UNIVERSITÉ DE LORRAINE | BIBLIOTHÈQUES UNIVERSITAIRES

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 bibliothèque : ddoc-theses-contact@univ-lorraine.fr
(Cette adresse ne permet pas de contacter les auteurs)

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

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Generating and answering questions across text and knowledge graphs

THÈSE

présentée et soutenue publiquement le 2 decembre 2024

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Kelvin Han

Composition du jury

Président :	Anne Vilnat	Professeure émérite, Université Paris-Saclay, France
Rapporteurs :	Anne Vilnat	
	Frédéric Bechet	Professeur, Université Aix-Marseille, France
Examinateurs :	Sophie Rosset	Directrice de recherche, CNRS, LISN, France
	Catherine Faron	Professeure, Université Côte d'Azur, France
Directrice de thèse :	Claire Gardent	Directrice de recherche, CNRS, LORIA, France
Co-encadrent de thèse :	Thiago Castro Ferreira	Chercheur, Federal University of Minas Gerais, Brésil

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

Mis en page avec la classe thesul.

Remerciements / Acknowledgements

This thesis was supported by Project QUANTUM (Project-ANR-19-CE23-0025), which was funded by the Agence Nationale de la Recherche (the French National Research Agency, ANR).

I would like to sincerely thank my supervisors, Claire for giving me an opportunity and patiently helping me to grow in research, and Thiago for guiding my start in research. I would also like to thank the members of my committee for very generously giving their time to be reviewers and examiners of my thesis. Thank you also to every one of the previous and current members (permanent staff, PhDs, engineers and interns) of the Synalp team whom I have had the honour to work alongside during my time in LORIA. A special thanks to every of my office mates in B112 (as well as a shout-out to Estelle, who very kindly helped read the French translations of sections of this thesis). Thank you also to the Project QUANTUM consortium members for your support. A special note for Prerak and Siyana with whom I have had the great pleasure of sharing a start in the NLP masters at IDMC and embarking on similar PhD journeys at Loria. Finally, I deeply thank my family and friends back home for your patience these past years while I have been away.

To my family, for your patience.
(À ma famille, à votre patience.)

"I keep six honest serving-men
(They taught me all I knew);
Their names are What and Why and When
And How and Where and Who.
I send them over land and sea,
I send them east and west;
But after they have worked for me,
I give them all a rest.

I let them rest from nine till five,
For I am busy then,
As well as breakfast, lunch, and tea,
For they are hungry men.
But different folk have different views;
I know a person small—
She keeps ten million serving-men,
Who get no rest at all!

She sends'em abroad on her own affairs,
From the second she opens her eyes—
One million Hows, two million Wheres,
And seven million Whys!"

(Rudyard Kipling,
from The Elephant's Child,
in Just So Stories, 1902)

Abstract

Question generation (QG) is the task of automatically producing a question given some information source containing the answer. It is a subtask within natural language generation (NLG) but is also closely associated with question answering (QA), which is a counterpoint to QG. While QG is concerned with generating the linguistic expression for seeking information, the QA task is concerned with meeting that need by automatically identifying the answer to a question given some information source. Both tasks have direct applicability in domains such as information retrieval, dialogue and conversation, and education. Recent research also indicates that QG and QA, when used jointly in QA-based evaluation, are helpful for factual verification (especially for NLG outputs such as summarisation and data-to-text generations). When used together to produce a discourse representation, they can also help reduce the propensity of large language models (LLMs) to produce text with hallucinations and factual inaccuracies.

While QA has long been studied, and approaches have been proposed as early as the 1960s, QG only started to gain more research attention in recent years. Most research on the tasks is focused on addressing only one of them and doing so for a single modality. In QG, previous approaches typically rely on architectures that require heavy processing and do not generally consider the generation of questions across the entirety of the input information source nor the diversity of the ways a question can be phrased. In QA, although work has been done for answering questions given some unstructured input (e.g. a piece of text), and work has also been done for doing so given some structured input (e.g. knowledge graph (KG) or tables), these methods are typically not transferable for use on another input modality.

In this thesis, we are focused on QG foremost, with the aim of identifying ways to generate questions across both structured and unstructured information, namely text and KG inputs, in a manner that is controllable for increasing the diversity, comprehensiveness, and coverage of these questions. We also study QG and QA in concert with a model that can controllably generate both simple and complex questions from one modality and also answer them on another modality, an ability that has relevance for improving QA-based evaluation. Finally, we examine doing so for lower-resourced languages other than English, with the view that being able to do so helps enable similar QA-based evaluation for these languages.

Keywords: question generation, question answering, natural language generation, knowledge graphs, multilingual

Résumé

La génération de questions (QG) est une tâche qui consiste à produire automatiquement une question à partir d'une source d'information en entrée contenant la réponse. Il s'agit d'une sous-tâche de la génération automatique de textes (NLG), elle est également liée à la tâche de questions-réponses (QA), qui est l'opposé de la QG. L'objectif de la QG est de générer une expression linguistique pour rechercher l'information, l'objectif du QA est d'identifier automatiquement la réponse à une question à partir d'une source d'information en entrée. Les deux tâches ont des applications dans des domaines tels que la recherche d'information, les dialogues et les conversations, et aussi dans l'éducation. Lorsque les tâches de QG et de QA sont tout deux utilisées pour évaluation de textes basées sur la QA, elles sont aussi utilisées pour la vérification des faits (notamment les sorties de la NLG qui peuvent être sur le résumé ou la génération de texte à partir des données). La plupart des recherches sur ces deux tâches se concentrent soit sur l'une soit sur l'autre, et généralement dans une seule et unique modalité. Dans le domaine de la QG, les approches antérieures reposaient sur des architectures nécessitant un prétraitement intensif. Les questions ainsi générées ne couvraient ni l'entièreté des informations en entrée, ni la diversité des nuances possibles. Dans le domaine des QA, bien que des approches aient été proposées pour répondre aux question à partir d'informations non structurées (par exemple, un document textuel brute), mais aussi structurées (par exemple, des graphes de connaissances (KG) ou des tableaux), ces méthodes ne sont pas transférables pour une autre modalité. Dans cette thèse, nous nous concentrons d'abord sur la QG, afin d'identifier les moyens de générer des questions à partir d'informations structurées et également non structurées, et de le faire de manière contrôlée pour augmenter la diversité et la couverture des questions générées. Ensuite, nous étudierons également la conduite de la QG et des QA par un modèle capable de générer des questions simples et complexes de manière contrôlée à partir d'une modalité, puis répondre sur une autre modalité. Enfin, nous examinerons la possibilité de faire la même tâche pour les langues avec peu de ressources autres que l'anglais, ce qui pourrait faciliter l'évaluation basée sur les QA pour ces langues.

