better handle the rare word problem (Dinan et al., 2018; Fan et al., 2018a). Our generative Transformer models have 8 encoder layers and 8 decoder layers, with FFN size 2048, embedding dimension 512, and 4 attention heads. We optimize using Adam (Kingma and Ba) and the inverse square root learning schedule (Vaswani et al., 2017) with 10k warmup updates. The initial learning rate is 0.0001 and we optimize for model perplexity. We use a dropout of 0.5 and set gradient clipping to 0.1. We set k = 5 for all cases. For both datasets, we model a vocabulary size of 54944 based on the byte-pair-based vocabulary[4] from the Reddit pre-training. We tuned the learning rate and batchsize hyperparameters together.

- PRE-TRAINING. We pre-train the Transformer seq2seq model used for both datasets on 250M comments from Reddit. The Reddit dataset was made available by `pushshift.io`. The comments are parsed to maintain conversational threads of users responding to each other, so the encoder network has been exposed to conversational context at training time. Note that the Reddit dataset does not include aspects such as personality, as those are unique to specific datasets such as Engaging ImageChat. The context size in pre-training is set to 512 tokens. The ResNeXt encoder used to model

[2] In Shuster et al. (2020), retrieval Transformer models report Hits@N using a fixed candidate set of 99 distractor candidates and 1 true candidate. We compute F1 using their open-sourced model by scoring the entire training set of over 350K utterances with the model and taking the top scoring candidate as the response.

[3] `https://github.com/facebookresearch/ParlAI`

[4] See Sennrich et al. (2016) for an explanation of the byte-pair encoding technique.

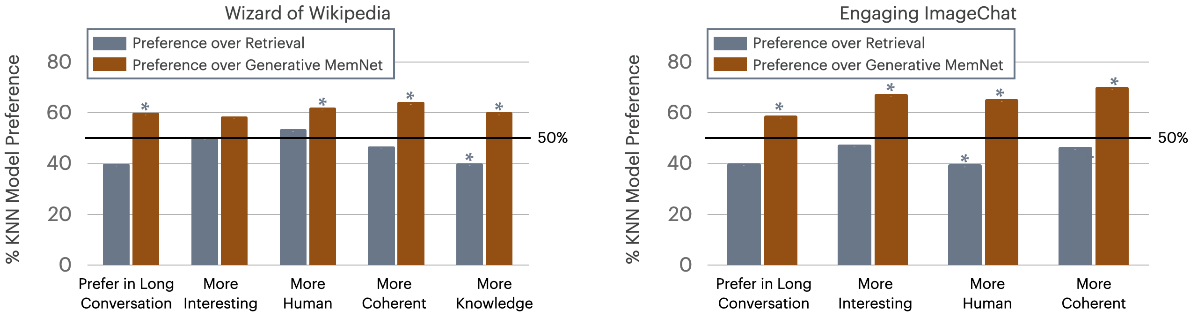


FIGURE 7.2: **Human Evaluation Results on Both Datasets**. More than 50% indicates the KNN Model is preferred. Stars indicate statistical significance at $p < 0.05$.

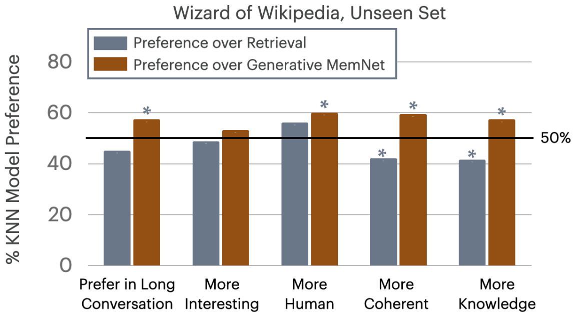


FIGURE 7.3: **Human Evaluation on the Unseen Test Set** of Wizard of Wikipedia. More than 50% indicates the KNN Model is preferred. Stars indicate statistical significance at $p < 0.05$.

images for the Engaging ImageChat dataset was pre-trained on 3.5 billion images (Mahajan et al., 2018).

Evaluation

- GENERATION. We generate with beam search, setting the beam size to 4. We use 3-gram blocking. This technique disallows repeated n-grams from being generated multiple times and reduces repetition.

- AUTOMATIC METRICS. Following Dinan et al. (2018), we compute *F1*, a metric of unigram overlap, between the generated utterance and the human-written reference utterance from the dataset. For generative models, utterances are generated using beam search. For retrieval models, the next utterance is predicted by ranking the entire set of training utterances, and the highest scoring utterance is chosen.

  In Wizard of Wikipedia, there are two test sets: a set of *seen* topics, or topics that have been seen at training time with new test-time dialogs. The second set is *unseen*, or topics that have not been encountered at all during training time. We evaluate on both subsets.

- HUMAN EVALUATION. We follow the setup and use the analysis questions proposed in the `Acute-Eval` dialog evaluation system (Li et al., 2019). For reproducibility, we adopt this existing evaluation setting that has been applied to several dialog datasets. We use the question wording suggested by `Acute-Eval` and follow their self-chat procedure and interface. As one of the original datasets assessed in this system was Wizard of Wikipedia,

---

Human: I love watching disney movies! Do you like Disney?
Model: I love disney, I love watching disney movies and different animations and characters that make them unique
Fetched Knowledge: *disney consumer products and interactive media - the company specializes in producing animated films*
Fetched Training Utterance: *i love music, I love going to music shows and dancing and having fun*
Human: What's your favorite disney movie? There are so many good ones.
Model: I love the incredibles, they are my favorite disney movie.
Fetched Knowledge: *disney announced intentions to develop additional superhero films after the success of the incredibles*
Fetched Training Utterance: *i love kiteboarding, it is one of my favorite activities on the water.*
Human: omg, me too! the second one just came out i think. I don't watch a lot of sequels though, I think the last one I saw was toy story.
Model: That's cool! I remember when toy story was first released in 1995.
Fetched Knowledge: *talk of a second film following toy story began around a month after the film's opening, in december 1995*
Fetched Training Utterance: *that's neat! i like so many of her songs! i remember when her first album was released in 2008.*
Human: I've actually never seen toy story before. Kind of embarrassing...
Model: Ohhh ... I think toy story is a classic? It was first released in 1999.
Fetched Knowledge: *toy story 2 is a 1999 american computer-animated comedy film produced by pixar animation studios for walt disney pictures*
Fetched Training Utterance: *Ohhh ... What do people with color blindness do to cope with the effects?*

FIGURE 7.4: **Conversation Between Human and KIF-Augmented Transformer on Wizard of Wikipedia**. The top-scoring Wikipedia knowledge and training utterances fetched by KIF are displayed with model output.

their evaluation setting extends naturally to ours. We collect 100 human-bot conversational dialogs on a crowdsourcing platform for both datasets. The dialogs are eight turns long. Then, we show pairs of the collected conversations side by side, one conversation with a human and model A and the other conversation with a human and model B. We ask annotators the following questions:

- Who would you prefer to talk to for a long conversation?
- If you had to say one of the speakers is interesting and one is boring, who would you say is more interesting?
- Which speaker sounds more human?
- Which speaker has more coherent responses in the conversation?
- If you had to say that one speaker is more knowledgeable and one is more ignorant, who is more knowledgeable? (Wizard of Wikipedia only)

We measure the percentage of time one model was chosen over the other, taking the majority agreement between three evaluators. To reduce variance, dialogs paired in the evaluation were collected on the same topic for Wizard of Wikipedia and collected on the same image and personalities for Engaging ImageChat. Topic and images selected for evaluation are unique and taken randomly from the test set.

Baselines

We compare Transformers augmented with KIF to other existing approaches on Wizard of Wikipedia and Engaging ImageChat. The best approaches, judged by human evaluation, are retrieval models, the Retrieval Transformer Memory Network from (Dinan et al., 2018) and the Retrieval Transformer from Shuster et al. (2020). These have been shown to be strong baselines compared to other retrieval techniques based on TF-IDF (Chen et al., 2017). Thus, we report the existing retrieval models for both datasets, but focus on comparing to other generative baselines.

We compare to three additional generative baselines. Note that in Wizard of Wikipedia, the construction of the dataset is that sentences of Wikipedia

Human:	Hey, how are you doing
Fetched Training Utterances:	*I'm great, thanks for asking. Craving some chocolate. Do you like chocolate?* *Hello, how is it going? I know some trivia about this movie* *Hello, it's lunch time here, and I'm in the mood for a great steak*
Model Response:	Hello, I'm great, thanks for asking
Human:	What are your hobbies?
Fetched Training Utterances:	*I work at an elementary school. I hope you find a job you love too [...]* *I have a hound, we just got her. Although, I grew up with Labrador Retrievers.* *I just love ice cream. I love the types with fruits and flavours. Do you like ice cream?*
Model Response:	I work at a tennis court. I like to play tennis. Do you like your hobby?
Human:	hi buddy, what do you think about cinematography?
Gold Chosen Knowledge:	*cinematographers use a lens to focus reflected light from objects into a real image [...]*
Fetched Knowledge:	*cinematography is the art of motion-picture photography* *typically, a lens is used to repeatedly focus the light reflected from objects [...]* *the modern photographic camera evolved from the camera obscura*
Model Response:	I love cinematography, it is my hobby. It is a kind of motion-picture photography.
Human:	Speaking of blue skies, have you seen the 1946 movie staring bing crosby?
Gold Chosen Knowledge:	*blue skies is a 1946 american musical comedy film [...] and starring bing crosby [...]*
Fetched Knowledge:	*blue skies is a 1946 american musical comedy film [...] and starring bing crosby [...]* *blue skies the band has since broken up* *blue skies was was composed in 1926 as a last - minute addition to betsy the musical*
Model Response:	Oh cool, that's an American comedy film starring Bing Crosby. Do you like watching movies?

FIGURE 7.5: **Examples of Top-3 Fetched Training Utterances and Fetched Knowledge** when responding to a human chat from the dataset using a trained Wizard of Wikipedia model. Examples are taken from validation.

knowledge are provided with the utterances in a concatenated form. Models must identify the relevant information in this provided knowledge, or can access more Wikipedia knowledge beyond the provided sentences. The following baseline methods always have access to the information provided in the dataset already, but no additional Wikipedia knowledge beyond that.

- *Transformer Memory Networks*. To contrast the ability of KIF to existing work, we compare our models to published Transformer Memory Networks (Dinan et al., 2018). These models encode each piece of external information independently with a Transformer Encoder, and these are stored as memory slots. To access information in the memory slots, a model performs dot-product attention between the memory slots and the dialog context. In Dinan et al. (2018), the knowledge selection from Wikipedia was supervised with either *(a)* a two-stage model where the first model was trained to predict the right knowledge and a second model conditions on the predicted knowledge to generate the next utterance, or *(b)* an end-to-end model with an auxiliary loss for knowledge prediction accuracy.
- *Retrieve and Refine*. We implement a hybrid model (Weston et al., 2018) that incorporates top retrieval candidates as additional input to Generative Transformer MemNets. In Retrieve and Refine, a fixed number of candidates are retrieved and concatenated to the conversational history in the encoder, making the input much longer. For both datasets, the Retrieve and Refine mechanism that fetches a fixed number of training utterances is added to the Generative Transformer MemNet with Reddit Pre-Training baseline.

  Unlike the KIF-Augmented Transformer, the retrieval is conducted with a separate model so there is no backpropagation to affect the retrieval.

With KIF, models can alter the retrieved candidates by learning the mapping operator. Further, a fixed amount of information is always retrieved, without the capability to easily rescale to focus on specific candidates. KIF modules have weighting mechanisms to focus more on certain information, and the modules are combined with gating so models can learn which knowledge sources are more important and adjust flexibly. Lastly, Retrieve and Refine is only used to retrieve one source of information: training set utterances.

- *Response Generation with MR*. We implement the model proposed in Qin et al. (2019), which encodes the conversation history and document contextually with a biLSTM before generating the next dialog utterance. The initial model was applied to a machine reading task where a knowledge document was provided along with the conversation history. For Wizard of Wikipedia, we replace the knowledge document with the Wikipedia sentences provided in the dataset. The model then uses the conversation to identify the most relevant information in the document using a cross-attention mechanism. For the Engaging ImageChat dataset, as there is no document provided with the dataset, we replace the expected document with the conversation history, and use the most recent utterance in the conversation to attend to the conversation history.

  We make an additional improvement to this baseline: in Qin et al. (2019), the embeddings used pre-trained CoVE vectors (McCann et al., 2017). We found our Reddit pre-trained Transformer embeddings to work more effectively as they are trained for dialog. Thus, we replace CoVE embeddings with domain-specific ones.

All of Transformer generative baselines are initialized with the same pre-training on Reddit that we use for our models for fair comparison on modeling quality.

7.3.6 *Results*

We describe the results of incorporating KIF modules into Transformer networks. We display an example conversation between a human and our model in Figure 7.4, and show the top scoring Wikipedia knowledge and Training Utterance fetched by KIF modules. We compare to various baselines using automatic and human evaluation, and discuss our experiments. We present various ablation settings to understand the key features that make our method function.

KIF is Effective for Incorporating Knowledge

▸ AUTOMATIC EVALUATION. Comparing KIF augmented Transformer networks to published baselines and Retrieve and Refine, we find improved results.

For Wizard of Wikipedia, the improvement in F1 score over the best baseline is around 8 points (see Table 7.1). A major contributing factor is the construction of the dataset — as each dialog turn is grounded in a specific knowledge sentence from Wikipedia, improving the ability to identify the relevant fact strongly improves performance. Contrasting the results from the *seen* and *unseen* test sets in Table 7.1, the improvement on *unseen* is worse — it is harder to fetch training utterances for unseen topics.

While Imagechat has no explicit dependency on knowledge, we still see a 2 point improvement compared to the Generative Transformer MemNet (with the additional Reddit pre-training), indicating that KIF can be generally useful (see Table 7.2). Compared to an even stronger baseline that we tune in this work, Retrieve and Refine, we see 1 point improvement.

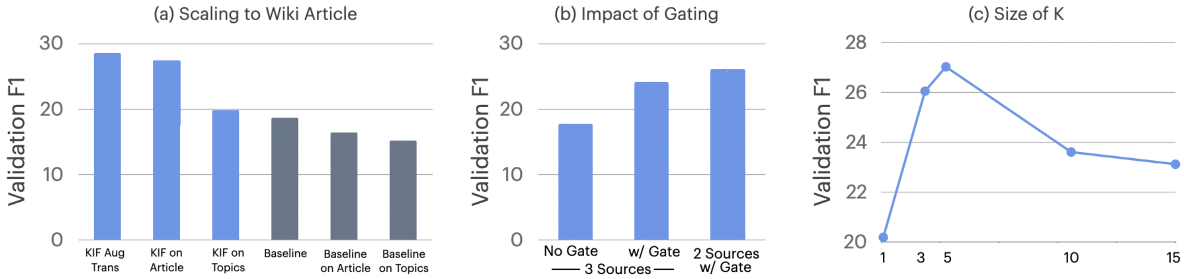


FIGURE 7.6: **Ablations** on Wizard of Wikipedia. (a) KIF can **scale** to hundreds of relevant sentences (blue) while the baseline model, the Generative Transformer MemNet (gray), scales poorly (b) **Gating** can remove irrelevant information. In the 3 Sources case, one source of external information is unrelated. (c) **Performance as** *k* **varies.**

▸ HUMAN EVALUATION. Results are shown in Figure 7.2. On both datasets, we find there is large improvement over existing generative models (green bars) that is statistically significant for some of the evaluation questions. Evaluators agree that KIF-augmented Transformers are generally more coherent and human-sounding compared to the Generative MemNet.

Compared to existing retrieval models (blue) is more nuanced. Along the lines of existing work (Zhang et al., 2018b; Dinan et al., 2018), we find that retrieval-based models score very well in human evaluations that ask how human or interesting a dialog sounds. This is because retrieval models return human-written utterances from the training set and do not suffer from decoding mistakes present in generative models. For example, on Engaging ImageChat, while our model has significantly improved over the generative baseline (see green bars in Figure 7.2, right), it does not beat retrieval based methods in sounding more human or being more interesting (see blue bars in Figure 7.2, right). As the Retrieval baseline returns human-written text for other humans to evaluate, we hypothesize that humans score each other's writing quite well. Compared to generative models, which we focus on improving, retrieval models often produce longer text with more interesting, nuanced vocabulary usage, and do not make generation mistakes such as repetition. These factors often lead to the stronger performance of retrieval models.

A surprising result is that KIF-augmented Transformers are more human sounding than retrieval models on Wizard of Wikipedia. This is because the dataset's utterances are long and factual due to the tendency of crowdworkers to copy Wikipedia. Sometimes humans chatting with the retrieval bot would respond *uh... that's an interesting fact?* Otherwise, our model scores similarly to retrieval models, with most evaluations not having statistically significant difference.

We conduct a second evaluation on the Unseen Test Set of the Wizard

Knowledge	Training Utterance	Generation
buzz lightyear's name is in honor of astronaut edwin 'buzz' aldrin	my favorite character in that book series is hermione granger	cool! my favorite character in that movie is **buzz lightyear**
mr potato head is based on the **real-life mr. potato head** toy	my favorite character in that book series is hermione granger	my favorite character in that movie is **real-life mr potato head**
slinky dog is a toy dachschund with a metal slinky for a body	my favorite character in that book series is hermione granger	cool! my favorite character is the **slinky dog**
slinky dog is a toy dachschund with a metal slinky for a body	**i really like** the character hermione granger	cool! **i really like** slinky dog
slinky dog is a toy dachschund with a metal slinky for a body	**my favorite character** of all time **has to be** hermione granger	i love that movie, **my favorite character has to be** slinky dog the dachshund
slinky dog is a toy dachschund with a metal slinky for a body	i agree with you! that's **my favorite** character as well	i think so too! **my favorite** is slinky

Table 7.3: **Effect of Fetched Information on Generated Utterances.** The **top** section provides examples for a fixed training utterance, changing the knowledge — the generated text maintains the construction of the training utterance but changes the favorite character to match the knowledge. The **bottom** section provides examples for fixed knowledge but changing the training utterance — the generated text modifies its form to match the training utterance, but the favorite character information remains consistent.

of Wikipedia dataset. Results are shown in Figure 7.3. Trends are similar compared to the results on the Seen Test set, though the preference for the KIF-augmented Transformer is greater over the retrieval baseline. We hypothesize that because the Unseen Test Set is on entirely held out topics, the retrieval baseline can struggle to identify relevant utterances. In contrast, the KIF-augmented Transformer, similar to the generative baseline from Dinan et al. (2018), can use the generative capability to produce utterances.

Lastly, we conduct an additional study to examine the variance of the comparative dialog judgements. The evaluation study for Wizard of Wikipedia is repeated three times on different days, and evaluators who have answered on previous days are not allowed to evaluate again in any subsequent experiments. Overall, we find reasonable interannotator agreement rates, around 73% averaged across all evaluations, which is similar to the agreement rates reported in (Li et al., 2019). We find there is greater variance on questions asking which dialog is *more human* and *more interesting*, most likely as different evaluators can interpret these in different ways. Further, we see that comparison with the Retrieval model has less variance compared to the Generative model, possibly because the Retrieval model's human written text is devoid of mistakes. Overall, we find that the conclusions (and statistical significance) are stable across multiple evaluations.

Analysis of Fetched Knowledge

Example conversations from our KIF-augmented generative model are shown in Figure 7.4 on Wizard of Wikipedia. We find that relevant knowledge is identified that affects the *content* of the generated utterance. For example, the model finds knowledge sentences about Disney movies as the human conversationalist starts the conversation discussing Disney. The model leverages the fetched knowledge to write the content of the generated utterance. In a concrete example, the fetched sentence *disney announced intentions [...] after the success of the incredibles* leads the model to generate the utterance *i love the incredibles, they are my favorite disney movie*.

In contrast, the model uses the form of the fetched training utterance often as a template for writing a response. For example, the model copies the training utterance *Ohhh ... what do people with color blindness do to cope with the effects?* and starts the model generation with *Ohhh ...* and continues with the question *i think toy story is a classic?* following the form of the selected training utterance.

Figure 7.5 displays the top-3 fetched training set utterances and knowledge sentences on the Wizard of Wikipedia dataset when responding to a human utterance. KIF modules can identify multiple relevant items. In response to the human question about *blue skies the 1946 movie* the model identifies both the comedy film and the band.

Finally, the elements retrieved by KIF modules provide a more interpretable understanding of what the model is conditioning upon to generate a dialog response. In Table 7.3, we display for the same dialog history, changing the model's fetched training utterance and knowledge sentence for our own examples. The model heavily incorporates our manual changes of the fetched information into the generated utterance. For example, changing the knowledge directly affects what the model generates as the favorite character — from *buzz lightyear* to *mr potato head* to *slinky dog* — while changing the fetched training utterance changes the form of the generated sentence.

Scaling KIF to Challenging Retrieval Settings

KIF modules can be used in more realistic and challenging settings for knowledge retrieval that test the scalability of the module. In Figure 7.6(a), we compare the Generative Transformer MemNet Baseline with KIF-Augmented Transformers in three settings. The first is the standard Wikipedia sentences provided by the dataset (average 34 sentences). Then, we extend to providing the model with the full Wikipedia article (on average, 57 sentences) and finally to multiple Wikipedia articles (on average, totaling 205 sentences), identified using the conversation's topic. This increasing size of available knowledge could be realistic for settings where it is unclear what information is most relevant, if filtering steps to preprocess the data remove potentially relevant information, or if information synthesis from multiple knowledge sources is necessary to produce a high quality generation. As the Wikipedia knowledge becomes more difficult to identify, performance decreases, but still outperforms the baseline that uses the dataset-provided set of 34 sentences.

Comparing the scaling capability of KIF to the standard Generative Transformer MemNet Baseline highlights the advantage of using KNN. The attention-based mechanism used in Dinan et al. (2018) struggles to identify salient information when given increasingly larger quantities of knowledge, unlike the KNN information fetch. We hypothesize the attention mechanism

Model	Test F1
Wizard of Wikipedia	
Training Utterances Only	18.1
Wiki Knowledge Only	23.9
Training Utterances and Wiki Knowledge	25.9
Engaging ImageChat	
Training Utterances Only	13.9
Related Images Only	13.8
Training Utterances and Related Images	14.4

TABLE 7.4: **Using Multiple KIF Modules on Multiple Sources** is important for improved performance.

Model	Valid F1
Wizard of Wikipedia	
Previous Utterance Only	24.6
+ dialog Context	26.4
+ Turn Embedding	27.4
Engaging ImageChat	
Previous Utterance Only	13.3
+ dialog Context	14.5
+ Turn Embedding + Personality	15.1

TABLE 7.5: **Important Features for KNN Search** using KIF. Salient conversation features improve performance on both datasets.

is challenged by softmax-ing over a larger quantity of inputs, as it can be difficult to make sharp distinctions.

Ablations

- IMPORTANCE OF MULTIPLE KNOWLEDGE SOURCES. One benefit of the KIF module approach is that several modules can be combined, each capturing information from a different source. In both settings, Wizard of Wikipedia and Engaging ImageChat, two modules were used to incorporate multiple forms of knowledge — training utterances to capture the capability of a retrieval-based model and knowledge from Wikipedia or related chats based on image features. We perform here an ablation study to evaluate the impact of using only one source of information. As can be seen in Table 4, performance decreases when only one source of information is used (see Table 7.4).

  For Engaging ImageChat, this study also underlines the importance of being able to fetch in a multi-modal fashion. The general form of the KIF module — requiring only a feature vector to find nearest neighbors from — allows fetching on multiple modalities such as text and images. In Table 7.4, using the Image-based KIF to fetch text from Related Images is important to reach the strongest performance (compare Training Utterances Only that uses text-based KIF and using both Training Utterances and Related Images).

- USING DIALOG FEATURES FOR KNN PERFORMANCE. The quality of the KNN search is critical to the performance of KIF modules. As the external knowledge is kept fixed, KIF must be able to align the dialog context with the

Model	Valid F1
KIF-Augmented Transformer	27.4
One KIF Module fetches multiple times	
2 Fetches	26.9
3 Fetches	26.0
Multiple KIF Modules fetch once each	
2 Fetches	26.5
3 Fetches	25.9

TABLE 7.6: **Multi-hop with KIF** to retrieve information with multiple fetch steps. We find that subsequent fetches are unnecessary.

knowledge to identify relevant pieces of information. In Table 7.5, we show that matching on more features can improve the quality of the retrieved information. Using only the encoding of the immediate previous utterance can improve results on Wizard of Wikipedia by 7 F1 points, but this is further improved by also leveraging the encoding of context (+1.8 F1) and using the dialog turn number (+1 F1). These features are available in the datasets, and we leverage them to improve the relatedness of retrieved knowledge.

► MULTI-HOP RETRIEVAL WITH KIF. Work in memory networks (Weston et al., 2015; Sukhbaatar et al., 2015) employed multi-hop mechanisms. Such capacity could be useful when multiple sources are necessary or information is incrementally fetched. To emulate multi-hop memory mechanisms, we use KIF to retrieve relevant information for $N = 2$ or $N = 3$ fixed hops. As the number of hops is fixed, the multi-hop operation remains differentiable. We do not allow the model to retrieve the same information in a second hop.

We experimented in two settings. First, the same KIF module is used multiple times to fetch different information, and then all of the fetched knowledge is concatenated. Results are shown in Table 7.6 (top). Second, we examine spreading the fetches into *different* KIF modules at various encoder depths. This could be interpreted as the model learning to access more information each layer. As the model progresses deeper, more abstract and high level representations are built, which could allow different knowledge to be retrieved. Results are shown in Table 7.6 (bottom).

In both multi-hop settings, no improvement in performance on the Wizard of Wikipedia dataset is observed. We hypothesize this can be partially attributed to the construction of the dataset — as humans explicitly based their written dialog utterance on one knowledge sentence. Further, it is possible that concatenation brings together too much information for the model to incorporate, and thus adding additional fetches makes the retrieval more noisy.

► EFFECT OF GATING. We analyze the effect of the gating mechanism by evaluating the capability of the gate to identify and focus on salient information. On Wizard of Wikipedia, we concatenate a third source of information: dialog turns from a completely different corpus called PersonaChat (Zhang et al., 2018b). This dataset looks quite different — short utterances without factual knowledge — and should be easy for the model to identify as distinct from Wizard of Wikipedia. As shown in Figure 7.6(b), if KIF on PersonaChat is included without gating, it has a harmful effect as the model includes irrelevant information. When equipped with gating, the model learns to use the gate to ignore some inputs, and can recover almost the full performance of a model without this irrelevant information source.

- Size of K in KNN. Figure 7.6(c) shows the performance on Wizard of Wikipedia when varying the amount of knowledge. Being able to access multiple relevant pieces of information is helpful, but too much information can be harmful. This is likely because the weighted sum becomes blurry if too many sentences are incorporated.

7.3.7 *Conclusion*

We present a KNN-based Information Fetching module that learns to identify relevant information from external knowledge sources by learning a mapping-based read operation. KIF modules benefit from the scalability and efficiency of K Nearest Neighbors search, enabling computation with large external memories. We show in the context of two dialog datasets that relevant knowledge can be identified and incorporated to create more engaging, high quality dialog.

8

Scaling Knowledge Access to the Open Web

"The internet could be a very positive step towards education, organisation and participation in a meaningful society. The Internet is the first thing that humanity has built that humanity doesn't understand, the largest experiment in anarchy that we have ever had."
—Noam Chomsky

▸ SYNOPSIS Curated knowledge sources such as Wikipedia have become ubiquitous solutions for finding information online. However, Wikipedia is only a small fraction of the knowledge available online. To access greater quantities of information, models must be able to retrieve and utilize content from the full internet. We describe a retrieval-augmented architecture for text generation that is capable of writing full Wikipedia articles, with the motivation of enriching Wikipedia with more representative knowledge. We apply this system to the challenge of writing *biographical articles*, particularly for marginalized groups such as women, who have substantially less content on Wikipedia.

8.1 MOTIVATION: KNOWLEDGE FROM THE OPEN WEB

Every day, we have access to more knowledge in our pockets than most of humanity had throughout human history. The ability for us to use our smartphones to do a quick online search to bring up information about almost any topic is unprecedented. To conduct tasks of great complexity, curated information sources such as Wikipedia may not be sufficient. For example, Wikipedia is quite exhaustive, but it does not contain the answer to *every* question someone might ask. Of course, the actual task of generating a Wikipedia article cannot be done by solely reading content already available in Wikipedia. For these applications, a even more vast expanse of knowledge is needed — the knowledge available on the entire internet. In this chapter, we focus on the challenge of Wikipedia article generation as a usecase and testbed for the development of models that can retrieve information from the full web and generate long, structured output in the form of Wikipedia articles.

8.2 WIKIPEDIA ARTICLE WRITING AS A KNOWLEDGE-BASED TEXT GENERATION TASK

Text generation systems have a great number of applications that can help people with every day tasks. One of those is the curation of information on the internet into more easily digestible, organized form. The creation of online encyclopedias such as Wikipedia has eased the challenge of information finding and made knowledge accessible to millions of people globally, all with simple searches.

To write a Wikipedia article, human editors research a topic, cite their references, and then write various sections split by headings and subheadings.[1] Thus, the fundamental task of Wikipedia article generation is that of long form text generation: writing a long article. However, a necessary component is that of identifying relevant information, or evidence to use to write an article. This makes Wikipedia article generation critically dependent on accessing sufficient knowledge.

8.2.1 *Challenges*

Before we discuss our proposed system, we start by describing several challenges of this retrieval plus generation task of Wikipedia article writing.

- Long Input

The internet has become the default source for information online, but the dominance of the modern-day web is tightly coupled with the development of the search engine services that index it. Without the ability to rapidly identify relevant content, the open web would have far too much information for anyone to sort through. Imagine having to skim the spines of every book in the library of Congress each time you wanted to get the answer to a question, for example. This is the same challenge that these open web retrieval models face. To ease this difficulty, most models do not directly query the web,[2] but simulate what a human might do: first use a standard search engine to narrow down possible relevant content based on a query, and second read through the returned search results to identify the useful information for the task.[3]

Despite leveraging search engines first, systems that wish to access the knowledge available on the open web must still face the *long input* challenge — that of digesting the content returned from the search. Search engines return large quantities of information, and often the first couple of search results are not the best. Even if they are, useful (and not redundant) information could still be contained in subsequent search results. Thus, these models must process very long input. This challenges the GPU memory limitations of standard Sequence-to-Sequence systems, which process input sequences of a fixed maximum size.[4] In this chapter, we discuss models with retrieval mechanisms that take long inputs and identify sub-portions that are most relevant for the Wikipedia article writing task.

- Long Output

Beyond digesting large quantities of potentially relevant knowledge, the task of writing an entire Wikipedia article implies writing a large quantity of output text. This is the challenge of *long output*. For decades, researchers have focused on generating more text — from work in paragraph-level summarization, to writing short creative stories, to automatically writing reports on sports games, and so on. Recent work in pretraining such as GPT-3 (Brown et al., 2020) has also shown promising capability at generating paragraph-level texts. However, writing an entire Wikipedia article is still yet

[1] `https://en.wikipedia.org/wiki/Wikipedia:Policies_and_guidelines`

[2] Practically, it is a great challenge to index the web. The use of search engines both mimics the human task but also is important to overcome engineering constraints of search speed and accuracy.

[3] Commonly used search engines in natural language processing research include Apache Lucene or Elasticsearch, which are open source solutions, as well as Microsoft Bing (accessible via a paid API)

[4] This is a particularly common constraint in recent years, as large pretrained models have small maxiumum input sequence lengths

more. High quality Wikipedia articles consist of multiple sections, each of which could be multiple paragraphs, structured with headings, subheadings, and more. This level of organization outstrips the capability of state-of-the-art generation systems today. Even when trained on millions of Wikipedia articles (Liu et al., 2018b), the continuity and readability of automatically generated Wikipedia articles becomes rapidly degenerate.

► FACTUALITY

The utilization of automated systems, such as those for text generation, to populate encyclopedic content online brings a great need for focus on potential challenges such as factuality. So many around the world use Wikipedia as a trusted source for information — the presence of inaccuracy would have huge downsides.[5] Generative systems based on neural networks often have the ability to *hallucinate*, or generate relevant but unsubstantiated text (e.g. the information is not contained in the input). Previous work strongly indicates this comes from the language modeling component learned by Sequence-to-Sequence systems[6]. Fundamentally, these models learn local associations between words to predict a probability distribution over the next word that should be written. Thus, words that often appear together — even if factually inaccurate in context — are often written. In this chapter, we focus on analysis of factuality in generated text, as Wikipedia articles should represent factual sources of knowledge online. However, factuality concerns broadly plague text generation across most tasks, and evaluation of generated content is a largely unsolved problem.[7]

8.3 GENERATING BIOGRAPHIES FOR MARGINALIZED GROUPS ON WIKIPEDIA

8.3.1 *Introduction*

"The cool thing about the Internet is that it's allowing women more access to their own history."
—Kathleen Hanna

Online encyclopedias such as Wikipedia are ubiquitously used as one of the default sources of knowledge on the internet. This ease of access to information and its availability in a convenient, centralized location makes Wikipedia one of the major sources of dissemination of knowledge across the globe. However, the knowledge contained in Wikipedia is not neutral — it is biased in various ways. Many studies, including those from the Wikimedia Foundation itself, have emphasized that biographies in particular are overwhelmingly written about men. The impact of this bias is profound — it presents a slice of the world through the accomplishments of one subset of gender, which marginalizes and diminishes the accomplishments of other genders. This leads to many subtle yet far-reaching effects, from students not writing their first book reports on a woman to bias in models trained on Wikipedia, as Wikipedia has long been used as a source of data.

We study how to generate biographies for marginalized communities, focusing on women. We confront several major challenges. First, this is fundamentally a long-form generation task. Improvements driven by pretraining have improved generation fluency at the level of multiple sentences. However, Wikipedia biographies contain multiple paragraphs in a structured form with headings, as well as citations. Second, the task confronts obstacles around the factuality of generated content, as articles must be factually

---

[5] This is a known problem on Wikipedia, so much so that they catalogue known inaccuracies and controversial content. For example: `https://en.wikipedia.org/wiki/Wikipedia:List_of_hoaxes_on_Wikipedia` and `https://en.wikipedia.org/wiki/List_of_Wikipedia_controversies`

[6] We refer the reader to a work with an excellent title, *Why are sequence-to-sequence models so dull?* from Jiang and de Rijke (2018).

[7] Most studies leverage human evaluation, which has been found effective in more constrained tasks such as machine translation. In translation, the generated text should be a direct translation of the input text, and thus the factuality can be evaluated with a short, clear reference. However, challenges of evaluating factuality in Wikipedia article generation are further compounded by the long input and long output obstacles previously discussed.

accurate. Third, Wikipedia articles are written using reference material, often found on the web. Thus, models need to ingest web searches as a pre-requisite to writing accurate, long-form biographies.

We develop a method that starts with the subject and occupation of the biography, then leverages web search to find relevant evidence. Given web search results, we employ a retrieval-augmented generation architecture based on large-scale pretraining to identify relevant information and generate the biography. We generate section by section, using a caching mechanism similar to Transformer-XL to reference previously written sections and achieve greater document-level context. Finally, after each generated sentence, we append a citation based on which web searches were retrieved.

To evaluate the quality of our generated text, we create a dataset of 1,527 Wikipedia biographies about women, including three intersectional categories: women in science, women in Asia, and women in Africa. We add web-based evidence to create a full evaluation dataset. We utilize this dataset to quantify the quality of generation using several automatic metrics such as ROUGE-L, entailment, and named entity coverage. Finally, we conduct a large-scale human evaluation to measure the factuality and coverage of our generated biographies. We hope that our techniques can eventually be used as a starting point for human Wikipedia writers, for biographies and beyond.