Mots-clés: génération des questions, réponses aux questions, génération automatique de textes, graphes de connaissances, multilingues

Traduction des termes

génération automatique de textes	*natural language generation (NLG)*
génération de questions	*question generation (QG)*
génération de questions a partir de graphes de connaissances	*knowledge graph question generation (KQGG)*
grands modèles de langues	*large language models (LLMs)*
graphes de connaissances	*knowledge graphs (KG)*
questions-réponses a partir de graphes de connaissances	*knowledge graph question-answering (KQGA)*
questions-réponses/réponses aux questions	*question-answering (QA)*
traitement automatique des langues	*natural language processing (NLP)*

Contents

Génération et réponse à des questions à partir des textes et des graphes de connaissances **xi**

1 Questions de recherche et contributions . . . xii

2 Vue d'ensemble de la thèse . . . xiii

2.1 Liste des publications . . . xiv

List of tables **xix**

Chapter 1 Introduction **1**

1.1 Research questions and contributions . . . 2

1.2 Thesis outline . . . 3

1.2.1 List of publications . . . 4

Chapter 2 Background **5**

2.1 Definitions . . . 6

2.1.1 Forms of questions . . . 6

2.1.2 Simple and complex questions . . . 6

2.1.3 Terminology and notation . . . 8

2.2 Natural language generation with autoregressive language models . . . 8

2.3 Models for generating and answering questions . . . 10

2.3.1 Generating questions from text . . . 11

2.3.2 Answering questions from text . . . 12

2.3.3 Generating questions from KG . . . 14

2.3.4 Answering questions from KG . . . 16

2.4 Knowledge graphs . . . 18

2.5 QG/QA-based evaluation of generated text . . . 19

2.6 Bridging modalities: QG/QA across Text and KG . . . 20

2.7 Multilingual QG/QA . . . 22

2.8 Data . . . 23

2.8.1 QG/QA data for KG . . . 23
2.8.2 QG/QA data aligned across modalities . . . 24
2.8.3 Aligned QG/QA data outside of English . . . 24
2.9 Evaluation . . . 25
2.9.1 Evaluation for QA . . . 25
2.9.2 Evaluation for QG . . . 25

Chapter 3 Generating Questions from Wikidata Triples **31**
3.1 Introduction . . . 32
3.1.1 Task and Terminology . . . 33
3.2 The WikiDataQG Dataset . . . 34
3.2.1 (RDF, Question) Datasets . . . 34
3.2.2 Adding typing and lexicalisation information . . . 36
3.3 Approach . . . 37
3.4 Experiments . . . 38
3.4.1 Training Details . . . 38
3.4.2 Elsahar's Model. . . . 38
3.4.3 Automatic Evaluation . . . 38
3.4.4 Human Evaluation . . . 39
3.5 Results and Discussion . . . 39
3.5.1 With and without additional NL information on Seen data . . . 39
3.5.2 Zero-Shot Learning . . . 40
3.5.3 Additional data . . . 40
3.5.4 Downstream QA Evaluation . . . 43
3.5.5 Ablation . . . 44
3.6 Conclusion . . . 44

Chapter 4 Generating and Answering Simple and Complex Questions from Text and from KGs **47**
4.1 Introduction . . . 48
4.2 Method and data . . . 49
4.2.1 Associating Graphs and Texts with Questions and Answers (Step 1) . . . 49
4.2.2 Training QG Models on Q-KELM (Step 2) . . . 50
4.2.3 Extending Q-WebNLG^0 (Step 3) . . . 52
4.3 QTT, a multimodal QG-QA Model . . . 52

4.4 Experiments . . . 55
4.4.1 Evaluation data . . . 55
4.4.2 Evaluating QG Coverage . . . 55
4.4.3 Evaluating QA Consistency . . . 55
4.4.4 Downstream: QA with FiD . . . 57
4.4.5 Downstream: Data-QuestEval metric . . . 57
4.4.6 Evaluation settings . . . 57
4.5 Results . . . 58
4.5.1 QG Coverage . . . 58
4.5.2 QA Consistency . . . 58
4.5.3 Downstream Evaluations . . . 60
4.5.4 Ablation . . . 60
4.5.5 Human evaluation . . . 62
4.5.6 Question generation examples . . . 63
4.6 Conclusion . . . 64

Chapter 5 Multilingual Generation and Answering of Questions from Texts and Knowledge Graphs **65**
5.1 Introduction . . . 66
5.1.1 Terminology and notations . . . 67
5.2 Approach . . . 67
5.3 Data . . . 68
5.4 Creating QA/QG Data for English . . . 68
5.5 Creating QA/QG Data for Russian and Portuguese . . . 70
5.6 Multimodal multi-task QG-QA model . . . 70
5.7 Evaluation and results . . . 72
5.7.1 Baseline: single-task models . . . 72
5.7.2 Evaluating Question Coverage . . . 72
5.7.3 Evaluating QA Consistency . . . 72
5.7.4 Data-QuestEval metric . . . 76
5.7.5 Multilingual Retrieval-based QA . . . 79
5.8 Conclusion . . . 79

Conclusion **81**
9 Summary of findings . . . 81

10 Future directions 82

Annexes **85**

Annex A Appendices for Chapter 3 **85**

A.1 Mapping the WebNLG DBpedia triples to Wikidata 86

A.2 Combining the three datasets 87

A.3 Examples: models' inputs and generation outputs 88

Annex B Appendices for Chapter 4 **89**

B.1 Detailed results 90

B.2 Implementation details: CQ-Gen and SQ-Gen 90

B.2.1 Training data 90

B.2.2 Technical details 92

B.3 Implementation details: QTT 92

B.3.1 Training data 92

B.3.2 Technical details 94

Annex C Appendices for Chapter 5 **99**

C.1 Data details 100

C.1.1 Synthetic QA data in English 100

C.1.2 Synthetic QA data in Portuguese and Russian 100

C.2 Training and evaluation details 100

C.2.1 Negative sampling for QA 100

C.2.2 Upsampling 101

C.2.3 Evaluation settings 101

C.3 Detailed results 103

C.3.1 QA consistency: Token F1 and Exact Match 103

C.3.2 Finer-grained QA consistency tests 104

C.3.3 Multilingual Retrieval-based QA 105

Bibliography **107**

Génération et réponse à des questions à partir des textes et des graphes de connaissances