8.3.2 *Related Work*

Bias in Wikipedia Biographies

Gender bias on Wikipedia is a well-known problem (Hinnosaar, 2019; Dinan et al., 2020; Schmahl et al., 2020), particularly in the case of biographies (Graells-Garrido et al., 2015; Stratigakos, 2016; Luo et al., 2018; Schmahl et al., 2020). This bias is compounded by geographical location, as information about certain areas of the world is far more prevalent (Kaffee et al., 2018a; Beytía, 2020). This bias exists not only in what articles are written, but also in articles targeted for deletion — articles about certain marginalized groups are removed at higher rates (Worku et al., 2020). Wikipedia reflects biases present in society (De-Arteaga et al., 2019; Young et al., 2020; Schmahl et al., 2020), though numerous initiatives exist to de-bias Wikipedia. These range from training programs (Iglesias, 2020) to projects such as Women in Red[8] and WikiProject Women.[9,10] The success of these initiatives has been studied (Langrock and González-Bailón, 2020) and found to be effective, but not at addressing the systemic challenges that create bias in the first place.

In the natural language processing community, work has focused on combating gender bias in co-reference resolution (Zhao et al., 2018a), dialogue (Dinan et al., 2019; Lee et al., 2019; Liu et al., 2020a), detection of abusive language (Park et al., 2018), machine translation (Stanovsky et al., 2019), and word embeddings (Gonen and Goldberg, 2019). These works present a variety of strategies, including data augmentation, additional data collection efforts, modified generation, and fair evaluation (Yeo and Chen, 2020). A comprehensive survey can be found in Blodgett et al. (2020). However, most of these efforts are focused on specific tasks or models — our

[8] `https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Women_in_Red`

[9] `https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Women`

[10] Other interesting initiatives include `https://www.newscientist.com/article/mg24532680-800-jess-wades-one-woman-mission-to-diversify-wikipedias-science-stories/` and `https://www.nature.com/articles/d41586-018-05947-8`.

work uniquely targets generation of full Wikipedia biographies to combat gender bias present on Wikipedia.

Generation of Wikipedia Articles

A large body of work in generation utilizes Wikipedia, often for data-to-text tasks that use Wikidata and RDF triples (Perez-Beltrachini and Gardent, 2017; Kaffee et al., 2018b; Chen et al., 2020b; Wang et al., 2020; Agarwal et al., 2020; Parikh et al., 2020), as well as graphs (Jin et al., 2020). Some have focused on long text, such as writing summaries (Chen et al., 2020a) or sections of articles (Kaffee et al., 2020), expanding stubs (Banerjee and Mitra, 2015), and writing full articles (Liu et al., 2018b). Some of these works utilize structure to learn templates (Sauper and Barzilay, 2009), Markov logic networks (Liu et al., 2010), or word graphs (Banerjee and Mitra, 2015), but we anticipate that pretraining and large neural network based techniques will vastly improve upon this quality. Closest to our work, Liu et al. (2018b) use web evidence to write full length articles, but do not focus on biographies or bias.

Work has focused on automatic creation of biographies, such as generation from infoboxes (Lebret et al., 2016) or Wikidata (Chisholm et al., 2017), as well as extracting biographical sentences (Biadsy et al., 2008). The majority of this work focused on short biographies.

Retrieval in Generative Models

Retrieval mechanisms have been used across a variety of tasks, including dialogue (Moghe et al., 2018; Dinan et al., 2018; Shuster et al., 2021), fact verification (Thorne et al., 2018), and sentence generation (Guu et al., 2018). Most notably, retrieval has been heavily used in question answering (Chen et al., 2017; Kwiatkowski et al., 2019; Seo et al., 2019; Karpukhin et al., 2020). Recent innovations in incorporating retrieval mechanisms have increased the quality and scale of retrieval-augmented generative methods (Guu et al., 2020; Lewis et al., 2020b; Izacard and Grave, 2020).

8.3.3 *Method*

Wikipedia biographies begin with an introductory paragraph followed by various subsections.[11] They usually start with the subject's early life, career, and then later accomplishments. Certain groups have templates, e.g. directors have a filmography, athletes have a record of major competitions, and scientists a list of well-known publications.

To account for this structure and generate long-form text based on retrieved web evidence, our system, illustrated in Figure 8.1, generates a biography section by section as follows. Based on the subject, their occupation, and the section heading, the model first identifies relevant evidence from a set of web search results (*retrieval module*). It then conditions upon that evidence to generate the section, using a Transformer Sequence-to-Sequence model (*generation module*) which can access previous sections using a caching mechanism. Finally, the model indicates which evidence

[11] Many biographies contain infoboxes, which we do not generate.

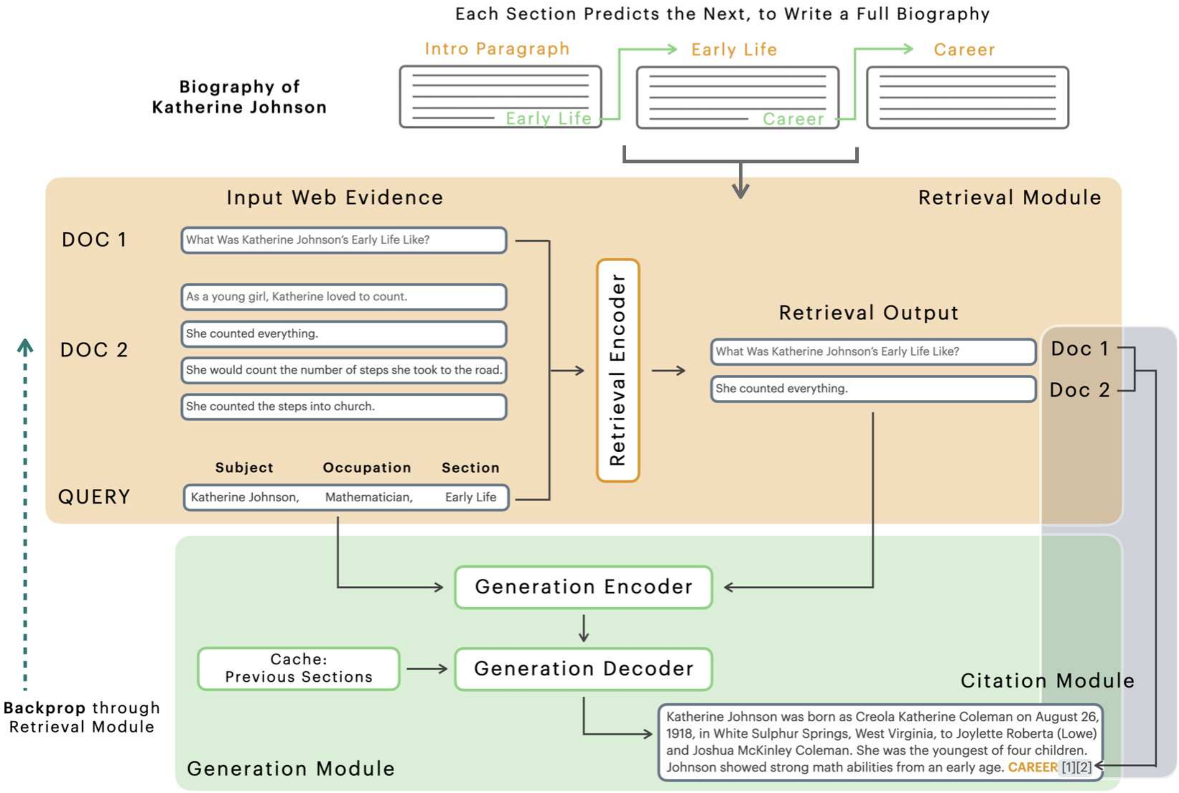

Each Section Predicts the Next, to Write a Full Biography
Biography of Katherine Johnson
Intro Paragraph
Early Life
Career
Early Life
Career
Input Web Evidence
Retrieval Module
DOC 1
What Was Katherine Johnson's Early Life Like?
DOC 2
As a young girl, Katherine loved to count.
She counted everything.
She would count the number of steps she took to the road.
She counted the steps into church.
QUERY
Subject
Occupation
Section
Katherine Johnson,
Mathematician,
Early Life
Retrieval Encoder
Retrieval Output
What Was Katherine Johnson's Early Life Like?
She counted everything.
Doc 1
Doc 2
Generation Encoder
Cache: Previous Sections
Generation Decoder
Citation Module
Backprop through Retrieval Module
Generation Module
Katherine Johnson was born as Creola Katherine Coleman on August 26, 1918, in White Sulphur Springs, West Virginia, to Joylette Roberta (Lowe) and Joshua McKinley Coleman. She was the youngest of four children. Johnson showed strong math abilities from an early age. CAREER [1][2]

FIGURE 8.1: **Model Architecture.** Our method writes a Wikipedia article section by section, with each section predicting the next in sequence. To write one section, the model starts with a *retrieval module* that uses a query consisting of the subject name, occupation, and section heading to identify the most relevant information from the web. The query and retrieval output passes to the *generation module*, which generates the desired section while using a cache to reference previously written sections. Finally, to complete the full Wikipedia article, the *citation module* appends citations based on the retrieved content. The entire system is learned end-to-end, with backpropagation from the generation module through the retrieval module.

documents it used and outputs those as citations, mimicking a standard Wikipedia article (*citation module*).

Retrieval Module

Given a query Q and a set of web documents D retrieved from the web based on this query, the task of the retrieval module is to retrieve the subset of D that is most relevant given Q. The challenge is sifting through the large quantity of potentially useful information.

- QUERY. The query Q consists of three parts: **(1)** the name of the person for which the biography is generated, **(2)** their occupation, and **(3)** the section heading. Including the occupation narrows the realm of potential relevant content, especially as proper names are often ambiguous (e.g. *Jane Wang*). Similarly, the section header allows the model to retrieve different information for each section (e.g. *Personal Life* compared to *Career*).

- DOCUMENTS. The query Q is sent through a search engine to retrieve a number of web hits which form the set of documents D that are candidates for retrieval. The web results are represented only as text, and all non-text information is discarded.

- RETRIEVAL. To retrieve the relevant subset of D, each sentence in D is encoded with RoBERTa base trained with LayerDrop (Fan et al., 2020a; Liu et al., 2019b; Devlin et al., 2019). The concatenation of the subject's name, occupation, and section heading is also encoded. We then conduct maximal inner product search (MIPS) to identify which document sentences are most relevant given the current query Q, following the strategy used in retrieval-augmented generation (RAG) (Lewis et al., 2020b). In RAG, the set of documents D is pre-encoded with BERT and MIPS is used to align the document with the query. Our strategy is similar, except that the RoBERTa encoder is not fixed, so the retrieval module learns based on the performance of the generation module. We cap the retrieval quantity at 1000 words, which represents about 40 sentences that could be used to generate each section.

Generation Module

To generate the sections given the input query and retrieved evidence, we use a Transformer-based Sequence-to-Sequence model initialized with BART-Large (Lewis et al., 2020a). BART is trained using a generative denoising pretraining mechanism. The input to BART is the concatenation of the subject's name, occupation, the section heading, and the retrieved evidence. Note that the maximum number of input tokens for BART is 1024 words, which is why we cap the retrieval at 1000 words, as described in the previous section. The decoder conditions on the input information to generate the output biography section.

One challenge with this is that the sections would be generated completely independently, which does not reflect the structure of a Wikipedia article and might result in redundancy between generated sections. Thus, we equip the Sequence-to-Sequence model with a mechanism to refer to previous sections using the cache mechanism from Transformer-XL (Dai et al., 2019). This mechanism caches the previous section's hidden states at every layer, using it as memory to generate the current section.

Citation Module

Recent work has focused on models that not only perform a task, but also produce an explanation (DeYoung et al., 2019). Much of this work has focused on question answering (Latcinnik and Berant, 2020; Lamm et al., 2020; Lakhotia et al., 2020; Gonzalez et al., 2020) and generating explanations in natural language (Camburu et al., 2019; Narang et al., 2020; Kumar and Talukdar, 2020; Hase et al., 2020). A similar requirement exists on Wikipedia — not only to collate the information into an article, but to provide the *original references* for users to verify. Thus, to complete the generation of a full Wikipedia biography, we cite the information used, as in any real article. On Wikipedia itself, each sentence could contain citations. We simplify this, citing at the end of each section. To do this, we track the original document the retrieved evidence originates from, and reference that document at the end of the generated section.

Bringing it All Together

To write a full biography, models must generate the introductory paragraph followed by each section. For a new article, the introductory paragraph is

TABLE 8.1: **Example Section Headings and Occupations in our Evaluation Dataset.** Across biographies, there are a large number of different section headings, but many are standard. For occupations, the most popular occupations have significantly more biographies than others.

Most Common Section Headings
Career, Personal Life, Early Life, Biography, History
Most Common Occupations
Writer, Politician, University Teacher, Physician, Researcher

given as a section heading called *toplevel*. For each subsequent section, we follow the process outlined above to retrieve evidence, then write a section, then add citations. At the end of each section, the model predicts the *section heading* of the next section. This allows the model to generate an entire article section by section.

8.3.4 *Creating an Evaluation Dataset*

We create a novel evaluation dataset which consists exclusively of women biographies. We collate candidate biographies, then retrieve information about occupation, and finally gather web sources using web search. The resulting dataset, summarized in Table 8.2, consists of 1527 biographies, each linked to a set of retrieved web articles.

► IDENTIFYING BIOGRAPHICAL SUBJECTS.

We first source various notable women on Wikipedia using internet lists (e.g. *Famous Women you should know*) and existing efforts by collective groups of Wikipedia editors, such as the Women in Red project. Several recent efforts focus on Women in Science,[12] and so we specifically include scientists as a category. Overall, we collate almost two thousand candidate Wikipedia women biographies. We then narrow down by selecting articles that have previously *Featured Article* or *Good* quality.

However, this strategy identifies women who are predominantly white or from western countries. We next focus on identifying biographies of women in Africa and Asia, two areas where representation is significantly lacking. We leverage the Women in Red effort, which maintains an ongoing list of recently written or improved articles about women. A number of these articles are listed under e.g. Women in Asia, but are about a European diplomat or missionary that had spent a few years in Asia. We manually remove these. The final evaluation dataset contains 1,527 biographies in four groups: Women, Women in Science, Women in Asia, and Women in Africa (see Table 8.2).

► BIOGRAPHY TEXT AND OCCUPATION.

After finalizing candidate Wikipedia biographies, we use the MediaWiki API[13] to query the text of the article. To identify the occupation, we use the Wikidata API[14] to query the information. Individuals often have more than one occupation (e.g. *Rachel Carson* is an author and an environmental activist). As seen in Table 8.2, on average, articles have around 6 sections with 130 words each. The most common occupations include writers, teachers, and doctors (see Table 8.1), though the entire dataset contains almost 500 different occupations, with people having on average 2 occupations (see Table 8.2).

[12] `https://towardsdatascience.com/who-is-wikipedia-famous-within-natural-language-processing-fa0c8e91bdf6?gi=b910dd838c47`, `https://www.newscientist.com/article/mg24532680-800-jess-wades-one-woman-mission-to-diversify-wikipedias-science-stories/`

[13] `https://www.mediawiki.org/wiki/API`

[14] `https://query.wikidata.org/`

Biographies about Women	419
Biographies about Women in Science	808
Biographies about Women in Asia	164
Biographies about Women in Africa	136
Total Biographies	1,527
Average Number of Sections	5.8
Average Length of a Section	132.3
Average Length of Total Article	765.9
Avg Number of Occupations	1.91
Total Number of Unique Occupations	474
Avg Number of Web Hits (max 20)	18.1
Avg unigram overlap of Web Hits and Biography	24.9%

TABLE 8.2: **Breakdown and Statistics of Biographies in our Evaluation Dataset.** We display a variety of information about our created dataset, analyzing the content and length of the articles.

- RETRIEVING WEB EVIDENCE.

Next, we identify web sources with reference evidence for each biography. We follow the construction of similar datasets, such as WikiSum (Liu et al., 2018b) and ELI5 (Fan et al., 2019b), which search through CommonCrawl. We query CommonCrawl based on the subject's name and occupation and return the top 20 search results. Statistics are presented in Table 8.2. Out of a maximum of 20 possible hits, on average each biography returns around 18.

8.3.5 *Experimental Details*

We describe our training data, baselines, and automatic and human evaluation metrics.

- TRAINING DATA.

We utilize the WikiSum (Liu et al., 2018b) dataset of Wikipedia articles paired with web references. We filter to biographies using a combination of querying for occupations in Wikidata and using Named Entity Recognition[15] to recognize names. We query each article title in the WikiSum dataset to attempt to find an occupation and see the title is recognized as a named entity, to identify the bibliographical subset of WikiSum. This produces 677,085 biographies, each associated with a set of web articles.

- BASELINE.

We compare our method described in Section 8.3.3 to a pretraining and finetuning generation baseline. We use the BART model (Lewis et al., 2020a) and finetune on the Biography subset of the WikiSum data. To compare this baseline with our method equitably, the baseline is also trained to generate section by section. However, it does not use the retrieval module, the caching mechanism, or the citation module (as described in Section 8.3.3). Additional training details are in the Appendix.

- GENERATION.

We generate from all models with beam search, setting the beam size to 5. We allow the model to generate an output of any length, with no restrictions. For human evaluations, we set the minimum and maximum

[15] `https://spacy.io/usage/linguistic-features/`

length such that it matches the length of the gold target to minimize the effect of length on human interpretations.

► AUTOMATIC EVALUATION.

We evaluate the quality of generated biographies with three automatic metrics. First, we measure the **ROUGE-L** between the generated text and the Wikipedia reference text to assess the similarity. ROUGE-L is commonly used in multi-sentence summarization and is a measure of longest common substring overlap.

Next, we use **Natural Language Entailment** as a high level proxy for quantifying a form of factuality: if two sentences entail each other in both directions, then they are semantically equivalent. We use a model pretrained and finetuned on MNLI, open sourced by Liu et al. (2019b). To evaluate entailment, we split the generated biography and reference biography into sentences, then for each sentence in the generated biography we calculate if it is semantically equivalent to a sentence in the reference. We then compute the percentage of generated sentences that are semantically equivalent to at least one sentence in the reference biography, where entailment is evaluated bidirectionally.