La génération de questions (QG) est une tâche qui s'inscrit dans le domaine de la génération automatique de textes (NLG), qui concerne la génération automatique de textes ayant les caractéristiques d'un texte écrit par un humain. La tâche de QG est l'opposé de celle de questions-réponses (QA), qui est une tâche bien établie dans le domaine du traitement automatique des langues (NLP). La tâche de QA consiste à trouver la réponse correcte à une question à partir d'une source d'information, alors que la QG consiste à produire la question à partir d'une source d'information (généralement accompagnée de la réponse recherchée pour la question). La source d'information fournie dans le cadre de la QG ou des QA peut se présenter dans une forme non structurée, telle qu'un texte ou une image, ou dans une forme structurée, telle qu'un sous-graphe d'un graphe de connaissances (KG), un tableau, ou même une combinaison d'entrées de différentes formes. Les QA sont utilisés, par exemple, dans la recherche d'informations (Kolomiyets and Moens, 2011), dans le dialogue et la conversation (Reddy et al., 2019), ainsi que dans l'éducation (Agarwal et al., 2019). La QG est également utile dans les mêmes domaines que les QA, mais elle peut aussi être utilisée pour générer de nouvelles paires de QA afin d'augmenter les données pour l'entraînement des modèles de QA. Lorsque la QG et les QA sont utilisées ensemble, elles sont également utiles pour vérifier le contenu sémantique (Wang et al., 2020; Scialom et al., 2021; Rebuffel et al., 2021), ainsi que pour représenter la structure discursive (Wu et al., 2023) d'un texte. Dans la NLG, les séquences ordonnées de paires de QA en tant que forme de plan de discours se sont aussi révélées (Narayan et al., 2023; Huot et al., 2023) utiles pour réduire l'occurrence des hallucinations et d'incohérences factuelles lorsqu'elles sont fournies comme entrées aux grands modèles de langues ($LLMs$) afin de conditionner leurs générations. Alors que la tâche de QA a fait l'objet d'une attention considérable de la part des chercheurs, avec diverses méthodes et ressources proposées depuis les années 1960, la QG, quant à elle, a fait l'objet de beaucoup moins d'attention. Dans cette thèse, nous nous concentrerons principalement sur la QG, en particulier sur la génération de questions factuelles vers le texte et les KG.

Il existe différents types de questions, en ce sens qu'elles (i) répondent à une variété de besoins en matière de recherche d'informations et (ii) prennent des formes différentes. Les formes prises par les questions ont une correspondance générale avec leur objectif, par exemple les mots interrogatifs (qui, quoi, lequel, quand, où, pourquoi et comment en français) reflètent le type sémantique de l'information recherchée (i.e. la réponse). En raison de la caractéristique des questions et des relations qu'elles entretiennent avec leurs contextes et leurs réponses, la génération d'une question bien formulée nécessite de s'assurer de la qualité de leur forme (fluidité) autant qu'à celle de leur contenu (pertinence par rapport à la réponse ; cohérence par rapport

au contexte). Parmi les types de questions factuelles sur lesquelles nous nous concentrerons dans cette thèse, nous pouvons trouver : la vérification de connaissances (par des questions fermées), la demande d'informations spécifiques ("Quelle est la longueur de la vallée du Val de Loire ?") ou l'obtention de plus d'informations sur une entité ou un concept (comme "Quel est le lieu de naissance du designer Louis Majorelle? ").

1 Questions de recherche et contributions

Un fil conducteur de cette thèse est l'effort fourni pour contrôler l'augmentation de la diversité et la couverture des questions qui peuvent être générées automatiquement à partir d'une entrée donnée. Dans le but que ces avancés facilitent les avancées dans les applications en aval, telles que l'augmentation des données et les méthodes basées sur la QG/QA pour l'évaluation du contenu sémantique d'un texte. Un autre fil conducteur à cette thèse est l'objectif d'exploiter la QG et les QA conjointement et de façon intermodale, et pour que les langues autres que l'anglais, puissent permettre la vérification du contenu sémantique d'un texte généré automatiquement. L'objectif de cette thèse est d'étudier et de proposer des méthodes et des modèles pour répondre aux questions de recherche suivantes :

1. 1. Quels avantages les modèles de langues pré-entraînés peuvent-ils apporter à la tâche de génération de questions à partir d'un triplet de *KG* ? Comment pourrait-on en bénéficier pour les *QG* qui sont des questions typiques avec un focus sur sa contrôlabilité ?

Pour cette question de recherche, nous nous sommes intéressés à l'amélioration de la génération de questions simples (Section 2.1) à partir d'une entrée de KG. Il s'agit ici de générer une question à partir d'un seul fait de KG de la forme $\langle$ SUJET , RELATION , OBJET $\rangle$. Bien que cette tâche ait été explorée et que des méthodes aient été proposées à cet effet, nous avons souhaité la réexaminer pour deux raisons principales. Tout d'abord, la capacité à lexicaliser correctement les entités et les relations d'un KG qui n'ont pas été vues au moment de l'apprentissage. Le succès initial des modèles génératifs pré-entraînés tels que BART (Lewis et al., 2020a) et T5 (Raffel et al., 2020) - avec leur connaissance paramétrique de la structure des langues ainsi que des entités et des concepts glanés dans le texte sur lequel ils ont été entraînés - pour des tâches telles que la traduction automatique et le résumé - suggère qu'ils pourraient être adaptés pour relever ce défi. Deuxièmement, étant donné que l'ensemble des données pour l'entraînement de modèles $KGQG$ simples ne contient qu'une seule question pour un triplet KG donné, et que ces questions ne recherchent que l'objet du fait KG, les travaux antérieurs se sont concentrés sur la génération de questions d'un tel type et d'une telle forme. Toutefois, les questions de recherche d'informations peuvent être formulées de plusieurs manières et différentes parties du triplet fait KG (le sujet ou l'objet) peuvent également faire l'objet d'une interrogation.

2. Comment pouvons-nous améliorer la compréhension et la couverture des systèmes de *QG*, et cela peut-il se faire également pour les questions complexes, ainsi qu'à travers les modalités de la *QG* et des *QA* ?

Nous nous sommes intéressés ici à la manière dont l'amélioration de ces aspects (complexité, compréhension et couverture, et des intermodalités entre la QG et les QA) pourrait contribuer à renforcer les évaluations basées sur la QG/QA du contenu sémantique des sorties de la génération de données en texte (Rebuffel et al., 2021). Le fait de pouvoir générer et répondre à des questions complexes (en plus des questions simples) sur un texte peut renforcer la confiance de

ces évaluations basées sur les *QA*. En effet, bien que générer et répondre à un ensemble complet de questions simples nous permettent de vérifier les faits atomiques d'un texte, un texte fluide est plus qu'un ensemble de faits atomiques, et les questions simples seules peuvent ne pas fournir un signe complet sur la qualité sémantique du texte. Par exemple, supposons que nous ayons le texte suivant :

> La personne X est décédé d'un mésothéliome, il était réalisateur du film documentaire « Invisible Threat ». C'est en relation avec l'exposition à l'amiante. »

qui est généré pour les éléments suivants des faits *KG* :

> { ⟨ Personne X , cause du décès , mésothéliome ⟩, ⟨ Personne X , réalisateur , « Invisible Threat » ⟩, ⟨ « Invisible Threat » , genre , documentaire ⟩, ⟨ mésothéliome , cause , exposition à l'amiante ⟩ }.

Un ensemble de questions simples pour le texte pourrait être :

> { « De quoi la personne X est-elle décédée ?», « Qui est le réalisateur de « Invisible Threat » ? », « Quel est le genre de « Invisible Threat » ? », « Qu'est-ce qui cause le mésothéliome ? » }

L'ambiguïté du texte, qui est généralement jugée indésirable, pourrait amener un modèle de *QG* à remplacer la dernière question par « Quel est le principal sujet du film « Invisible Threat » ? », ou un modèle de *QA* pourrait déterminer qu'il n'est pas possible de répondre à la dernière question. En revanche, la possibilité de générer des questions complexes telles que « Quelle est la cause du décès de la personne X, qui est liée à l'exposition à l'amiante ? » et d'y répondre, à la fois sur le texte et sur l'ensemble des faits *KG*, fournit un signe sur la qualité du texte avec un niveau de confiance plus élevé qu'avec le seul ensemble de questions simples.

3. Comment la *QG* et les *QA* multimodales peuvent-elles être étendues à d'autres langues moins bien dotées en ressources, et la *QG* et l'*QA* multilingues entre modalités peuvent-elles être réalisées ?

Pour cette question de recherche, nous voulions identifier des moyens d'étendre la *QG* et les *QA* multimodales à des langues autres que l'anglais, où les données et les ressources disponibles pour la formation des modèles de *QG* et des *QA* dans ces langues sont beaucoup plus limitées qu'en anglais. Bien que des recherches aient été menées sur les *QA* multilingues et que des études initiales aient été réalisées sur la *QG* multilingue, ces travaux se sont concentrés uniquement sur la *QG* ou sur les *QA*; à notre connaissance, aucun travail n'a été porté sur une combinaison de tous ces éléments. Cela est dû en partie à la disponibilité limitée des données pour l'apprentissage et l'évaluation des *QA* et de la *QG* dans ces langues, et cela est encore davantage compliqué par le fait que pour les *KG* publics - tels que Wikidata que nous utilisons dans notre travail, la majorité des étiquettes pour les entités sont en anglais. Cela nécessite que les *QG* et *QA* intermodales soient également multilingues, par exemple les questions générées à partir d'un texte russe sont en russe, mais elles sont répondues par rapport à un graphique *KG* avec des étiquettes anglaises et les réponses sont comparées entre les deux langues.

2 Vue d'ensemble de la thèse

Cette thèse est organisée de la manière suivante. Tout d'abord, le chapitre 2 fournit une vue d'ensemble des notions clés, des tâches, des données et des modèles que nous utiliserons.

Ensuite, dans le chapitre 3, nous détaillerons le travail effectué pour répondre à la première

question de recherche sur l'exploitation du paradigme du pré-entraînement et de l'affinage des modèles de langues pour générer des questions simples à partir de triplets de *KG* en utilisant Wikidata comme la source de *KG*. Pour ce faire, nous avons collecté un ensemble de questions sur des faits de *KG* uniques et plus diversifiés (couvrant les sujets ou les objets des faits de *KG* et avec différents types de questions), que nous avons ensuite consolidé avec d'autres ensembles de données existants. Nous montrons que l'affinage d'un modèle pré-entraîné tel que BART, avec des questions de type contrôle et notre ensemble de données *KGQG* simple et élargi, permet de générer des questions variées en anglais et que l'utilisation de notre modèle pour l'augmentation des données pour l'entraînement d'un modèle de *QA* aide à inverser la chute des performances des modèle lorsque la distribution des types de questions de l'ensemble de test change.

Dans le chapitre 4, nous allons au-delà du simple *KGQG* en abordant un d'autres aspects (la *QG* et des *QA* sur/à partir des modalités texte et *KG* et des questions complexes) et répondre à notre deuxième question de recherche. Notre travail a consisté à créer un ensemble de données synthétiques pour entraîner un modèle capable de générer et de répondre à des questions simples et complexes dans un texte et avec un *KG* en anglais. Nous montrons que (i) le modèle génère une couverture plus large de questions sur le même élément d'information dans chaque modalité, (ii) il a une meilleure cohérence de *QA* par rapport à une base utilisée dans une métrique sans référence qui s'est déjà avérée utile pour évaluer la génération de texte à partir de données, et (iii) le remplacement des modèles de base par notre modèle obtenir de meilleures corrélations de la métrique avec les jugements humains sur les résultats de la génération de texte à partir de données.

Dans le chapitre 5, nous développerons le travail des deux chapitres précédents pour adresser notre troisième question de recherche sur l'extension de la *QG* et des *QA* multimodales dans des contextes multilingues. Nous avons utilisé la traduction automatique pour obtenir des données synthétiques pour le russe et le portugais brésilien, qui sont moins bien dotés (en termes de données, d'outils et de modèles de *NLP*) que l'anglais. Nous avons ensuite exploité cet ensemble de données multilingues pour affiner un LM multilingue pré-entraîné et obtenir des modèles qui effectuent la *QG* et les *QA* de manière intermodale dans ces langues. Cela nous a permis de générer des questions en russe ou en portugais brésilien dans ces langues pour une modalité, puis d'y répondre de manière interlingue dans l'autre modalité. Cela a permis de généraliser une métrique sans référence de l'anglais à ces autres langues. Nous avons ensuite démontré que l'utilisation de notre modèle permet d'améliorer la métrique à l'aide de jugements humains pour les résultats dans ces langues.

Enfin, nous conclurons et récapitulerons nos résultats et nos contributions pour chaque question de recherche et nous indiquerons les orientations potentielles.

2.1 Liste des publications

Les publications dans la liste suivante constituent les principaux éléments de cette thèse :

- Kelvin Han, Thiago Castro Ferreira and Claire Gardent. Generating Questions from Wikidata Triples. In Proceedings of the Thirteenth Language Resources and Evaluation Conference. June 2022, Marseille, France. European Language Resources Association.

- Kelvin Han and Claire Gardent. Generating and Answering Simple and Complex Questions from Text and from Knowledge Graphs. In Proceedings of The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter

of the Association for Computational Linguistics. November 2023, Bali, Indonesia, Indonesia. Association for Computational Linguistics.

- Kelvin Han and Claire Gardent. Multilingual Generation and Answering of Questions from Texts and Knowledge Graphs. In Findings of the Association For Computational Linguistics: EM*NLP* 2023. December 2023, Singapore, Singapore. Association for Computational Linguistics.