Finally, we assess the **Coverage** of information in the generated biography, constraining this to analyzing mentions of named entities. We report the percentage of named entities detected in the reference which are also detected in the generated text. We extract entities with BLINK, a BERT-based entity linking system (Wu et al., 2019).

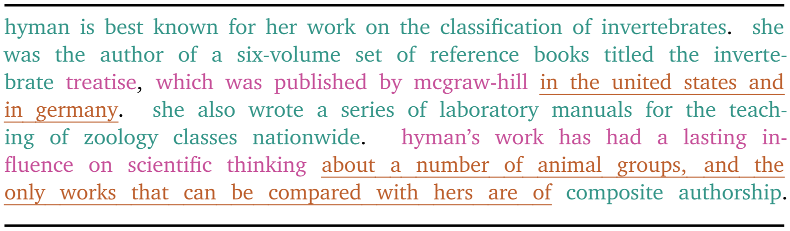
► HUMAN EVALUATION

Long-form text generation is very difficult to assess automatically (Thomson and Reiter, 2020; Howcroft et al., 2020), particularly for factuality (Goodrich et al., 2019; Maynez et al., 2020; Peshterliev et al., 2021) and hallucination (Zhou et al., 2020; Dušek and Kasner, 2020). We conduct a detailed, large-scale human evaluation with the goal to assess **Coverage** (How much of the information in the reference section is in the generated section?) and **Factuality** (How much of the generated section is in the reference and, for the information added in the generated text, how much of that information is verifiable based on the web evidence?).

To reduce the challenge of evaluation, the text is compared section by section, and the generated text is the same length as the reference by constraining the max length of beam search (to remove length as an evaluation artifact). First, each sentence of the generated section is shown next to the full reference section and the entire document cited in the generated section (recall our generated biographies cite the retrieved evidence). Evaluators are asked to decide **(1)** if the information in the generated sentence is present in the reference section (ground truth) and **(2)** if the information in the generated sentence is present in the cited document (web evidence). This question assesses if the information from the generated section is factual with respect to either the reference Wikipedia text or the retrieved web documents. Then, the evaluation is flipped to assess coverage with respect to the Wikipedia reference. Each sentence of the reference is shown next to the generated section, and evaluators are asked to decide **(3)** if the information

Model	ROUGE-L	Entail	Named Entity Coverage
BART Pretraining + Finetuning	17.4	15.8	21.9
+ Retrieval Module	18.8	17.2	23.1
+ Caching Mechanism	19.3	17.9	23.4

TABLE 8.3: **Full Results on Biography Generation on our constructed evaluation dataset.** We compare the BART baseline with our method across different automatic metrics to assess fluency, factuality, and coverage.



hyman is best known for her work on the classification of invertebrates. she was the author of a six-volume set of reference books titled the invertebrate treatise, which was published by mcgraw-hill in the united states and in germany. she also wrote a series of laboratory manuals for the teaching of zoology classes nationwide. hyman's work has had a lasting influence on scientific thinking about a number of animal groups, and the only works that can be compared with hers are of composite authorship.

FIGURE 8.2: **Example Generation** of the *Work* section for a biography about Libbie Hyman, a zoologist. Green indicates text in the reference article, Pink indicates text in the web evidence, and Orange (underlined) indicates hallucination. See the biography on Wikipedia: `https://en.wikipedia.org/wiki/Libbie_Hyman`.

Model	ROUGE-L
Retrieval with Different Queries	
with Subject Name Only	19.6
with Name and Occupation	19.8
with Name, Occupation, Section Heading	21.4
Writing Articles in Sections	
Entire Article	14.4
Section by Section	15.9

TABLE 8.4: **Ablations** of types of Queries for the Retrieval Module and generation section by section.

in the reference sentence is present in the generated section. In total, human annotators evaluated 100 sections with length between 200 to 500 words. Additional details are in the Appendix.

8.3.6 *Results*

We describe our main results and analyze the bias present in generated biographies. An example generation is shown in Figure 8.2.

Quality of Generated Biographies

- AUTOMATIC EVALUATION. We examine the model's overall performance on our constructed evaluation dataset about different groups of women. Results are summarized in Table 8.3. Compared to the pretraining+finetuning baseline, adding the retrieval module statistically significantly[16] increases results by 1.4 ROUGE-L. Adding a caching mechanism improves further by 0.5 ROUGE-L. This trend is reflected across the entailment and entity coverage metrics, indicating that retrieving the most relevant information to write a biography is critical.

Next, we examine the impact of our modeling choices. Compared to previous work on WikiSum (Liu et al., 2018b; Fan et al., 2019a), we add an end-to-end retrieval mechanism based on RAG that substantially improves results. Further, instead of retrieving solely based on the subject name, as was previously done (Liu et al., 2018b), we retrieve on a detailed query (the name, occupation, and section heading). Table 8.4 indicates that this enriched query improves the retrieval quality by almost 2 ROUGE-L. We

[16] We use the confidence interval reported in the ROUGE package.

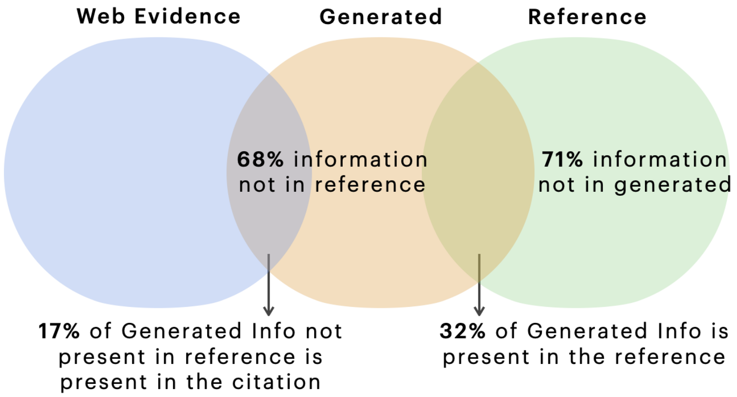

Web Evidence
Generated
Reference
68% information not in reference
71% information not in generated
17% of Generated Info not present in reference is present in the citation
32% of Generated Info is present in the reference

FIGURE 8.3: **Human Evaluation.** We compare the coverage of content between generated and reference biographies, as well as the factuality of generated content.

TABLE 8.5: **ROUGE-L Performance broken down by sub-categories.** We compare the BART baseline with our method across different subsets of women, as well as the biography subset of WikiSum Test.

Model	**WikiSum Text**	**Women**	**Scientists**	**Women Asia**	**Women Africa**
BART Pretraining	19.0	17.4	18.2	16.7	16.4
+ Retrieval	21.4	18.8	19.3	17.9	17.1
+ Caching	21.8	19.3	19.7	18.4	17.3

conjecture it helps improve disambiguation and retrieve evidence that is relevant to the desired entity rather than to one of its homonyms.

We also generate the biographical articles section by section, rather than an entire article at once. This allows the retrieval mechanism to be focused on the section information. As shown in Table 8.4, this also has a positive effect of +1.5 ROUGE-L.

► HUMAN EVALUATION. Next, we examine quality with human evaluation, as shown in Figure 8.3. Our goal is to understand how much information in the generated text is present in the reference text or the web evidence, as a proxy for factuality and coverage. Overall, 68% of the information in generated sections is not present in the reference text. In reverse, 71% of information in the reference text is not in the generated text. These comparisons indicate that the generated text is far from perfect coverage. However, how much of the 68% of information added in the generated text is factually accurate? We found that 17% of the added information can be validated by examining the web evidence, which shows that some information added by the generative model is valid biographical information.

TABLE 8.6: **Statistics of Biographies on Wikipedia.** Biographical content is overwhelming about men

% of Articles about Men	78.4[17]
Avg Length of Articles about Men	860.5
Avg Length of Articles about Women	682.4
% Male-Gendered words in Biographies	72%

[17] The Women in Red project estimates that 81.14% are about women, similar to our calculated statistic.

TABLE 8.7: **Improved Performance when Finetuning** on biographical articles about women. We finetune on biographies about women that do not include this subset of women in Asia and Africa.

Model	**WikiSum Test**	**Women Asia**	**Women Africa**
Our Method	19.0	16.7	16.4
+ finetune on Women	18.9	17.3	16.8

We examine why there is low information overlap between the generated and reference text. First, information in the reference biography may not be available on the web[18] or may not be retrieved. In a manually examined subset of 250 sentences taken from reference biographies, we found that about 50% of the information was not contained in the web evidence. The other 50% was partially present in the web evidence but were not retrieved by the generative model. Second, annotators must compare sentences, but sentences contain partial information. For example, if *Person is was born in Chicago in 1968* was in the generated text and *Person was born in Chicago* was in the reference text, this would count as the generation having information not in the reference.

In summary, our investigation suggests two explanations for the low coverage reported by human annotators: lack of information in the web evidence and difficulty assessing whether two sentences contain the same core knowledge.

Gender Bias in Generated Biographies

We analyze gender bias in generated biographies, breaking down two major sources of bias: bias in data and bias in models.

- DATA BIAS.

We first discuss bias from the relative prevalence and quality of biographies about certain groups compared to others. We analyze our training dataset and summarize statistics in Table 8.6. Almost 80% of biographies are written about men, articles about men are on average 20% longer, and the distribution of occupations tends more towards politicians and other historical leadership figures (e.g. kings). Biographies contain many male-gendered words (based on gendered wordlists collected in previous work such as Zhao et al. (2018b)). Further, words used in biographies about men and women differ, meaning models trained on this data could learn a different distribution over the next likely word. We find that gendered words are included in biographies about women unnecessarily (Schmahl et al., 2020), such as writing *female scientist* rather than *scientist*.

Biography generation requires a solid body of information as a prerequisite, meaning the web is a critical dependency. Data bias also manifests in the difference in quality of the retrieved web evidence — biographical articles about men have a larger number of web articles and these articles are longer and more relevant.

- MODEL BIAS.

Bias also exists in models. If a model generates worse quality biographies about certain groups compared to others, we refer to this phenomenon as model bias (Buolamwini and Gebru, 2018). We analyze model bias by comparing the results on our evaluation set about women with those on the WikiSum test set (consisting of 70% biographies of men). As shown in Table 8.5, across the board, the quality of generated biographies is higher for the WikiSum Test set. This is especially prominent for Women in Asia and Africa, which are more than 2.5 ROUGE-L worse on average.

---

[18] Note that search hits from the Wikipedia domain are removed from web search results.

We investigate the potential of a straightforward strategy to mitigate differences in training data: that of training on biographical articles about women. We mimic this by finetuning our model on a subset of our evaluation dataset, and then testing on Women in Asia and Africa, the two categories that perform most poorly. As shown in Table 8.7, finetuning statistically significantly improves performance, though the improvement is not large (+0.5 ROUGE-L). Mitigating bias is a fundamental challenge that is not easily resolved by finetuning alone. However, techniques like data augmentation and finetuning have proven effective (Dinan et al., 2020; Zmigrod et al., 2019), and may work towards reducing the gap in performance.

8.3.7 *Discussion*

In this section, we discuss several known limitations and ethical considerations of our work. We do not recommend any kind of text generation technology to be deployed on Wikipedia given this is an active area of research.

Dependency on Evidence from the Web reflects Bias on the Internet

Biographies, whether written as books or available online, reflect societal bias. While many Wikipedia editors rely on web-based references to create their articles, and we follow the same strategy in this work, relying on the web is flawed. The prominent reason is that the internet is full of bias in it of itself. For example, Donna Strickland, who received a Nobel Prize, did not have a Wikipedia article[19] as there was not sufficient content about her on the web as a basis for her article. Thus, it is important to recognize that the availability of references is problematic, affecting the downstream ability to write accurate, comprehensive biographies. Further, information on the web can be contradictory, information can be affected by the passage of time, and not information on the web is necessarily factually correct. Our proposed modeling mechanism does not have a way to explicitly recognize or correct for these challenges, which also plagues text generation generally.

Focus on English Limits Inclusivity from Other Languages

Our work focuses on text generation in English only, which limits inclusivity purely on the basis of language. This is challenging as the content of the internet and Wikipedia itself is different in various languages. For example, articles about people from Germany may be more likely to be located on the German version of Wikipedia. Another factor is that the content of the references may be written in another language, and then used by a bilingual individual to write an article in English about that subject. This is often the case for many biographical subjects who may be more well known in a non-English speaking area.

Evaluation focuses on Women Only, Not Other Groups

There are a very large number of marginalized groups in the world and numerous important intersectional aspects to consider. When discussing identity, a wide variety of factors and personal views influence individuals

---

[19] `https://wikimediafoundation.org/news/2018/10/04/donna-strickland-wikipedia/#:~:text=Donna%20Strickland%20is%20an%20optical,of%20a%20Sloan%20Research%20Fellowship.`

when thinking about how they describe themselves. Our evaluation dataset focuses on women alone, which leaves out many groups, including non-binary people. Further, Wikipedia may not reflect the up-to-date information — names and gender are both mutable, for example — and Wikipedia articles do not ask each subject to self-report their gender. Finally, we note that by grouping people into hard categories, there can potentially be harm — such as limiting people from opportunities because of their gender or race. However, we strongly believe that it is important to recognize bias in its various forms as it exists, particularly in popular, default online sources of information such as Wikipedia.

Bias in Style, Word Choice, and Tone

In this work, we focus on bias manifesting as unequal prevalence and length of biographical content on Wikipedia, focusing specifically on different intersectional groups of women. However, bias manifests in a number of other ways. Studies have indicated that the words used in biographies about women compared to biographies about men (Dinan et al., 2019) also differs, and is reflective of gendered terminology. For example, many articles about women are actually written with a lot of information about men, such as their husband's careers, and articles about actresses describe more often their physical appearance. This is also a manifestation of bias, and we do not present any focused modeling techniques to address this type of bias explicitly.

Biographies as Records

In the modern internet, a large number of events are recorded for the public record. These include events that people may personally prefer to forget, often termed *right to be forgotten*[20]. Automatically generating biographies about individuals may collate such information in an easily accessible public place, which can conflict with this personal right. This has a complex but important interaction with marginalized groups. For example, many celebrities who are women, transgender, or a part of another marginalized group are far more likely to have news articles written about intimate personal details such as plastic surgeries. Thus, it is important to consider the interaction of biographical data with individual privacy. This is a larger challenge of biographical information generally.

8.3.8 *Conclusion*

We developed a novel retrieval- and cache-augmented generation model to generate long-form, section-structured, reference-linked biographies. To evaluate our model, we create a large scale evaluation dataset of women biographies and related web search results. Experimental evidence reveals that an enriched query, caching, and backpropagation through a retrieval-augmented encoder-decoder model each contributes to improved performance. We analyze gender bias in the training data as well as bias exhibited in trained models and show that these biases negatively impact the quality of generated women biographies. We hope that this work can be used as a

---

[20] `https://en.wikipedia.org/wiki/Right_to_be_forgotten`

starting point for combating these biases and help fill in the large gaps in Wikipedia biographical content.

9

Knowledge on the Web, in Structured Form

> *"The Internet changes the structure of society all the time – this massiveness made of individuals."*
> —Ai Weiwei

▸ SYNOPSIS If free form natural language was the most efficient way of comprehending information, why would tables exist? We discuss the motivation behind treating large quantities of information not only as text, but creating a structured graphical representation for this knowledge. Structured information has strong benefits of ease of organization and identification of redundancy, but the structured form may not naturally exist. We describe a system for automatically converting large quantities of text from the open web into a knowledge graph, and then using the structured information as input for text generation tasks including long-form question answering and Wikipedia article generation.

9.1 MOTIVATION: KNOWLEDGE IN STRUCTURED FORM

Information comes in many forms. When we want to learn about a new topic, we may turn to an authoritative textbook. When we check tomorrow's weather, we look at a table of dates and temperatures. When we examine student grades, we look at databases with rows and columns. We broadly refer to the organization of knowledge in tables, databases, and knowledge graphs as *structured forms* of knowledge. Unlike natural language, where information could be presented in a wide array of different possible ways, structured forms of knowledge usually have a predefined notion of information organization. For example, consider the student gradebook we described. Each student would occupy a row, with each column being the grade of that student on a specific assignment. New assignments and new students would follow that predefined schema. Contrast that with text such as *Student A did assignment 1 and received an A. On assignment 1, student A received an A. Student A received an A on assignment 1.* — three different ways of communicating the same information.

Structured information in specific domains — such as student grades — is very easily digestible, a strong benefit of such predefined schemas.[1] This can benefit various different tasks that require looking up information. For example, a database of student names and grades is very easily queried programatically, even with simple formulas in tools such as Microsoft Excel, without need for natural language understanding models or even Python.

[1] As a noted downside, the design of these schemas need to be flexible enough to incorporate all information and anticipate future usecases, making schema design often require domain experts.

However, a large drawback of structured information is that they must be constructed to be useful. Knowledge exists naturally in many forms, but the organization of information into clean Excel spreadsheets is an unrealistic expectation. For years, organizations have tried to collect and collate information into various databases. An illustrative example is the Google Knowledge Graph effort, or open source repositories of information on Wikipedia such as Wikidata or DBPedia. Despite these efforts, the organization of all information on the internet is an endless and ongoing task. It is made more difficult by the desire for consolidation, or fitting existing knowledge into one large, clean structure.

In this chapter, we consider how to automatically convert free text into a simple, structured knowledge graph, and then utilize this knowledge graph for text generation tasks such as writing long, explanatory answers to questions and writing Wikipedia articles. Similar to the previous chapter, these models directly digest content from search engine results from the open web, tackling a similar long input challenge — but instead of retrieving from natural language, opt to construct a structured knowledge graph. Unlike existing knowledge graph efforts, our goal is not to create one true knowledge graph, but to organize textual information into a more structured form to do a specific, targeted task — such as answering *one question* correctly or writing a single Wikipedia article. This eases the challenge of knowledge graph creation, as the knowledge graph does not need to be endlessly reused. We describe our method for constructing these structured knowledge representations and how to incorporate them into standard Sequence-to-Sequence architectures.