Table of figures

2.1 Graphical overview of the components in the FALCON (Harabagiu et al., 2000) rules-based pipeline QA system. . . . 13
2.2 Overview of the BERT (Devlin et al., 2019) model's pretrain and finetune approach. 14
2.3 Overview of (Scialom et al., 2021) QG/QA-based framework for evaluating the quality of generated summaries. . . . 20
2.4 Generating and answering questions from text and from KG. On the top is a snippet of the Wikipedia page for a city, and on the left is a subgraph of Wikidata corresponding to the information (entities and relations) mentioned in the text. . 21

3.1 Input/Output Examples: 1 and 2 show how the same input triple may map to multiple questions with different question types. . . . 33

4.1 Procedure for generating Q-WebNLG. . . . 50
4.2 Fine-tuning tasks used for QTT. . . . 54
4.3 **QA Accuracy**. Bold lines denote QA comparisons within/between modalities and/or approaches. Dotted arrows indicate the context X or X' that the question (q_X) is posed against to obtain the answers. . . . 56
4.4 **Comparative QG coverage DQE vs. QTT.** Total and average number of generated questions is much higher for QTT across both modality and input size. The delta increases with the size of the input. . . . 59
4.5 **Comparative QG coverage DQE vs. QTT.** Total and average number of generated questions is much higher for QTT across both modality and input size. The delta increases with the size of the input. . . . 60

5.1 **Comparing our approach against the baseline.** Using KELM and controllable QG increases coverage. Aligning questions with contexts and answers across modalities improves consistency. Machine translation provides multilingual training data. . . . 69
5.2 **Portuguese: Comparative QG coverage (Text) — Baseline vs. Our Approach.** The number of generated questions is much higher for our approach across both modality and input size. . . . 73
5.3 **Portuguese: Comparative QG coverage (KGQG) — Baseline vs. Our Approach.** The number of generated questions is much higher for our approach across both modality and input size. . . . 74
5.4 **Russian: Comparative QG coverage (Text) — Baseline vs. Our Approach.** The number of generated questions is also much higher for our approach across both modality and input size. . . . 75

5.5 **Russian: Comparative QG coverage (KGQG) — Baseline vs. Our Approach.** The number of generated questions is also much higher for our approach across both modality and input size. 76
5.6 **QA Accuracy**. Bold lines denote QA comparisons within/between modalities and/or approaches. Dotted arrows indicate the context X or X' that the question (q_X) is posed against to obtain the answers. $\mathbf{L}^{cross}$: cross-lingual answer comparison (e.g. PtBr/Ru against En); $\mathbf{L}^{same}$ denote comparison in the same language. 77

List of tables

2.1 The 13 conceptual classes of questions proposed by (Lehnert, 1978), together with a description of each class and an example for it. 7

3.1 Datasets statistics - 1. 34

3.2 Dataset statistics - 2. (S/O/S&O denotes the proportion of entities occupying the subject, object or subject as well as object positions of the triples in the data. Vocab. Size is the number of distinct subword-tokens in the NL question part of the data, Question Size is the number of word tokens in each NL question.) . . . 34

3.3 Distribution of question types in the SQ, ZQ and WQ datasets. 36

3.4 Results on the SQ dataset under a **SEEN** setting, i.e. no zero-shot constraints (B-4: BLEU-4, BSc: BERTScore, R-L: Rouge-L and M: Meteor). 40

3.5 Results on the SQ dataset under a **zero-shot** setting for RDF properties. 41

3.6 Results on the SQ dataset under a **zero-shot** setting for RDF entities (subject/object for Elsahar, question focus and the other entity in the triple for the BART models). 41

3.7 Results on the SQ dataset under a **SEEN** setting. BART$_{rdf,qt,wkdqg}$: model fine-tuned on the WKDQG data. $\texttt{Test}_A$ (for alternative) is the SQ test set with a different question type provided to the model. $\texttt{Test}_O$ (for original) is the SQ test set with the original question type. 42

3.8 Human evaluation on 50 outputs (D:Difference from the reference, A:Semantically Adequate, N:Naturalness, E:Entity Lexicalisations). For each criterion, the first two lines of the columns indicate which model is preferred. E.g., BART$_{rdf,qt,wkdqg}$'s output is judged more different from the reference 76% of the time than BART$_{rdf,qt}$'s. 42

3.9 Ablation Study: each line indicates the (non-cumulative) removal of the corresponding component from BART$_{rdf,qt}$on the SQ dataset 43

3.10 Examples of the outputs of BART$_{rdf,qt}$(O) on Test$_O$ compared with those of BART$_{rdf,qt,wkdqg}$ on Test$_A$ for the same sample (except for a different question type provided to the model). In boldface are the question-type markers provided to the model. 44

3.11 Results of QA systems with and without questions generated with our approach. The column headers denote the train-dev-test set compositions. **O** denotes original, **A** alternative and **E** enriched sets of questions. 45

3.12 Results of QA system performance on SQ with and without generated varied questions in the train, dev and/or test sets. In the table above, **O** denotes the use of original SQ questions, **E** denotes enrichment (of O) with generated varied questions in train and dev, **A** denotes a question of an alternative (to O's) question type for each sample in the test set. **_w** denotes the set of SQ samples successfully mapped to Wikidata. In brackets are the sizes of the dataset splits. 45

4.1 Q-KELM **Dataset.** For each (g, t) pairs in the filtered version of KELM (see Section 4.2), QA pairs are created for both t and $g' \subseteq g$. The question q is generated from t, heuristically aligned with the corresponding graph g', and both the text and the graph answer are extracted from t and g', respectively. 51

4.2 **Data Statistics.** Number of questions in the QA datasets; nf: the size of the question (no. of facts). Training multimodal QG models on Q-KELM and applying them to WEBNLG drastically enlarges the Q-WEBNLG^0 training data. 51

4.3 **CQ-GEN** Examples of inputs and targets for complex questions training instances, for text and for graph. [ANS], [INP], [QST], [Sp1], and [Sp2] are special tokens we use to demarcate parts of the input/target. [SUB], [PRP] and [OBJ] are used in graph inputs to demarcate subject, property and object elements of a triple. 52

4.4 QTT-DATA instances derived from WEBNLG Data. Enclosed letters refer to the triple/text above. 53

4.5 **QTT DATA**. Average, minimum and maximum number of questions for text and graph inputs of size nf (the size is the number of facts matched by the question) 53

4.6 **Consistency Results** Avg. of BScs between answers. In $_{\text{subscripts}}$ are std. dev. across 5 random runs; $^{\text{superscripts}}$ are the difference between X-Appr and Internal. X/Y indicates the QG/QA model used. QTT betters DQE on all consistency tests and for all modalities. 61

4.7 **Consistency Results with FiD** (Similar to Table 4.6.) FiD^0 is the public FiD checkpoint trained on TriviaQA. FiD^D/FiD^Q denotes that checkpoint fine-tuned on training data from DQE/QTT for that modality. 61

4.8 **Correlations with human judgements.** Comparing when DQE and QTT are used in the computation of the Data-QuestEval metric. All (p-values) « 0.001. . . 61

4.9 **Ablation results.** All models here use the text answer selector. Comp. denotes consistency comparison, and Mod. denotes QG and QA modalities, respectively. . 62

4.10 **Human evaluation for QG.** Each score is the average of all annotators' ratings. Values in brackets are the Fleiss' kappa coefficient for that particular aspect. . . . 62

4.11 Examples of the questions generated by QTT for both text and graph inputs, as well as observations of the errors present in them. 63

5.1 **Data Statistics.** Number of questions in the QA datasets; nf: the size of the question (no. of facts) 70

5.2 Q-WEBNLG^{En} instances derived from WEBNLG Data and translated into Portuguese and Russian to give Q-WEBNLG^{PtBr} and Q-WEBNLG^{Ru}. Enclosed letters refer to the triple/text above. 71

5.3 **Baseline** training data. Number of (t, a_t, q) triplets in the obtained datasets for each language. 72

5.4 Avg BSc, where questions generated by System A for a given g or t are scored against the set of generated questions by System B for the same g or t. 73
5.5 **Consistency Results**. Average of BScs between answers. In $_{\text{subscripts}}$ are std. dev. across five random runs; in $^{\text{superscripts}}$ are the difference between X-Appr and Internal, the differences provides a meaningful comparison between the baseline and our approach since each of their QA performances are different. Whenever our QA is used, the drop in performance is reduced. 78
5.6 **Correlations (Pearson's r) with human judgments.** The baseline vs our approach used to compute the Data-QuestEval metric. All (p-values) « 0.001. . 79
5.7 **QA consistency (BSc)** comparing the QG of Baseline and Ours, using mGEN for QA. 79

A.1 Results (automatic metrics) for RDF-only models. BART$_{rdf,qt,wkdqg}$: model fine-tuned on the WKDQG data and evaluated on each of the SQ, WQ and ZQ test sets. ALL shows the results proportionally averaged on the three test sets. 87
A.2 Examples of system outputs (**Generated**), used in the automatic evaluation. Other columns compare the formats for the input and target between the models (Elsahar and ours). 88

B.1 **Fine-grained analysis of QTT's QA performance (BSc). Num Facts** denote the number of facts (nf) the set of QA-pairs relate to (i.e. 1 denotes an SQ of 1 fact, 2 denotes a CQ of 2 facts etc...). The nf sets are mutually exclusive. 90
B.2 **SQ-GEN** Examples of inputs and targets for simple questions training instances for text and for graph. [ANS], [INP], [QST], [Sp1], and [Sp2] are special tokens we use to demarcate parts of the input/target. [SUB], [PRP] and [OBJ] are used in graph inputs to demarcate subject, property and object elements of a triple. . 91
B.3 **QTT** Examples of inputs and targets for **complex** TextQG and KGQG training instances. [ANS], [INP], [QST], [Sp1], and [TEND] are special tokens we use to demarcate parts of the input/target. 95
B.4 **QTT** Examples of inputs and targets for **simple** TextQG and KGQG training instances. [ANS], [INP], [QST], [Sp1], and [TEND] are special tokens we use to demarcate parts of the input/target. 95
B.5 **QTT** Examples of inputs and targets for TextQA and KGQA training instances. [INP], [Sp1], and [TEND] are special tokens we use to demarcate parts of the input/target. 96
B.6 **QTT** Examples of inputs and targets for KG-to-Text and Text-to-KG training instances. [INP], [Sp1], and [TEND] are special tokens we use to demarcate parts of the input/target. 97
B.7 **QTT** Examples of inputs and targets for EntType (on text and on graph) training instances. [INP], [Sp1], [Sp3], and [TEND] are special tokens we use to demarcate parts of the input/target. 97

C.1 **Consistency Results**. Average of **Token F1** between answers. In the first brackets () are the standard deviations across five random runs; for the right column, in the second brackets() are the differences between X-Appr and Internal. 103
C.2 **Consistency Results.** Average of **Exact Match** between answers. In the first brackets () are the standard deviations across five random runs; for the right column, in the second brackets() are the differences between X-Appr and Internal 104

C.3 **Fine-grained analysis of QTT's QA performance (BSc) for Portuguese. Num Facts** denote the number of facts (nf) the set of QA-pairs relate to (i.e. 1 denotes an SQ of 1 fact, 2 denotes a CQ of 2 facts etc...). The nf sets are mutually exclusive. 104
C.4 **Fine-grained analysis of QTT's QA performance (BSc) for Russian. Num Facts** denote the number of facts (nf) the set of QA-pairs relate to (i.e. 1 denotes an SQ of 1 fact, 2 denotes a CQ of 2 facts etc...). The nf sets are mutually exclusive. 105
C.5 **QA consistency (Token F1)** comparing the QG of Baseline and Ours, using mGEN for QA. 105
C.6 **QA consistency (Exact Match)** comparing the QG of Baseline and Ours, using mGEN for QA. 106

1

Introduction

Question generation (QG) is a task within the field of natural language generation (NLG), which is itself concerned with the machine generation of text that bears likeness with and has the qualities expected of human-written text. The QG task is a counterpoint to that of question answering (QA), which is a long-established task in the field of natural language processing (NLP). QA involves finding the correct answer to a question from some information source, whereas QG requires producing a question given some source of information as input (typically together with the answer expected of the question). The information source provided in QG or QA could be in unstructured form, such as a piece of text or an image, or structured form, such as a subgraph from a knowledge graph (KG), a table, or even a combination of inputs of different forms. QA has uses, for example, in information retrieval (Kolomiyets and Moens, 2011), in dialogue and conversation (Reddy et al., 2019), as well as in education (Agarwal et al., 2019). QG is also helpful in the same domains as QA is, but it can also be used to generate new QA-pairs to augment the data for training QA models. When QG and QA are used in concert, they are also helpful for verifying the semantic content (Wang et al., 2020; Scialom et al., 2021; Rebuffel et al., 2021), as well as representing the discourse structure (Wu et al., 2023) of text. In NLG, ordered sequences of QA-pairs as a form of discourse plan have also been shown (Narayan et al., 2023; Huot et al., 2023) to be helpful for reducing the occurrence of hallucinations and factual inconsistencies when provided as inputs to large language models (LLMs) to condition their generations upon. While the QA task has received considerable research attention, with various methods and resources proposed for them since the 1960s, QG has seen far less attention. Our primary focus in this thesis is thus QG, specifically for generating factual questions across text and KG modalities.

Questions are different amongst themselves – in that they (i) serve a variety of information-seeking needs[1] and (ii) take different forms. The forms taken by questions have a general correspondence with their purpose, e.g. the question words (who, what, which, when, where, why and how in English) used reflects the semantic type of the information being sought after (i.e. the answer). Due to the nature of questions and the relations that they have with their contexts and answers, generating a well-formed question requires ensuring the quality of their form (fluency) as much as their content (relevance to the answer; coherence with the context). Some of the types of factual questions that we focus on in this thesis include: for verification of some knowledge (such as yes or no questions), for requesting some specific information ("*How long is the Loire Valley?*") or enquiring for more information on some entity or concept (such as "*Where is the*

[1]Or on occasion for pragmatic communicative goals, such as in the form of rhetorical questions (Sun et al., 2024), but these are not the subject of this thesis.

birthplace of the designer Louis Majorelle?").