9.2 LONG-FORM QUESTION ANSWERING AS A KNOWLEDGE-BASED TEXT GENERATION TASK

Finding information online is often driven by curiosity — from having a discussion with a sibling and wanting to prove you are right to learning more about the world we live in. Curiosity is exemplified in the task of question answering, or asking questions and using automated systems to answer them. Nowadays, question answering systems have become fairly prevalent. An iPhone, for example, is equipped with a question answering system that can query to report back on tomorrow's projected weather. These usecases are often driven by lookup-based question answering systems, which must understand the user question, but then utilize a specific API (such as a weather API) to answer the question.

In the recent years, a large amount of progress has been made on question answering datasets such as Stanford Question Answering Dataset (SQuAD) and Natural Questions (NQ). These are *extractive* question answering datasets, meaning the task is to understand the question and select a short span from the provided input paragraph[2] and copy the span as the answer. Most systems developed for these tasks treat it as classification: predicting the start position of the answer and the end position of the answer. These datasets usually have an average answer length around 4-5 words. Given that the answers must be extractive, the inherent complexity of the answer is limited. It is, for example, quite difficult to answer an immensely complex question with a simple phrase. Further, because of the simplicity

[2] As a technicality, the Natural Questions dataset contains a long answer task, which is a paragraph-level extractive task.

of these questions, the datasets assume the input information of a single paragraph is sufficient for answering the question.

In this chapter, we focus instead on models that can tackle the challenge of *long-form* question answering. This task focuses on generating paragraph-length explanations to open-ended questions, often starting with *how* or textitwhy. It has two major challenges that make it an interesting testbed for the development of knowledge-based generation systems. The first challenge of is the length and diversity of answers that span multiple sentences: questions are complex and cannot be easily addressed by a short response or by extracting a word or phrase from an evidence document. The second challenge is the length and diversity of the content from knowledge sources required to answer our questions. Evidence is queried from the web to answer each question, which produces a large quantity of possible input evidence to use to answer the question. In contrast to previous datasets where the human written answer could be found with lexical overlap methods, long-form question answering poses a significant challenge in siphoning out important information, as no single sentence or phrase contains the full answer.

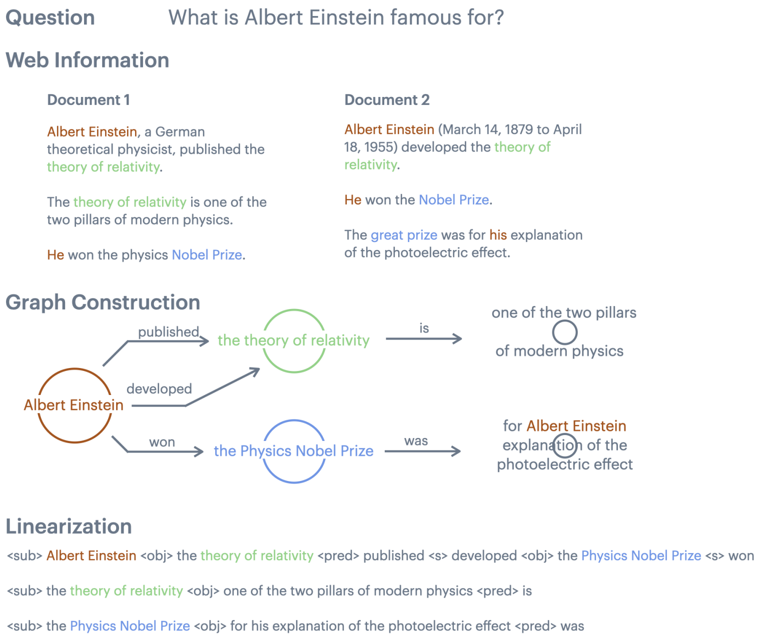
9.3 USING LOCAL KNOWLEDGE GRAPH CONSTRUCTION TO SCALE SEQ2SEQ MODELS TO MULTI-DOCUMENT INPUTS

9.3.1 *Introduction*

Effective information synthesis is at the core of many Natural Language Processing applications, such as open-domain question answering and multi-document summarization. In such tasks, a fundamental challenge is the ability to distill relevant knowledge from hundreds of thousands of tokens of noisy and redundant input such as webpages. Current approaches predominantly conduct TF-IDF-based information extraction to identify key portions of the information, and then provide this as sequential input to a Sequence-to-Sequence (Seq2Seq) model. The sub-selected portions are limited to a few thousand words, as models often struggle to encode much longer sequences.

In this work, we propose a mechanism to re-structure free text into local knowledge graphs that are then linearized into sequences, creating a canonical form in which information is presented to models. By constructing a graph intermediary, redundant information can be merged or discarded, producing substantially compressed input — small enough to be fully encoded by Seq2Seq models. Such a method can be seen as merging previous work on symbolic knowledge bases for information extraction with newer approaches using deep neural networks to encode knowledge.

Our approach, shown in Figure 9.1, takes a query and its corresponding multi-document web search results and builds for each query a specific local knowledge graph. We present several modeling contributions to effectively encode the entire graph as a sequence and attend to the most relevant portions within this linearization. We demonstrate the effectiveness of this approach on two large-scale generative tasks with both long and noisy multi-document web input and paragraph length output: long-form



Albert Einstein, a German theoretical physicist, published the theory of relativity.

The theory of relativity is one of the two pillars of modern physics.

He won the physics Nobel Prize.

Albert Einstein (March 14, 1879 to April 18, 1955) developed the theory of relativity.

He won the Nobel Prize.

The great prize was for his explanation of the photoelectric effect.

<sub> Albert Einstein <obj> the theory of relativity <pred> published <s> developed <obj> the Physics Nobel Prize <s> won

<sub> the theory of relativity <obj> one of the two pillars of modern physics <pred> is

<sub> the Physics Nobel Prize <obj> for his explanation of the photoelectric effect <pred> was

FIGURE 9.1: **Multi-Document Input to Linearized Graph.** Multi-document input resulting from web search queries are converted to a graph structured knowledge base using coreference resolution and information extraction, then linearized into a sequence for Seq2Seq models. Color indicates coreference resolution. Node weight is indicated by circle radius and edge weight by line thickness.

question answering on the ELI5 dataset (Fan et al., 2019b) and Wikipedia lead paragraph generation as a multi-document summarization problem (Liu et al., 2018b).

9.3.2 *Related Work*

Interest in generative sequence modeling has intensified due to recent improvements (Peters et al., 2018; Devlin et al., 2019; Radford et al.), making the challenge of information synthesis more relevant. In contrast to extractive tasks which only require models to identify spans and can do so effectively on long documents by looking at the paragraphs independently, generative sequence models must combine multiple pieces of evidence from long and noisy multi-document input to generate correct and convincing responses.

Multi-Document Input

Previous work in multi-document summarization (Barzilay et al., 1999) applies various techniques to handle long input, including clustering to find similar information (Honarpisheh et al., 2008), extractive methods to select relevant sentences (Daumé III and Marcu, 2002; Gillick and Favre, 2009; Berg-Kirkpatrick et al., 2011; Di Fabbrizio et al., 2014; Bing et al., 2015; Cao et al., 2017) including maximal marginal relevance (Fabbri et al., 2019), and incorporating queries (Baumel et al., 2018) and graphs (Ganesan et al., 2010; Yasunaga et al., 2017). However, there are few large scale multi-document summarization datasets and many approaches have focused on extractive selection or hybrid extractive-abstractive models. In this work, we use graph construction to re-structure multi-document input for abstractive generation.

Advancements in question answering have examined performance on datasets with multi-document input, such as TriviaQA (Joshi et al., 2017). Various approaches have been proposed, including leveraging TF-IDF and bigram hashing with an RNN to find relevant information (Chen et al., 2017), models that score individual paragraphs for sub-selection (Clark and Gardner, 2017), and nearest neighbor search with paragraph re-ranking (Das et al., 2018a). However, these approaches have been applied to extractive question answering tasks that require span identification, rather than abstractive text generation in an information synthesis setting.

Using Knowledge Bases

Previous work has explored various ways of representing information in knowledge bases (Bordes et al., 2011) and improving these representations (Chen et al., 2013). Knowledge bases have been leveraged to improve performance on various tasks, from coreference resolution (Ng and Cardie, 2002) and question answering (Zheng, 2003; Bao et al., 2014; Cui et al., 2017; Sun et al., 2018) to signal processing (Bückner et al., 2002). Various works convert text into Abstract Meaning Representations (Liu et al., 2018a) for domains such as news (Vossen et al., 2015; Rospocher et al., 2016) and link nodes to large knowledge bases such as DBPedia (Auer et al., 2007). Wities et al. (2017) combine open information extraction with coreference and lexical inference to build knowledge representations. They apply this to tweets and analyze the accuracy on various aspects of graph construction. Das et al. (2018b) construct graphs from procedural text to track entity position to answer when and if entities are created, destroyed, or moved. In contrast, we build graphs from substantially longer multi-document input and use them for multi-sentence text generation.

Recently, many have explored neural architectures that can encode graph structured input (Bruna et al., 2013; Kipf and Welling, 2016; Beck et al., 2018; Zhou et al., 2018; Xu et al., 2018; Lai et al., 2019). These models often represent graphs as adjacency matrices to generalize architectures such as convolutional networks to graph inputs. Rather than encoding a static knowledge graph or leveraging external knowledge graphs, we build a local graph for each query and model these using standard Seq2Seq models. We leave the incorporation of graph networks for future work.

9.3.3 *Graph Construction*

We describe how symbolic graph representations of knowledge can be constructed from text. Our approach assumes a multi-document input (such as web pages) resulting from the execution of a query. The graph construction process (1) compresses the web search input to a significantly smaller size, allowing models to encode the entirety of the compression, and (2) reduces redundancy through merge operations, allowing relevant information to be more easily identified.

▸ TEXT TO TRIPLES TO GRAPH Graph construction proceeds in several steps outlined in Figure 9.2. We apply *Coreference Resolution* (Clark and Manning, 2016a,b)[3] and *Open Information Extraction* (Stanovsky et al., 2018)[4] to

[3] We use the implementation available here: `https://github.com/huggingface/neuralcoref`

[4] We use the implementation available here: `https://github.com/gabrielStanovsky/supervised-oie`

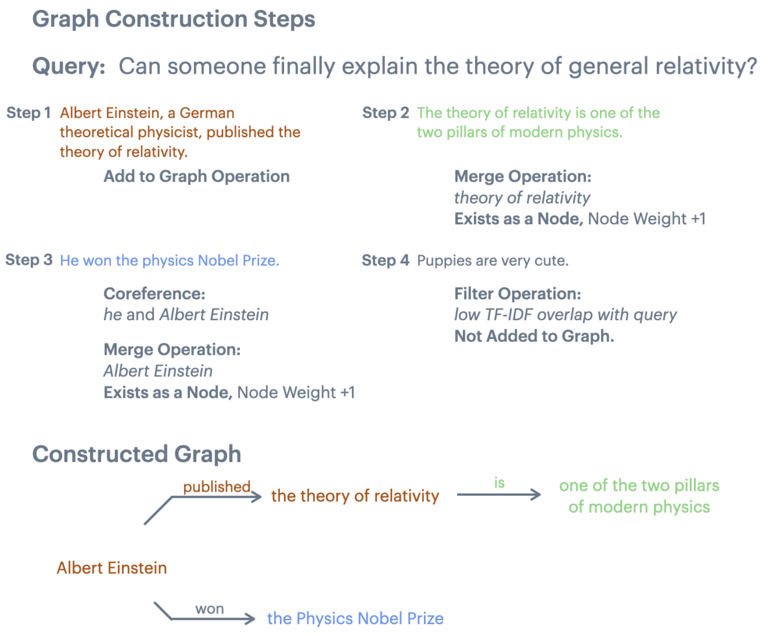

Graph Construction Steps
Query: Can someone finally explain the theory of general relativity?
Step 1 Albert Einstein, a German theoretical physicist, published the theory of relativity.
Add to Graph Operation
Step 2 The theory of relativity is one of the two pillars of modern physics.
Merge Operation: theory of relativity
Exists as a Node, Node Weight +1
Step 3 He won the physics Nobel Prize.
Coreference: he and Albert Einstein
Merge Operation: Albert Einstein
Exists as a Node, Node Weight +1
Step 4 Puppies are very cute.
Filter Operation: low TF-IDF overlap with query
Not Added to Graph.
Constructed Graph
Albert Einstein
published
the theory of relativity
is
one of the two pillars of modern physics
won
the Physics Nobel Prize

FIGURE 9.2: Steps of **Graph Construction**. Color relates the document sentence used to produce the graph output.

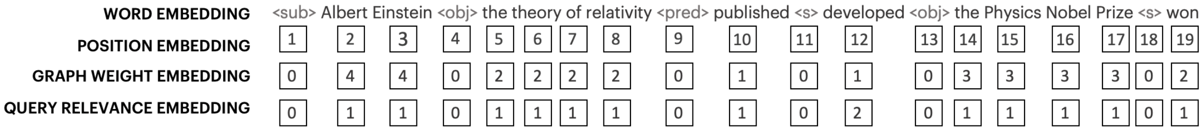


FIGURE 9.3: **Graph Attribute Embeddings**. In addition to word and position embeddings, models receive a Graph Weight embedding to encode node and edge weight and a Query Relevance embedding that encodes search result rank.

convert sentences into a *Triple* of the form (subject, predicate, object). The sentence *Albert Einstein, a German theoretical physicist, won the Nobel Prize* would become (*Albert Einstein, won, the Nobel Prize*).

A graph is constructed using the triples by representing subjects and objects as nodes connected by predicates as directed edges. For example, the triple would become *Albert Einstein* $\xrightarrow[\text{won}]{}$ *the Nobel Prize*. Nodes and edges have a *name* property that is the text they represent. They also have a *weight* property that denotes the number of times that node or edge has appeared. For example, in Figure 9.1, the node with name *Albert Einstein* has weight 4 and edge with name *won* has weight 2.

▸ MERGING NODES AND EDGES When subsequent triples are added to the graph, they are merged with the existing graph if they already exist to reduce information replication. To merge nodes, the TF-IDF overlap of the new node's name is calculated with the existing graph node names, and the new node is merged into an existing node if the TF-IDF is higher than some threshold (see Figure 9.2, steps 2 and 3 for example merge operations). Edges are merged similarly with existing edges between the same two nodes. Such merge operations allow strings such as *the Nobel Prize* and *Nobel Prize* to be represented as one node rather than separately. Similarly, coreference resolution aids in merging — by identifying that *Albert Einstein* and *He* refer to the same entity and thus merging them, the construction of the graph reduces redundancy. The size of the graph can be modified by controlling which triples are added using TF-IDF overlap (see Figure 9.2, step 4). TF-IDF overlap of the triple with the question can be used to determine if the triple

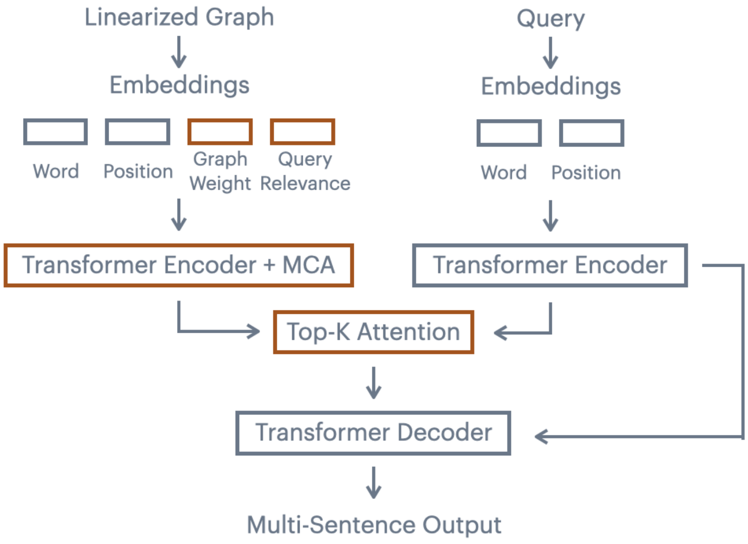


FIGURE 9.4: **Model Architecture.** Gray indicates standard Transformer elements, Green indicates modification.

contains relevant information. This improves robustness to noisy web search input and helps filter entirely irrelevant portions, such as scraped HTML tags.

9.3.4 *Modeling Graphs as Sequences*

Current models for text generation often use Seq2Seq architectures such as the Transformer (Vaswani et al., 2017). These models are designed to encode sequences rather than graphs. We describe now how to convert a graph into a structured input sequence. Our complete model will take as input a linearized graph by encoding graph attributes such as node and edge weight as embeddings. We add hierarchical and memory-compressed attention mechanisms to scale Seq2Seq models to encode the full graph and attend to the most relevant information within it (Figure 9.4), and finally we integrate the benefits of language modeling using multi-task training.

Graph to Sequence

- LINEARIZATION To represent the graph as a sequence for Seq2Seq, it is linearized into a structured standard form of subject node, object node, and predicate edge, separated by indicator tokens, as shown in Figure 9.1. For example, the linearization *<sub> Albert Einstein <obj> the Nobel Prize <pred> won* would be created. The linearization is accomplished through graph traversal in a breadth-first manner following the directed edges formed by predicates and starting from the node with the largest weight as the root. For two nodes that are connected by multiple predicates, the predicates are concatenated (shown in Figure 9.1), so a linearization such as *<pred> won <s> received* would indicate that Albert Einstein both won and received the Nobel Prize.

- ENCODING GRAPH INFORMATION Transformer Seq2Seq models have two embeddings: a word embedding and a position embedding. Simply linearizing the graph, however, loses attribute information such as node and edge weight. Instead, we encode these attributes as embeddings in addition to the word and position embeddings.

To represent *Graph Weight* (GW), node and edge weight is provided as an embedding for each token. The node weight and edge weight are equivalent to the number of merge operations + 1. For example, if *Albert Einstein* occurred 4 times in the text, the GW embedding for the tokens *Albert* and *Einstein* would be 4, as shown in Figure 9.3.

We encode a *Query Relevance* (QR) embedding to represent the relevance of the web search to the query as ranked by the information retrieval system (e.g. search engine). Information from the top web search results is likely more relevant than information from the last web search results, so providing this embedding allows the model to distinguish between these different information sources. In Figure 9.3, tokens representing sentences from the first document have QR embedding 1, and tokens from the second document have value 2.