1.1 Research questions and contributions

One common thread through this thesis is our efforts to **controllably** increase the **diversity**, **comprehensiveness and coverage** of the questions that can be automatically generated from a given input. This is with the view that such improvements can facilitate advances in downstream applications, such as data augmentation and QG/QA-based methods for evaluating the semantic content of a given text. Another common thread through this thesis is our aims to leverage **QG and QA jointly** and **cross-modally**, and for **languages outside of English**, as a means for verifying the semantic content of machine-generated text. Our aim in this thesis is to study and propose methods and models to address the following research questions:

1. What benefits can pretrained language models bring to the task of question generation from a KG fact triple? How might it be leveraged for question-type and focus-controllable QG?

For this research question, we were interested in improving the generation of simple questions (Section 2.1) from KG input. The task here is to generate an NL question given a single KG fact of the form ⟨ SUBJECT , RELATION , OBJECT ⟩. While this task has been explored and methods have been proposed for it, we wished to revisit it for two main reasons. Firstly, a longstanding challenge of the task was the ability to properly lexicalise KG entities and relations that were not seen at training time.[2] The initial success of pretrained generative models such as BART (Lewis et al., 2020a) and T5 (Raffel et al., 2020) – with their parametric knowledge about language structure as well as entities and concepts gleaned from the text they were trained on – for tasks such as machine translation and summarisation – suggested that they may be suitable for addressing this challenge. Secondly, because the primary dataset for training simple KGQG models only contains one single question for a given KG triple, and these questions only seek the object of the KG fact, previous work focused on generating questions of such type and form. There are, however, multiple ways that information-seeking questions can be phrased, and different parts of the KG fact triple (the subject or the object) can also be queried about.

2. How can the comprehensiveness and coverage of QG systems be increased, and can this be done for complex questions too, as well as across modalities for QG together with QA?

We were interested here in how improvements for these aspects (complexity, comprehensiveness and coverage, and cross-modality QG and QA) might help improve QG/QA-based evaluations of the semantic content of data-to-text generation outputs (Rebuffel et al., 2021). Being able to generate and answer complex questions (on top of simple ones) over a piece of text can add to the confidence of such QA-based evaluations. This is because, although generating and answering a comprehensive set of simple questions allows us to verify the atomic facts in a piece of text, a fluent piece of text is more than just a bag of atomic facts,[3] and simple questions alone may not provide complete signal about the semantic quality of the text. For example, suppose we have

[2]For instance, some previous work relied on KG embeddings and a significant amount of engineering to generate such simple questions.

[3]A piece of fluent text can be seen as a series of atomic units of information with meaningful connections between them – where both explicit as well as implicit lexical items are woven between the atomic units to bind them (Halliday and Hasan, 1976; Prasad et al., 2014).

the following text:

"*Person X died of mesothelioma, he was director of the documentary film "Invisible Threat". It was related to asbestos exposure.*"

that is generated for the following pieces of KG facts:

{⟨ Person X , cause of death , mesothelioma ⟩, ⟨ Person X , director , "Invisible Threat" ⟩, ⟨ "Invisible Threat" , genre , documentary ⟩, ⟨ mesothelioma , has cause , asbestos exposure ⟩,}

A set of well-formed simple questions for the text could be:

{"*What did Person X die of*", "*Who is the director of "Invisible Threat"?*", "*What genre is "Invisible Threat"?*", "*What causes mesothelioma?*"}

The ambiguity in the text, which is generally seen as undesirable, might, however, cause a QG model to replace the last question with "*What is the subject matter of "Invisible Threat"?*", or a QA model might determine that the last question is unanswerable. On the other hand, being able to generate and answer complex questions such as "*What is the cause of Person X's death, which is related to asbestos exposure?*" over both the text and the set of KG facts, provides a signal about the quality of the text with a higher level of confidence than only with the set of simple questions.

3. How can cross-modal QG and QA be extended into lower-resourced languages, and can cross-lingual QG and QA between modalities be carried out?

For this research question, we were interested in identifying ways to extend the multimodal QG and QA into languages other than English, where available data and resources for training QG and QA models in these languages are much more limited than in English. While some research has been carried out for multilingual QA and initial investigations into multilingual QG have been made, such work has been focused on either QA or QG only, and on either text or KG graph only; to our knowledge, none are across a combination of all these. This is in part due to the limited availability of data for training and evaluating QA and QG in these languages, and it is further complicated by the fact that for public KGs – such as Wikidata that we use in our work, the majority of the labels for entities are in English.[4] This necessitates that cross-modal QG and QA must also be cross-lingual, e.g. questions generated from a Russian text are in Russian, but they are answered against a KG graph with English labels and the answers are compared between two languages.

1.2 Thesis outline

This thesis is structured in the following manner. Firstly, Chapter 2 provides an overview of the key concepts, tasks, datasets and models we use throughout the thesis.

Subsequently, in Chapter 3, we detail the work done to address the first research question on leveraging the pretraining and fine-tuning paradigm for generating simple questions from KG

[4] Although entity labels in other languages are available, the coverage of entity labels outside English is irregular. Most of them are for major Western European languages such as Dutch, French, German, Spanish and Italian. At the time of writing for this thesis, language coverage for Wikidata entity labels ranged from 57.3% for Dutch before quickly dropping off to 22.3% for French. Source: https://www.wikidata.org/wiki/User:Pasleim/Language_statistics_for_items

triples, using Wikidata as the KG source. We did this by collecting a dataset of questions over single KG facts that are more diverse (covering the subjects or objects of the KG facts and with different question types), which we then consolidated with other existing datasets. We show that finetuning a pretrained model such as BART together with question type controls and our expanded simple KGQG dataset permit the generation of varied questions in English and that using our model as a means for data augmentation for training a QA model helps reverse the drop in QA models' performance when the test-set distribution of question types shifts.

In Chapter 4, we extend beyond simple KGQG into a number of other aspects (QG and QA on/from both text and KG modalities and complex questions) and address our second research question. Our work here involved creating a synthetic dataset to train a single model that can generate and answer simple and complex questions across text and KG graph in English. We show that (i) the model generates a broader coverage of questions over the same piece of information in each modality, (ii) it has better QA consistency over a baseline used in a reference-less metric already shown to be useful for evaluating data-to-text generation, and (iii) replacing the baseline models with our model brings about better correlations of the metric with human judgements of data-to-text outputs.

In Chapter 5, we expand the work in the previous two chapters to address our third research question on extending cross-modal QG and QA into multilingual settings. We used machine translation to obtain synthetic data for Russian and Portuguese Brazillian, which are lower-resourced (in terms of data, NLP tools and models) compared to English. We then leverage this multilingual dataset to finetune a multilingual pretrained LM and obtain models that carry out QG and QA cross-modally into these languages. Doing so allowed us to generate Russian or Portuguese Brazilian questions in these languages for one modality, and then answer them cross-lingually in the other modality. This enabled the extension of the reference-less data-to-text metric from only English to these other languages. We then showed that using our model leads to improvements in the metric with human judgements for outputs in these languages.