Models now have access to several different types of embeddings, but all embedding information contributes equally as there is no mechanism to distinguish between them. We introduce a mechanism for models to scale the graph embeddings. We denote the embedding for position t as e_t, such that e_t^{word} is the word embedding.

For the GW embedding, models learn a gating function g based on the word and GW embeddings. Such a mechanism provides capacity for the model to decide when the additional embeddings are useful based on the words in the input. The gate is calculated by applying an MLP W to the concatenation of the word and GW embeddings. The learned gate is then applied to GW embeddings to create the output h:

$$\begin{aligned} g_t^{\text{GW}} &= W[e_t^{\text{GW}}; e_t^{\text{word}}] \\ h_t^{\text{GW}} &= g_t^{\text{GW}} \circ e_t^{\text{GW}} \end{aligned}$$

Models learn a gating mechanism for the QR embedding in a similar manner. The full embedding the model receives is as follows:

$$e_t^{\text{word}} + e_t^{\text{pos}} + [h_t^{\text{GW}}; h_t^{\text{QR}}]$$

Hierarchical Attention

One challenge in modeling long sequences is that attention mechanisms struggle to make sharp selections when softmax-ing over long sequences (Fan et al., 2018b). When attention is blurry, there lacks a strong distinction between noise and relevant information.

We assume that graphs are constructed from query-based web search input and thus leverage this query to learn a subselection operation using *hierarchical top-k attention*, depicted in Figure 9.4. The query is encoded with a Transformer encoder and the linearized graph with another Transformer encoder. We model the interaction between the query and the input sequence (e.g. web search results or linearized graph) by computing an attention distribution between the question and the input, then selecting the top

k positions with the most attention weight. Such a mechanism can be thought of as building a query-dependent representation of the most relevant knowledge, which is commonly done in question answering architectures like BiDAF (Seo et al., 2017). The top k operation limits the number of tokens, making the attention mechanism sharper.

Scaling to Encode the Graph

Recent progress has improved the ability of language models to process longer sequences (Sukhbaatar et al., 2019a; Dai et al., 2019), but models remain limited in their capacity to encode long documents. The multi-document results of query-based web search have hundreds of thousands of tokens, beyond the limit of current Seq2Seq models to handle. For example, the ELI5 dataset provides an average of 200K tokens of web search input. However, by compressing the web search results into a knowledge graph, we significantly reduce the number of tokens by an order of magnitude and make it possible for a model to access the entirety of the search information.

To represent the full graph, models must scale to encode around 10K input tokens. The attention mechanism in Transformer architectures becomes computationally expensive for sequences of this length. Instead, we experiment with the *Memory-Compressed Attention* (MCA) mechanism proposed for language models in (Liu et al., 2018b)[5] and apply it to the encoder side of Seq2Seq models. At each self-attention layer, MCA alternates between (1) local attention, computed between smaller chunks of tokens and (2) strided convolutions to reduce the number of keys and values used in attention computation. By adding the MCA mechanism to the encoder (*E-MCA*), we are able to encode the complete linearized graph.

Multi-tasking with KB Completion

Fan et al. (2019b) used multi-task training on language modeling and various Seq2Seq tasks to incorporate the benefits of language modeling in Seq2Seq models. We extend this by training additionally on *knowledge graph completion*. Models receive at training time sequences of a linearized graph with nodes, edges, or both selectively masked and must predict the missing content words. For example, models might receive as input *<sub> Albert Einstein <obj> [mask] <pred> won* and need to predict *the Nobel Prize*. This can be seen as both a multi-word extension of the masked language model training proposed in (Devlin et al., 2019) and as learning the task of knowledge base completion (Lacroix et al., 2018; Bordes et al., 2011). At training time, the tasks are distinguished using an indicator token in the input.

9.3.5 *Experimental Setup*

We evaluate our approach on two datasets with multi-document web input and multi-sentence abstractive output. We use Seq2Seq models that leverage a query concatenated with web search results that have been processed into a supporting document — e.g. TF-IDF subselection, linearized graph, etc. — to generate long output.

[5] In Liu et al. (2018b), the mechanism is termed *DMCA* as it is applied on the decoder side

Datasets and Evaluation

- ELI5 First, we experiment with the *Explain Like I'm Five* (ELI5) (Fan et al., 2019b) question answering dataset of 270K complex questions paired with multi-sentence, explanatory answers (130 words on average). To facilitate question answering, the dataset provides the top 100 web search hits from querying the question, which results in 200K words on average. See Appendix for examples.

  Previous work (Fan et al., 2019b) used TF-IDF to find sentences in the web search that have the largest overlap with the question and created a *TF-IDF extraction* of about 850 words as input for Seq2Seq models. Instead, we construct a local knowledge graph for each question from the 100 web search hits. Following the average length of the TF-IDF support document constructed in (Fan et al., 2019b), we experiment with modeling the first $N = 850$ tokens of the linearized graph, then scale to encode the entire graph using E-MCA.

- WIKISUM Second, we experiment on the *WikiSum CommonCrawl* (Liu et al., 2018b) summarization dataset[6] with 1.5 million examples. This task formulates Wikipedia lead paragraph generation as a multi-document summarization problem, where the paragraph must be generated using the cited article references and other queried content from web search. The query used is the title of the Wikipedia article. See Appendix for examples.

  Previous work (Liu et al., 2018b) applied *TF-IDF Ranking* to order the paragraphs of web search given a query. Models receive the re-ordered paragraphs ranked by TF-IDF as input. Liu et al. (2018b) model the first $N = 500$ words of this re-ranking and then $N = 11,000$ using MCA. We construct the knowledge graph for each Wikipedia article from the first 200K words of the ranked web search results,[7] and experiment with encoding 500 and $11,000$ tokens.

- EVALUATION Both tasks evaluate the multi-sentence generation output against the gold output using F1 ROUGE. On WikiSum, we report only ROUGE-L following (Liu et al., 2018b).

  We conduct a comparative human evaluation on the ELI5 dataset. We use crowdworkers to examine the responses of two models on 300 different questions from the test set. For each question, 3 evaluators are shown two answers and asked to choose the one they prefer. To reduce variance, answers are standardized at 150 words each.

Training and Generation

To reduce the vocabulary size of varied web document content, we apply byte-pair encoding (Sennrich et al., 2016) to generate 40K codes for each dataset. We implement our models in fairseq-py (Ott et al., 2019) using the Transformer Big architecture and training schedule described in (Vaswani et al., 2017). Detailed parameters are listed in the Appendix. For generation, we use beam search with beam size 5 and tune a minimum and maximum length on the validation set.

[6] `https://github.com/tensorflow/tensor2tensor/tree/master/tensor2tensor/data_generators/wikisum`

[7] Average length of provided web input is around 50K words, and maximum length is around 900K words

Model	**Input Length**	**ROUGE 1**	**2**	**L**
Q + D to A*, TF-IDF	avg 850	28.3	5.1	22.8
Q + D to A, MMR	avg 850	28.1	5.0	22.9
Multi-task*	avg 850	28.9	5.4	23.1
Multi-task Triples	850	29.0	5.2	23.2
Multi-task Top20 Trip.	avg 570	28.8	5.3	23.2
Q + D to A Graph	850	28.8	5.3	23.3
Multi-task Graph	850	29.5	5.6	23.6
+ Top-100 Attention	850	29.7	5.7	23.8
+ E-MCA	11K	30.0	5.8	24.0

TABLE 9.1: **Answer Generation on ELI5** using Seq2Seq models receiving the **Q**uestion and a support **D**ocument (e.g. TF-IDF selection, Triples, Linearized Graph) to produce the **A**nswer.
* denotes results from (Fan et al., 2019b).

Model	**InputLen**	**ROUGE-L**
T + D to P*	500	34.2
LM + D-MCA*	11K	36.2
T + D to P	500	33.8
Multi-task	500	34.4
Multi-task Graph	500	34.9
+ Top-100 Attention	500	35.2
+ E-MCA	11K	36.5
LM + D-MCA + MoE-256*	7.5K	38.8

TABLE 9.2: **Lead Paragraph Generation on WikiSum CommonCrawl** using Seq2Seq models receiving the **T**itle and a support **D**ocument (e.g. TF-IDF ranking, Linearized Graph) to produce the Lead **P**aragraph.
* denotes results from (Liu et al., 2018b) that use data scraped from unrestricted web search, not the static CommonCrawl version.

Baselines

We compare our results to the Transformer sequence models presented in (Fan et al., 2019b) for ELI5 and (Liu et al., 2018b) for WikiSum.

We evaluate three additional baseline models:

- *Sentence Selection with Maximal Marginal Relevance:* (Fan et al., 2019b) used TF-IDF to identify relevant sentences in the web documents to form a support document of around 850 words. However, recent work (Fabbri et al., 2019) has shown that using maximal marginal relevance is an effective strategy for selecting relevant information while reducing redundancy. We explore using MMR to select sentences from the web text to concatenate to form a support document.
- *Seq2Seq Multi-task Triples:* To examine the impact of solely restructuring the input into Open IE Triples but not leveraging graph construction to reduce redundancy, we experiment with a Triples Only baseline. The triples are concatenated to form the input.
- *Seq2Seq Multi-task Top 20 Triples:* As an alternative to using graph construction to compress the full set of Open IE Triples, we explore using TF-IDF overlap with the query to find the most relevant information. We select the top 20 triples to concatenate as input.

9.3.6 *Results*

We examine the performance of our proposed approach and the choices made in graph construction and modeling. We analyze the quality of the compression created by graph construction and the robustness and interpretability of this process.

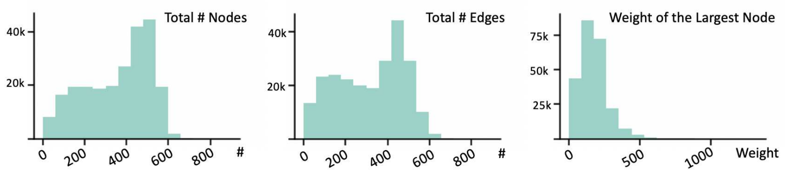


FIGURE 9.5: (Left) Distribution of Number of **Nodes**, (Middle) Number of **Edges**, (Right) **Weight** of the Largest Node in graph construction on the ELI5 training set.

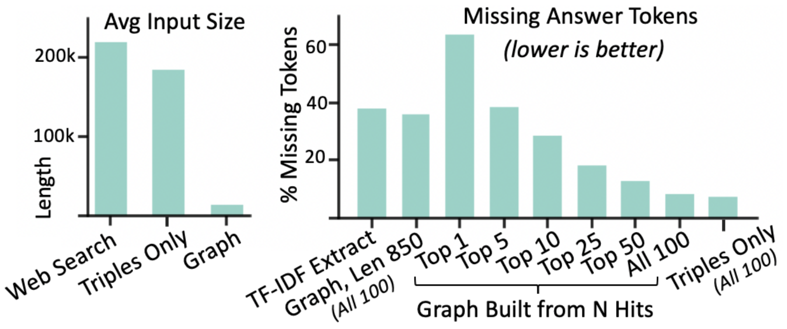


FIGURE 9.6: (Left) Graph construction drastically reduces input size by an order of magnitude. (Right) Graph construction encodes more tokens present in the answer compared to TF-IDF extraction and building the graph from more search hits increases answer token coverage. Analysis on ELI5 for both plots.

Linearized Graph Improves Performance

In Table 9.1, we compare our methods to various baselines on the ELI5 dataset. Using MMR to select the most relevant non-redundant input is similar to the TF-IDF baseline from Fan et al. (2019b). The Seq2Seq Multi-task Triples baseline standardizes the input by forming triples but does not remove redundant triples. It produces marginally better results compared to the baseline Multi-Task model. Sub-selecting to the Top 20 Triples is harmful, as similar text has high TF-IDF overlap with the query so redundant information is selected. Creating the graph structure brings an improvement of around 0.6 ROUGE-1.

Similar trends are seen for the WikiSum dataset in Table 9.2, where graph construction improves the Multi-task model by 0.5 ROUGE-1. These improvements are statistically significant at the 95% confidence level.

For both datasets, a further improvement is seen by using the hierarchical attention mechanism to attend to only the most relevant information in the linearized graph input. For ELI5, it brings an additional 0.2 ROUGE-1 improvement and on WikiSum a 0.3 ROUGE-1 improvement.

By using MCA to scale Seq2Seq models to encode the entire graph, further gains can be seen. Particularly in information synthesis tasks, prior work has shown the importance of reading more information. Liu et al. (2018b) achieved a 2-point ROUGE improvement by reading 11K tokens instead of 500. In our setting, E-MCA improves our results around 0.3 ROUGE on ELI5 and 1.3 ROUGE on WikiSum. We display random generations from both datasets in the Appendix.

We use human evaluation to compare the Multi-task baseline to the Multi-task Graph + Top-k Attention model. **57.4%** of evaluations prefer the Multi-task Graph + Top-k Attention model. We conduct a two-tailed binomial test and find this result is statistically significant with $p = 0.003$.

Analysis of Modeling Choices

Model	**ROUGE-1**
(a) Iterative Removal of Model Components	
Multi-task Graph	29.4
- Graph Embeddings	29.1
- KB Completion Multi-tasking	28.9
- LM Multi-tasking from (Fan et al., 2019b)	28.4
(b) Removing Graph Embedding Components	
Graph	
+ Gated Graph Weight + Query Relevance	28.6
No Graph Weight Embedding	28.4
No Query Relevance Embedding	28.3
No Gating	28.4
(c) Varying Number of Hits in Graph	
Multi-task Graph + Top-k Attention + E-MCA	
with Graph on 5 Search Hits	28.8
with Graph on 10 Search Hits	29.3
with Graph on 50 Search Hits	29.6
with Graph on 100 Search Hits	29.9
(d) Varying k in Hierarchical Top-k Atttention	
Multi-task Graph + E-MCA +	
Top-$k = 50$	29.1
Top-$k = 100$	29.5
Top-$k = 250$	29.4
Top-$k = 500$	29.3
Top-$k = 1000$	29.2

TABLE 9.3: **Ablations** on the ELI5 Validation Set. We display four different ablations to analyze the effect of various modeling choices on model performance.

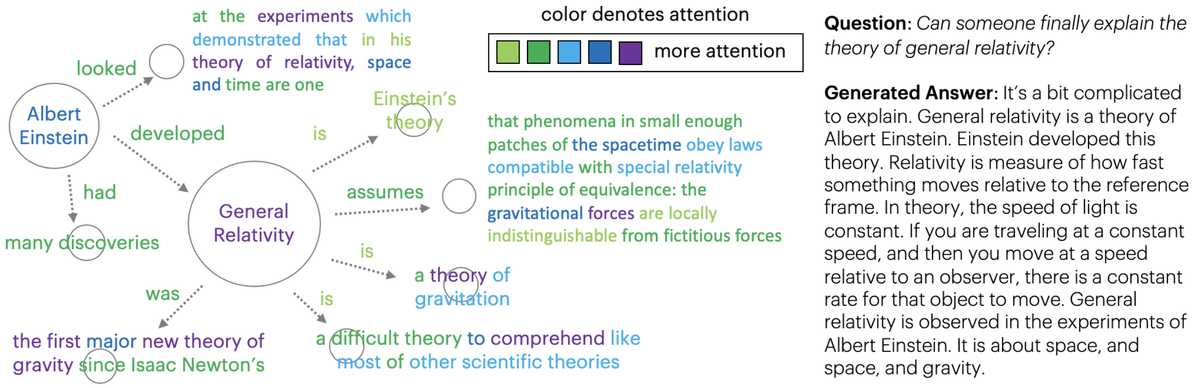
Model	**Input**	**ROUGE-1**
Seq2Seq Q + D to A	TF-IDF	28.3
Seq2Seq Q + D to A	Web	25.9
+ Shuffle	Web Shuffle	24.4
Seq2Seq Q + D to A Graph	Web	28.5
+ Shuffle	Web Shuffle	28.2

TABLE 9.4: Importance of **Web Search Relevance** on Validation for ELI5, modeling 850 input words.

- ABLATION ON MODEL COMPONENTS Table 9.3(a) sequentially removes the graph embeddings, the knowledge-base completion multi-tasking, and the multi-tasking from (Fan et al., 2019b) and reveals that each of these is important for performance.

- GRAPH ATTRIBUTE EMBEDDINGS Table 9.3(b) displays the effect of removing the graph attribute embeddings and gating mechanism. Removing each is slightly harmful, and the combination of all three together provide the best performance.

- MORE WEB SEARCH DOCUMENTS Figure 9.6 (right) shows that graph construction with more web search information is important for answer token coverage. The graph on the top search hit alone is missing 64% of the answer tokens, but this decreases as more search hits are used. Table 9.3(c) indicates that this lack of coverage of the answer tokens correlates with generation quality. Models receiving a graph built on the first 5 search hits alone are substantially worse than all 100 hits.


color denotes attention
🟩 🟩 🟦 🟦 🟪 more attention
Albert Einstein
looked
at the experiments which demonstrated that in his theory of relativity, space and time are one
developed
had
many discoveries
General Relativity
is
Einstein's theory
assumes
that phenomena in small enough patches of the spacetime obey laws compatible with special relativity principle of equivalence: the gravitational forces are locally indistinguishable from fictitious forces
is
a theory of gravitation
was
the first major new theory of gravity since Isaac Newton's
is
a difficult theory to comprehend like most of other scientific theories

Question: *Can someone finally explain the theory of general relativity?*

Generated Answer: It's a bit complicated to explain. General relativity is a theory of Albert Einstein. Einstein developed this theory. Relativity is measure of how fast something moves relative to the reference frame. In theory, the speed of light is constant. If you are traveling at a constant speed, and then you move at a speed relative to an observer, there is a constant rate for that object to move. General relativity is observed in the experiments of Albert Einstein. It is about space, and space, and gravity.

FIGURE 9.7: **Interpretable Attention** of Seq2Seq models on a subgraph when answering a question in ELI5

▸ TOP-k ATTENTION Table 9.3(d) shows the effect of the Top-k Hierarchical Attention mechanism for various values of k. Attending to too many tokens lowers ROUGE — for the ELI5 task of writing approximately 130 word answers, attending to 1000 input tokens likely means the model is focusing on irrelevant information and 50 tokens is too few.

Graph Improves Answer Token Coverage Despite Compression

Figure 9.5 displays the distribution of the number of nodes, edges, and the largest node weight for each local graph built on the ELI5 dataset. The 100 web search results are compressed to a few hundred nodes. By merging redundancy and trimming irrelevant triples from the graph, the input is reduced by an order of magnitude (Figure 9.6, left).

Despite compression, the graph retains more answer tokens than TF-IDF subselection. Figure 9.6 (right) displays the percentage of answer tokens not present in the input. The TF-IDF Extraction from (Fan et al., 2019b) is missing 38% of tokens. The graph constructed on all 100 web search results is only missing 8.7% of tokens, but has around 10K words. When analyzing just the first 850 tokens to match the average length of the TF-IDF extraction, the graph is better (only missing 35% of tokens). Further, the merging and discarding operations done during graph construction do not have a large effect on answer token coverage — the full set of triples marginally reduces the percentage of answer tokens missing to 7.3% instead of 8.7%. This indicates that much of the information in the full set triples is redundant and unnecessary for good answer token coverage.

Graph Representation is More Robust to Poor Search Relevance Ordering

We analyze the robustness of our approach to the ordering of web search results in Table 9.4. Instead of constructing the graph from the first web search result to the last, we shuffle the web search results and construct the graph on this shuffled input. We compare this to modeling the web search results directly (no TF-IDF retrieval) and a model that receives this shuffled web search input. The graph is more robust to shuffling — as more information can be encoded in the graph due to its compression effect, the search hit ordering is less critical.

Interpretable Attention on Subgraphs

Figure 9.7 shows an example of the nodes and edges the model focuses upon most when answering a question on ELI5. To construct this visualization, we calculate the top nodes the model attends to and then their top edges. The model attention on a sub-portion of the linearized input can be visualized as an interpretable graph that corresponds well to the model's generated answer. For example, the relationship *General Relativity* $\underset{\text{is}}{\longrightarrow}$ *Einstein's theory* becomes the generated sentence *General Relativity is a theory of Albert Einstein*.

9.3.7 *Conclusion*

Many open-domain NLP tasks rely upon multi-document input from the web to facilitate tasks such as answering questions or writing summaries, but current approaches struggle to encode the entirely of this information. We propose constructing one knowledge graph per query and show that this method compresses information and reduces redundancy. We show on two abstractive generation tasks that using the linearized graph achieves better performance than TF-IDF retrieval.

▸ Conclusion

In this section, we confront a fundamental need to incorporate knowledge in text generation systems. Without information, generation models would be unable to create accurate outputs to aid users across a variety of tasks. We develop a number of different techniques that incorporate information at increasingly greater scale: from single to multiple documents, then all of Wikipedia, and beyond that — the open web.

Systems that operate at such scale have numerous challenges, many related to efficiently scanning and identifying the most relevant subset of knowledge when the amount of possible information to look at is incredibly large. We create scalable, modular architectures such as KIF that utilize fast computation from libraries like `faiss`.

Beyond the challenge of scalability, architectures that optimally incorporate the knowledge are necessary for the development of successful generation systems. For example, our model for writing entire Wikipedia articles decomposes the article writing task into that of writing specific article subsections, making it possible to search for information subsection by subsection.

Part IV

EPILOGUE

Conclusion

"I may not have gone where I intended to go, but I think I have ended up where I needed to be."
—Douglas Adams

▸ LOOKING BACKWARD

This thesis investigates two central challenges in text generation systems: generation beyond English and incorporation of knowledge. These components are central to the creation of generative natural language models that can handle deeply complex, human-like writing objectives and be useful for people all over the world. Imagine a seamless experience where incredibly long Messenger chat threads in multiple different languages could be instantly summarized for people trying to catch up in the conversation, or people working on written works such as encyclopedic articles could immediately access organized briefs of related information and start with a fully-written article as a skeleton to edit. These experiences seem not too far from the technology we already take for granted and utilize every day. However, they also represent fundamental advances that we as a research community in text generation still need to make — as well as experiences that could become possible by integrating various advances made in this thesis.

The first half of this thesis studies text-to-text generation problems and focuses on advancements in multilingual generation. We develop a mining-based method that can create supervised training datasets for a variety of different languages, applied to the task of sentence simplification. Subsequently, we propose a method to utilize multilingual pretraining and cross-lingual embeddings to create a multilingual model capable of decoding into over 20 different languages, from the same structured English AMR input. Taken together, these methods are applicable to a variety of other tasks and represent complementary approaches to multilingual generation — automatically creating datasets and effectively representing multiple languages in one model.

The second half of this thesis faces the challenge of incorporating knowledge. Humans every day learn new information, and perform actions that synthesize and integrate knowledge from various diverse sources. Imagine, for example, deciding if you want to see the latest movie that was released — you might read a few reviews and a spoiler-free news article to decide if you want to see it or not. These tasks seem simple to us, but are complex for machines because fundamentally they require effective discovery and utilization of information. We contribute a variety of approaches and architectures that tackle both retrieving information from large external knowledge sources and effectively incorporating it into standard sequence-to-sequence models. Further, we continuously scale these methods to handle larger and larger knowledge sources, from individual documents required to answer questions to the open web evidence needed to write entire Wikipedia articles. We focus on efficient scaling and modular architectural components that can extend to a variety of different tasks.

▸ Looking Forward

What is next in text generation? We end this work by discussing a few of the self-perceived most interesting research directions.

Implicit Parameterized Knowledge. This work focuses on utilizing knowledge with the assumption that the knowledge is *external* and thus needs to be incorporated into the model as some sort of input. Recent work has demonstrated strong potential for models that intrinsically have *implicit* knowledge, or information stored in their parameters, learned from training on incredibly large corpora. This implicit knowledge can be demonstrated by e.g. using models to fill in facts in a sentence in masked infilling tasks. While such properties are incredibly interesting, it is ultimately an area for further study. For example, how can implicit knowledge represent different viewpoints or sources of information, or resolve contradictions, or explain which original references produced such information?

Practical Utilization. Recent advances in generation, and particularly incorporating knowledge in generation, utilize large pretrained models trained on web-scale corpora. While this has produced models that write with surface-level fluency, an area for future research remains the practical usage of such models. Effectively being able to distill, quantize, or otherwise prune or reduce model size while retaining performance will be critical. This is particularly relevant for many architectural configurations that utilize knowledge, as often the knowledge is pre-encoded as a fixed, large size set of dense vectors, which would similarly require storage or computation to access. On the positive side, many of these limitations have the potential to be addressed not necessarily through improvements in machine learning, but advances in hardware (GPUs, mobile chips) and other software (even faster nearest neighbor search, for instance).[8]

Global Applicability. Finally, the extensibility of current dominant techniques to a wide range of languages and domains remains a question. Often, leading approaches are a function of the environment they are developed in — meaning, our current methods likely rely on strong assumptions, such as large quantities of clean monolingual data in various languages. As we consider the creation of models that truly function for people all over the world, we will need to develop novel techniques that operate in truly low-resource scenarios.

[8] Of particular note, I am interested in techniques that can be combined together, as most likely one method alone may not achieve the required size for on-device computation. It is not a focus on this thesis, but methods such as quantization (Fan et al., 2020b) and pruning (Fan et al., 2020a) could be interesting directions to pursue.

Bibliography

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2020. Large scale knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. *arXiv preprint arXiv:2010.12688*.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. 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 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. 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 2440–2452.

Sandra M Aluísio, Lucia Specia, Thiago AS Pardo, Erick G Maziero, and Renata PM Fortes. 2008. Towards Brazilian Portuguese automatic text simplification systems. In *Proceedings of the eighth ACM symposium on Document engineering*, pages 240–248.

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 *Procedings of ACL 2020*, pages 4668–4679.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier automatic sentence simplification evaluation. In *EMNLP-IJCNLP: System Demonstrations*.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020b. Data-Driven Sentence Simplification: Survey and Benchmark. *Computational Linguistics*, pages 1–87.

Michelle A Amazeen. 2016. Checking the fact-checkers in 2008: Predicting political ad scrutiny and assessing consistency. *Journal of Political Marketing*, 15(4):433–464.

Alessio Palmero Aprosio, Sara Tonelli, Marco Turchi, Matteo Negri, and Mattia A Di Gangi. 2019. Neural text simplification in low-resource conditions using weak supervision. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 37–44.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. 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 4685–4697, Hong Kong, China. Association for Computational Linguistics.

Bence Bago, David G Rand, and Gordon Pennycook. 2020. Fake news, fast and slow: Deliberation reduces belief in false (but not true) news headlines. *Journal of experimental psychology: general*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Siddhartha Banerjee and Prasenjit Mitra. 2015. Wikikreator: Improving wikipedia stubs automatically. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 867–877.

Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. 2014. Knowledge-based question answering as machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 967–976.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of NAACL-HLT 2003*, pages 16–23.

Regina Barzilay, Kathleen R McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*.

Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint arXiv:1801.07704*.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. *arXiv preprint arXiv:1806.09835*.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 481–490. Association for Computational Linguistics.

Pablo Beytía. 2020. The positioning matters: Estimating geographical bias in the multilingual record of biographies on wikipedia. In *Companion Proceedings of the Web Conference 2020*, pages 806–810.

Fadi Biadsy, Julia Hirschberg, and Elena Filatova. 2008. An unsupervised approach to biography production using wikipedia. In *Proceedings of ACL-08: HLT*, pages 807–815.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca J Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging. *arXiv preprint arXiv:1506.01597*.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of" bias" in nlp. *arXiv preprint arXiv:2005.14050*.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning End-to-End Goal-Oriented Dialog. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Jan A Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. 2018. Learning to split and rephrase from wikipedia edit history. *arXiv preprint arXiv:1808.09468*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.

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.

Jürgen Bückner, Martin Pahl, O Stahlhut, and C-E Liedtke. 2002. A knowledge-based system for context dependent evaluation of remote sensing data. In *Joint Pattern Recognition Symposium*, pages 58–65. Springer.

Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. 2019. Retrieval-guided Dialogue Response Generation via a Matching-to-Generation Framework. 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 1866–1875.

Oana-Maria Camburu, Brendan Shillingford, Pasquale Minervini, Thomas Lukasiewicz, and Phil Blunsom. 2019. Make up your mind! adversarial generation of inconsistent natural language explanations. *arXiv preprint arXiv:1910.03065*.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2017. Improving multi-document summarization via text classification. In *Thirty-First AAAI Conference on Artificial Intelligence*.

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.

Thiago Castro Ferreira and Emiel Krahmer. 2019. Surface realization shared task 2019 (MSR19): The team 6 approach. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 59–62, Hong Kong, China. Association for Computational Linguistics.

Sarath Chandar, Sungjin Ahn, Hugo Larochelle, Pascal Vincent, Gerald Tesauro, and Yoshua Bengio. 2016. Hierarchical Memory Networks. *CoRR*, abs/1605.07427.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Danqi Chen, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2013. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. *arXiv preprint arXiv:1301.3618*.

Mingda Chen, Sam Wiseman, and Kevin Gimpel. 2020a. Generating wikipedia article sections from diverse data sources. *arXiv preprint arXiv:2012.14919*.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020b. Kgpt: Knowledge-grounded pre-training for data-to-text generation. *arXiv preprint arXiv:2010.02307*.

Wenlin Chen, David Grangier, and Michael Auli. 2016. Strategies for Training Large Vocabulary Neural Language Models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1975–1985.

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from wikidata. *arXiv preprint arXiv:1702.06235*.

Christopher Clark and Matt Gardner. 2017. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*.

Kevin Clark and Christopher D Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667*.

Kevin Clark and Christopher D Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7057–7067.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485.

William Coster and David Kauchak. 2011. Learning to simplify sentences using wikipedia. In *Proceedings of the workshop on monolingual text-to-text generation*, pages 1–9.

Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2017. Kbqa: learning question answering over qa corpora and knowledge bases. *Proceedings of the VLDB Endowment*, 10(5):565–576.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Marco Damonte and Shay B Cohen. 2018. Cross-lingual abstract meaning representation parsing. In *Proceedings of NAACL-HLT*, pages 1146–1155.

Marco Damonte and Shay B. Cohen. 2019. Structural neural encoders for AMR-to-text generation. 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 3649–3658, Minneapolis, Minnesota. Association for Computational Linguistics.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2018a. Multi-step retriever-reader interaction for scalable open-domain question answering.

Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2018b. Building dynamic knowledge graphs from text using machine reading comprehension. *arXiv preprint arXiv:1810.05682*.

Hal Daumé III and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 449–456. Association for Computational Linguistics.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941. JMLR. org.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128.

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.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*.

Giuseppe Di Fabbrizio, Amanda Stent, and Robert Gaizauskas. 2014. A hybrid approach to multi-document summarization of opinions in reviews. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 54–63.

Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. 2019. Queens are powerful too: Mitigating gender bias in dialogue generation. *arXiv preprint arXiv:1911.03842*.

Emily Dinan, Angela Fan, Ledell Wu, Jason Weston, Douwe Kiela, and Adina Williams. 2020. Multi-dimensional gender bias classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 314–331.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. In *International Conference on Learning Representations*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. *CoRR*, abs/1905.03197.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874.

Ondřej Dušek and Zdeněk Kasner. 2020. Evaluating semantic accuracy of data-to-text generation with natural language inference. *arXiv preprint arXiv:2011.10819*.

Richard Evans, Constantin Orasan, and Iustin Dornescu. 2014. An evaluation of syntactic simplification rules for people with autism. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 131–140.

Alexander R Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R Radev. 2019. Multi-news: a large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749*.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019a. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. In *Proceedings of EMNLP-IJCNLP 2019*, pages 4177–4187.

Angela Fan, David Grangier, and Michael Auli. 2018a. Controllable Abstractive Summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54.

Angela Fan, Edouard Grave, and Armand Joulin. 2020a. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019b. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018b. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Angela Fan, Pierre Stock, Benjamin Graham, Edouard Grave, Rémi Gribonval, Herve Jegou, and Armand Joulin. 2020b. Training with quantization noise for extreme model compression. *arXiv preprint arXiv:2004.07320*.

Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T Yarman Vural, and Yoshua Bengio. 2017. Multi-way, multilingual neural machine translation. *Computer Speech & Language*, 45:236–252.

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–739, San Diego, California. 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 LREC 2020*.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 10–18. Association for Computational Linguistics.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *arXiv preprint arXiv:1903.03862*.

Ana Valeria Gonzalez, Gagan Bansal, Angela Fan, Robin Jia, Yashar Mehdad, and Srinivasan Iyer. 2020. Human evaluation of spoken vs. visual explanations for open-domain qa. *arXiv preprint arXiv:2012.15075*.

Ben Goodrich, Vinay Rao, Peter J Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 166–175.

Isao Goto, Hideki Tanaka, and Tadashi Kumano. 2015. Japanese news simplification: Task design, data set construction, and analysis of simplified text. *Proceedings of MT Summit XV*.

Eduardo Graells-Garrido, Mounia Lalmas, and Filippo Menczer. 2015. First women, second sex: Gender bias in wikipedia. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, pages 165–174.

Edouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. 2017a. Efficient Softmax Approximation for GPUs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1302–1310.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2017b. Improving Neural Language Models with a Continuous Cache. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural Turing Machines. *arXiv preprint arXiv:1410.5401*.

D Graves. 2018. Understanding the promise and limits of automated fact-checking.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the International Conference on Machine Learning*, pages 5695–5704.

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A richly annotated corpus for different tasks in automated fact-checking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 493–503, Hong Kong, China. Association for Computational Linguistics.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? *arXiv preprint arXiv:2010.04119*.

Naeemul Hassan, Bill Adair, James Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. 2015. The quest to automate fact-checking. *Proceedings of the 2015 Computation + Journalism Symposium*.

Seth J Hill. 2017. Learning together slowly: Bayesian learning about political facts. *The Journal of Politics*, 79(4):1403–1418.

Marit Hinnosaar. 2019. Gender inequality in new media: Evidence from wikipedia. *Journal of Economic Behavior & Organization*, 163:262–276.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*.

Mohamad Ali Honarpisheh, Gholamreza Ghassem-Sani, and Ghassem Mirroshandel. 2008. A multi-document multi-lingual automatic summarization system. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

John J Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

David M Howcroft, Anja 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.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv preprint arXiv:2003.11080*.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *International Conference on Learning Representations*.

Encina Calvo Iglesias. 2020. Preparing biographies of stem women in the wikipedia format, a teaching experience. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, 15(3):211–214.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Shaojie Jiang and Maarten de Rijke. 2018. Why are sequence-to-sequence models so dull? understanding the low-diversity problem of chatbots. *arXiv preprint arXiv:1809.01941*.

Zhijing Jin, Qipeng Guo, Xipeng Qiu, and Zheng Zhang. 2020. Genwiki: A dataset of 1.3 million content-sharing text and graphs for unsupervised graph-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2398–2409.

Mayank Jobanputra. 2019. Unsupervised question answering for fact-checking. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 52–56, Hong Kong, China. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions Big Data*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of EACL 2017*, pages 427–431.

Armand Joulin and Tomas Mikolov. 2015. Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets. In *Advances in Neural Information Processing Systems*, pages 190–198.

Lucie-Aimée Kaffee, Hady Elsahar, Pavlos Vougiouklis, Christophe Gravier, Frédérique Laforest, Jonathon Hare, and Elena Simperl. 2018a. Learning to generate wikipedia summaries for underserved languages from wikidata. *arXiv preprint arXiv:1803.07116*.

Lucie-Aimée Kaffee, Hady Elsahar, Pavlos Vougiouklis, Christophe Gravier, Frédérique Laforest, Jonathon Hare, and Elena Simperl. 2018b. Mind the (language) gap: Generation of multilingual wikipedia summaries from wikidata for articleplaceholders. In *European Semantic Web Conference*, pages 319–334. Springer.

Lucie-Aimée Kaffee, Pavlos Vougiouklis, and Elena Simperl. 2020. Using natural language generation to bootstrap missing wikipedia articles: A human-centric perspective. *Semantic Web Journal*.

Tomoyuki Kajiwara and M Komachi. 2018. Text simplification without simplified corpora. *The Journal of Natural Language Processing*, 25:223–249.

Alireza Karduni. 2019. Human-misinformation interaction: Understanding the interdisciplinary approach needed to computationally combat false information. *arXiv preprint arXiv:1903.07136*.

Alireza Karduni, Isaac Cho, Ryan Wesslen, Sashank Santhanam, Svitlana Volkova, Dustin L Arendt, Samira Shaikh, and Wenwen Dou. 2019. Vulnerable to misinformation? verifi! In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 312–323.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *CoRR*, abs/2004.04906.

Akihiro Katsuta and Kazuhide Yamamoto. 2019. Improving text simplification by corpus expansion with unsupervised learning. In *2019 International Conference on Asian Language Processing (IALP)*, pages 216–221. IEEE.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of ACL 2013*, pages 1537–1546.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations*.

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.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task learning for rumour verification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3402–3413, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the wmt 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation*, pages 54–72.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739.

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2018. Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada. Association for Computational Linguistics.

Ádám Kovács, Evelin Ács, Judit Ács, Andras Kornai, and Gábor Recski. 2019. BME-UW at SRST-2019: Surface realization with interpreted regular tree grammars. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 35–40, Hong Kong, China. Association for Computational Linguistics.

Reno Kriz, Eleni Miltsakaki, Marianna Apidianaki, and Chris Callison-Burch. 2018. Simplification using paraphrases and context-based lexical substitution. In *Proceedings of NAACL 2018*, pages 207–217. Association for Computational Linguistics.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of ACL 2020*, pages 7918–7928, Online.

Sawan Kumar and Partha Talukdar. 2020. Nile: Natural language inference with faithful natural language explanations. *arXiv preprint arXiv:2005.12116*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. *arXiv preprint arXiv:1806.07297*.

Yuxuan Lai, Yansong Feng, Xiaohan Yu, Zheng Wang, Kun Xu, and Dongyan Zhao. 2019. Lattice cnns for matching based chinese question answering. *arXiv preprint arXiv:1902.09087*.

Kushal Lakhotia, Bhargavi Paranjape, Asish Ghoshal, Wen-tau Yih, Yashar Mehdad, and Srinivasan Iyer. 2020. Fid-ex: Improving sequence-to-sequence models for extractive rationale generation. *arXiv preprint arXiv:2012.15482*.

Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. 2020. Qed: A framework and dataset for explanations in question answering. *arXiv preprint arXiv:2009.06354*.

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2019. Large Memory Layers with Product Keys. In *Advances in Neural Information Processing Systems*, pages 8548–8559.

Gerasimos Lampouras and Andreas Vlachos. 2017. Sheffield at SemEval-2017 task 9: Transition-based language generation from AMR. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 586–591, Vancouver, Canada. Association for Computational Linguistics.

Isabelle Langrock and Sandra González-Bailón. 2020. The gender divide in wikipedia: A computational approach to assessing the impact of two feminist interventions. *Available at SSRN*.

Veronica Latcinnik and Jonathan Berant. 2020. Explaining question answering models through text generation. *arXiv preprint arXiv:2004.05569*.

Carolin Lawrence, Bhushan Kotnis, and Mathias Niepert. 2019. Attending to Future Tokens for Bidirectional Sequence Generation. In *EMNLP-IJCNLP*, pages 1–10, Hong Kong, China.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213.

Nayeon Lee, Andrea Madotto, and Pascale Fung. 2019. Exploring social bias in chatbots using stereotype knowledge. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 177–180.

Nayeon Lee, Chien-Sheng Wu, and Pascale Fung. 2018. Improving large-scale fact-checking using decomposable attention models and lexical tagging. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1133–1138, Brussels, Belgium. Association for Computational Linguistics.

Mike Lewis and Angela Fan. 2018. Generative question answering: Learning to answer the whole question.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. 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.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Margaret Li, Jason Weston, and Stephen Roller. 2019. ACUTE-EVAL: Improved Dialogue Evaluation with Optimized Questions and Multi-turn Comparisons. *arXiv preprint arXiv:1909.03087*.

Sizhen Li, Shuai Zhao, Bo Cheng, and Hao Yang. 2018. An end-to-end multi-task learning model for fact checking. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 138–144, Brussels, Belgium. Association for Computational Linguistics.

Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to Select Knowledge for Response Generation in Dialog Systems. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5081–5087. AAAI Press.

Chloe Lim. 2018. Checking how fact-checkers check. *Research & Politics*, 5(3):2053168018786848.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2018a. Toward abstractive summarization using semantic representations. *arXiv preprint arXiv:1805.10399*.

Haochen Liu, Wentao Wang, Yiqi Wang, Hui Liu, Zitao Liu, and Jiliang Tang. 2020a. Mitigating gender bias for neural dialogue generation with adversarial learning. *arXiv preprint arXiv:2009.13028*.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018b. Generating wikipedia by summarizing long sequences. In *ICLR*.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2019a. Unsupervised paraphrasing by simulated annealing. *arXiv preprint arXiv:1909.03588*.

Xiaojiang Liu, Zaiqing Nie, Nenghai Yu, and Ji-Rong Wen. 2010. Biosnowball: automated population of wikis. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 969–978.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020b. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Wei Luo, Julia Adams, and Hannah Brueckner. 2018. The ladies vanish?: American sociology and the genealogy of its missing women on wikipedia. *Comparative Sociology*, 17(5):519–556.

Emma Lurie. 2019. The challenges of algorithmically assigning fact-checks: A sociotechnical examination of google's reviewed claims.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. 2018. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 1–9.

Morgan Marietta, David C Barker, and Todd Bowser. 2015. Fact-checking polarized politics: Does the fact-check industry provide consistent guidance on disputed realities? In *The Forum*, volume 13, pages 577–596.

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*, pages 29–38.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. CamemBERT: a Tasty French Language Model. In *Proceedings of ACL 2020*.

Louis Martin, Benoît Sagot, Éric de la Clergerie, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of LREC 2020*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in Translation: Contextualized Word Vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of LREC 2018*.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (sr'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12. Association for Computational Linguistics.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, and Leo Wanner. 2019. The Second Multilingual Surface Realisation Shared Task (SR'19): Overview and Evaluation Results. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR), 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Hong Kong, China.

Simon Mille, Roberto Carlini, Alicia Burga, and Leo Wanner. 2017. FORGe at SemEval-2017 task 9: Deep sentence generation based on a sequence of graph transducers. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 920–923, Vancouver, Canada. Association for Computational Linguistics.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. parl.ai: A Dialog Research Software Platform. pages 79–84.

Sebastião Miranda, David Nogueira, Afonso Mendes, Andreas Vlachos, Andrew Secker, Rebecca Garrett, Jeff Mitchel, and Zita Marinho. 2019. Automated fact checking in the news room. In *The World Wide Web Conference*, pages 3579–3583.

Bhaskar Mitra and Nick Craswell. 2017. Neural models for information retrieval. *arXiv preprint arXiv:1705.01509*.

Andriy Mnih and Geoffrey Hinton. 2009. A Scalable Hierarchical Distributed Language Model. In *Advances in Neural Information Processing Systems*, pages 1081–1088.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M Khapra. 2018. Towards Exploiting Background Knowledge for Building Conversation Systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *arXiv preprint arXiv:2004.14546*.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 104–111. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL 2019 (Demonstrations)*, pages 48–53, Minneapolis, Minnesota.

Gustavo Paetzold and Lucia Specia. 2016a. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569.

Gustavo H Paetzold and Lucia Specia. 2016b. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*.

Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. *arXiv preprint arXiv:2004.14373*.

Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing gender bias in abusive language detection. *arXiv preprint arXiv:1808.07231*.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple ppdb: A paraphrase database for simplification. In *Proceedings of ACL 2016*, pages 143–148.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of ACL-CoLing 2015*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Laura Perez-Beltrachini and Claire Gardent. 2017. Analysing data-to-text generation benchmarks. *arXiv preprint arXiv:1705.03802*.

Stan Peshterliev, Barlas Oguz, Debojeet Chatterjee, Hakan Inan, and Vikas Bhardwaj. 2021. Conversational answer generation and factuality for reading comprehension question-answering. *arXiv preprint arXiv:2103.06500*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? 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 2463–2473.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Tobias Plötz and Stefan Roth. 2018. Neural Nearest Neighbors Networks. In *Advances in Neural Information Processing Systems*, pages 1087–1098.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Brussels, Belgium. Association for Computational Linguistics.

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, Belgium, Brussels. Association for Computational Linguistics.

Nima Pourdamghani, Kevin Knight, and Ulf Hermjakob. 2016. Generating English from abstract meaning representations. In *Proceedings of the 9th International Natural Language Generation conference*, pages 21–25, Edinburgh, UK. Association for Computational Linguistics.

Michael Pust, Ulf Hermjakob, Kevin Knight, Daniel Marcu, and Jonathan May. 2015. Parsing english into abstract meaning representation using syntax-based machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1143–1154.

Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, Bill Dolan, Yejin Choi, and Jianfeng Gao. 2019. Conversing by Reading: Contentful Neural Conversation with On-demand Machine Reading. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5427–5436.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

Jack Rae, Jonathan J. Hunt, Ivo Danihelka, Timothy Harley, Andrew W Senior, Gregory Wayne, Alex Graves, and Timothy Lillicrap. 2016. Scaling Memory-Augmented Neural Networks with Sparse Reads and Writes. In *Advances in Neural Information Processing Systems*, pages 3621–3629.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2737–2746, Melbourne, Australia. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2018. Coqa: A conversational question answering challenge. *arXiv preprint arXiv:1808.07042*.

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*, page 15. ACM.

Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing AMR-to-text generation with dual graph representations. 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 3181–3192, Hong Kong, China. Association for Computational Linguistics.

Benjamin Riedel, Isabelle Augenstein, Georgios P Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *arXiv preprint arXiv:1707.03264*.

Kevin Roitero, Michael Soprano, Shaoyang Fan, Damiano Spina, Stefano Mizzaro, and Gianluca Demartini. 2020. Can the crowd identify misinformation objectively? the effects of judgment scale and assessor's background. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Marco Rospocher, Marieke van Erp, Piek Vossen, Antske Fokkens, Itziar Aldabe, German Rigau, Aitor Soroa, Thomas Ploeger, and Tessel Bogaard. 2016. Building event-centric knowledge graphs from news. *Journal of Web Semantics*, 37:132–151.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2019. Leveraging pre-trained checkpoints for sequence generation tasks. *ArXiv*, abs/1907.12461.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Alexander M Rush. 2018. The annotated transformer. In *Proceedings of workshop for NLP open source software (NLP-OSS)*, pages 52–60.

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 Transactions on Accessible Computing*, 6(4):1–36.

Christina Joan Sauper and Regina Barzilay. 2009. Automatically generating wikipedia articles: A structure-aware approach. Association for Computational Linguistics.

Katja Geertruida Schmahl, Tom Julian Viering, Stavros Makrodimitris, Arman Naseri Jahfari, David Tax, and Marco Loog. 2020. Is wikipedia succeeding in reducing gender bias? assessing changes in gender bias in wikipedia using word embeddings. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 94–103.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv:1911.04944*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441.

Iulian V. Serban, Ryan Lowe, Laurent Charlin, and Joelle Pineau. 2016a. Generative deep neural networks for dialogue: A short review. *arXiv preprint arXiv:1611.06216*.

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016b. Building End-to-End Dialogue Systems using Generative Hierarchical Neural Network Models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2018. Order-planning neural text generation from structured data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Anastasia Shimorina and Claire Gardent. 2019a. LORIA / lorraine university at multilingual surface realisation 2019. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 88–93, Hong Kong, China. Association for Computational Linguistics.

Anastasia Shimorina and Claire Gardent. 2019b. Surface realisation using full delexicalisation. 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 3086–3096, Hong Kong, China. Association for Computational Linguistics.

Kurt Shuster, Samuel Humeau, Antoine Bordes, and Jason Weston. 2020. Image-chat: Engaging Grounded Conversations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2414–2429.

Kurt Shuster, Samuel Humeau, Hexiang Hu, Antoine Bordes, and Jason Weston. 2019. Engaging Image captioning via Personality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12516–12526.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Marco Antonio Sobrevilla Cabezudo, Simon Mille, and Thiago Pardo. 2019. Back-translation as strategy to tackle the lack of corpus in natural language generation from semantic representations. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 94–103, Hong Kong, China. Association for Computational Linguistics.

Haoyu Song, Yan Wang, Wei-Nan Zhang, Xiaojiang Liu, and Ting Liu. 2020. Generate, Delete and Rewrite: A Three-Stage Framework for Improving Persona Consistency of Dialogue Generation. *arXiv preprint arXiv:2004.07672*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *ICML*, volume 97, pages 5926–5936.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. Two are better than one: An ensemble of retrieval-and generation-based dialog systems. *arXiv preprint arXiv:1610.07149*.

Sanja Štajner, Iacer Calixto, and Horacio Saggion. 2015. Automatic text simplification for Spanish: Comparative evaluation of various simplification strategies. In *Proceedings of the international conference recent advances in natural language processing*.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. 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 885–895.

Gabriel Stanovsky, Noah A Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. *arXiv preprint arXiv:1906.00591*.

Despina Stratigakos. 2016. Unforgetting women architects: From the pritzker to wikipedia. *Places Journal*.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019a. Adaptive attention span in transformers.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. 2019b. Augmenting Self-attention with Persistent Memory.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-End Memory Networks. In *Advances in neural information processing systems*, pages 2440–2448.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. *arXiv preprint arXiv:1809.00782*.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of ACL 2018*, pages 2058–2068.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73.

Craig Thomson and Ehud Reiter. 2020. A gold standard methodology for evaluating accuracy in data-to-text systems. *arXiv preprint arXiv:2011.03992*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of LREC 2012*.

Sara Tonelli, Alessio Palmero Aprosio, and Francesca Saltori. 2016. SIMPITIKI: a Simplification corpus for Italian. *Proceedings of the Third Italian Conference on Computational Linguistics*.

Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. 2019. Learning to speak and act in a fantasy text adventure game. 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 673–683.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA. Association for Computational Linguistics.

Piek Vossen, Tommaso Caselli, and Yiota Kontzopoulou. 2015. Storylines for structuring massive streams of news. In *Proceedings of the First Workshop on Computing News Storylines*, pages 40–49.

William Yang Wang. 2017. “liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020. Towards faithful neural table-to-text generation with content-matching constraints. *arXiv preprint arXiv:2005.00969*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzman, Armand Joulin, and Edouard Grave. 2019. CCnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory Networks. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and Refine: Improved Sequence Generation Models For Dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of ACL 2018*, pages 451–462, Melbourne, Australia.

Rachel Wities, Vered Shwartz, Gabriel Stanovsky, Meni Adler, Ori Shapira, Shyam Upadhyay, Dan Roth, Eugenio Martínez-Cámara, Iryna Gurevych, and Ido Dagan. 2017. A consolidated open knowledge representation for multiple texts. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 12–24.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of EMNLP 2011*, pages 409–420, Edinburgh, Scotland, UK.

Zena Worku, Taryn Bipat, David W McDonald, and Mark Zachry. 2020. Exploring systematic bias through article deletions on wikipedia from a behavioral perspective. In *Proceedings of the 16th International Symposium on Open Collaboration*, pages 1–22.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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 833–844, Hong Kong, China. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of ACL 2012*, pages 1015–1024.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

Kun Xu, Lingfei Wu, Zhiguo Wang, Yansong Feng, Michael Witbrock, and Vadim Sheinin. 2018. Graph2seq: Graph to sequence learning with attention-based neural networks. *arXiv preprint arXiv:1804.00823*.

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*, pages 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.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. *arXiv preprint arXiv:1706.06681*.

Catherine Yeo and Alyssa Chen. 2020. Defining and evaluating fair natural language generation. *arXiv preprint arXiv:2008.01548*.

Amber G Young, Ariel D Wigdor, and Gerald C Kane. 2020. The gender bias tug-of-war in a co-creation community: Core-periphery tension on wikipedia. *Journal of Management Information Systems*, 37(4):1047–1072.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2020. A survey of knowledge-enhanced text generation. *arXiv preprint arXiv:2010.04389*.

Xiang Yu, Agnieszka Falenska, Marina Haid, Ngoc Thang Vu, and Jonas Kuhn. 2019. IMSurReal: IMS at the surface realization shared task 2019. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 50–58, Hong Kong, China. Association for Computational Linguistics.

Amy X Zhang, Aditya Ranganathan, Sarah Emlen Metz, Scott Appling, Connie Moon Sehat, Norman Gilmore, Nick B Adams, Emmanuel Vincent, Jennifer Lee, Martin Robbins, et al. 2018a. A structured response to misinformation: Defining and annotating credibility indicators in news articles. In *Companion Proceedings of the The Web Conference 2018*, pages 603–612.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018b. Personalizing Dialogue Agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of EMNLP 2017*, pages 584–594, Copenhagen, Denmark.

Chao Zhao, Marilyn Walker, and Snigdha Chaturvedi. 2020a. Bridging the structural gap between encoding and decoding for data-to-text generation. ACL.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018a. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018b. Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853, Brussels, Belgium. Association for Computational Linguistics.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018c. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of EMNLP 2018*, pages 3164–3173, Brussels, Belgium.

Yanbin Zhao, Lu Chen, Zhi Chen, and Kai Yu. 2020b. Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018d. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.

Zhiping Zheng. 2003. Question answering using web news as knowledge base. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, pages 251–254. Association for Computational Linguistics.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. 2020. Detecting hallucinated content in conditional neural sequence generation. *arXiv preprint arXiv:2011.02593*.

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*.

Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling graph structure in transformer for better AMR-to-text generation. 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 5459–5468, Hong Kong, China. Association for Computational Linguistics.

Yutao Zhu, Zhicheng Dou, Jian-Yun Nie, and Ji-Rong Wen. 2020. ReBoost: a retrieval-boosted sequence-to-sequence model for neural response generation. *Information Retrieval Journal*, 23(1):27–48.

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.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.