Finally, we conclude and round up our findings and contributions for each research question and highlight potential future directions.

1.2.1 List of publications

The publications in the following list make up the main elements of this thesis:

- Kelvin Han, Thiago Castro Ferreira and Claire Gardent. Generating Questions from Wikidata Triples. In Proceedings of the Thirteenth Language Resources and Evaluation Conference. June 2022, Marseille, France. European Language Resources Association.

- Kelvin Han and Claire Gardent. Generating and Answering Simple and Complex Questions from Text and from Knowledge Graphs. In Proceedings of The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics. November 2023, Bali, Indonesia, Indonesia. Association for Computational Linguistics.

- Kelvin Han and Claire Gardent. Multilingual Generation and Answering of Questions from Texts and Knowledge Graphs. In Findings of the Association For Computational Linguistics: EMNLP 2023. December 2023, Singapore, Singapore. Association for Computational Linguistics.

2

Background

Sommaire

2.1 Definitions . . . 6
2.1.1 Forms of questions . . . 6
2.1.2 Simple and complex questions . . . 6
2.1.3 Terminology and notation . . . 8
2.2 Natural language generation with autoregressive language models . 8
2.3 Models for generating and answering questions . . . 10
2.3.1 Generating questions from text . . . 11
2.3.2 Answering questions from text . . . 12
2.3.3 Generating questions from KG . . . 14
2.3.4 Answering questions from KG . . . 16
2.4 Knowledge graphs . . . 18
2.5 QG/QA-based evaluation of generated text . . . 19
2.6 Bridging modalities: QG/QA across Text and KG . . . 20
2.7 Multilingual QG/QA . . . 22
2.8 Data . . . 23
2.8.1 QG/QA data for KG . . . 23
2.8.2 QG/QA data aligned across modalities . . . 24
2.8.3 Aligned QG/QA data outside of English . . . 24
2.9 Evaluation . . . 25
2.9.1 Evaluation for QA . . . 25
2.9.2 Evaluation for QG . . . 25

To aid the reader in understanding the studies we cover in Chapters 3-5, we lay out and describe in this chapter some of the relevant definitions and concepts used in our work. We start in Section 2.1 with definitions for the forms of questions we address in this work. An overview of natural language generation using autoregressive language models, which we use in our work, is given in Section 2.2. In Section 2.3, we outline the tasks of question generation and answering from text, followed by a similar outline of the same two tasks but conducted on/from KG. In Section 2.4, we discuss notions related to KGs and outline how information is stored within them as well as how it is used for QG and QA. We introduce multilingual pretrained models in Section 2.7. Finally, we discuss in Section 2.5 the recent research direction of using combined

QG/QA-based methods for evaluating the semantic content of generated text. Subsequently, in Sections 2.8-2.9, we introduce the datasets used in our work and describe the evaluation methodologies used for assessing the performance of the models we propose.

2.1 Definitions

We begin by providing here an analytic framework for the types of questions that we focus on generating and answering in this thesis. As we are interested in the multi- and cross-modal QG and QA (i.e. questions that can be asked and answered on text as well as on KG), the set of questions we generate and answer is bound to the forms of questions ask-able for the information that is captured by both modalities. While there is a greater diversity and variety of information typically found in unstructured text, the structured information in KG conforms to the types of information that can be captured by the schema of the KG being used, thereby serving as a limitation on the range of question classes addressable.

2.1.1 Forms of questions

A taxonomy of 13 conceptual classes of questions (see Table 2.1) was proposed by (Lehnert, 1978) as part of her thesis for a cognitive model for human question answering – and is oft-cited within the natural language processing (NLP) field (Graesser and Person, 1994; Rus and Arthur, 2009; Chernov et al., 2015; Krishna and Iyyer, 2019).[5] Using Lehnert's taxonomy, we define the scope of the types of questions that we generate and answer in this thesis; broadly, the questions we address come under verification, concept completion, quantification and feature specification classes of questions. These types of information-seeking questions are particularly relevant for the technologies in the domains of search (e.g. user queries), education (e.g. automated testing), dialogue systems, factuality verification and knowledge graph completion. The ability to generate such questions automatically either directly enables these technologies, or else can be useful indirectly (such as for generating data for training models to handle these tasks).

2.1.2 Simple and complex questions

In Chapter 3, we will examine the generation of simple questions from KG, which we will then build upon to extend into the generation of complex questions in Chapters 4-5. We base our definitions of simple and complex questions with respect to QG and QA from KG. Specifically, a **simple question** from KG (q_{kg}) is one that involves a single KG relation (or asking for either the subject or the object of a single KG fact triple). An analogue for q_{kg} can be easily found for text – a similar question (q_t) can be generated from/answered by text that contains the same information as that of the KG fact triple – for e.g., spans of text containing simple assertions or clauses that hold a verb (for e.g. "*Paris is the capital of France.*" in text, and 〈 PARIS , CAPITAL , FRANCE 〉 as a KG RDF triple) or adjectival/attribute information ("*The Île de France region*

[5] Lehnert's theory for having a conceptual classification of questions is that: "*If a question has been understood conceptually, it is ready to be classified conceptually. A conceptual categorisation of a question determines which memory processes will be invoked to attempt further interpretation.*" (Lehnert, 1978). She gives an example of a conversation that takes place in a context with two individuals, without any money and food in their home, having a discussion about dinner that evening, and examines how this dialogue turn: "*How are we going to eat tonight?*"; "*With silverware*" is incongruous with the context; proposing that it is a sign that the interpretation of the question was incorrect. Instead of being understood by the answering party as a question asking for the conditions that will enable them to have food for a meal that evening, it had been interpreted as seeking information for the instruments used in eating, thereby giving rise to such an infelicitous answer.

Question category	**Information sought**	**Example**
Casual antecedent	For the causes of an event	Why did the founders of the Prix Femina establish it?
Goal orientation	For the motivation/ goal of an action	Why did Anne Hildago swim in the Seine?
Enablement	For requirements that allow a state to be reached	What is needed to open a factory in France?
Causal consequent	For the consequences of an event	What happens when someone finds a *feve* in a slice of *galette de rois*?
Verification	For confirmation/ refutation of a proposition	Was Valéry Giscard d'Estaing President of France?
Disjunctive	For the selection of something from a choice of at least two alternatives	Do you prefer Johnny Hallyday or Edith Piaf?
Instrumental/ procedural	For the tools/steps to complete an action	How do you make a Paris-Brest (pastry)?
Concept completion	For information on a known concept	When did the first Tour de France take place?
Expectational	For what should/ is likely to occur in a given context	How many kisses on the cheeks when greeting someone?
Judgemental	For a value judgement on something	Was the Palme d'Or for Anatomy of a Fall justified?
Quantification	For a quantity/ numerical value	How many square kilometres does Lake Bourget cover?
Feature specification	For an attribute of an entity or event	Where is the departmental seat for Côte-d'Or?
Request	(Command/request) For an action to be carried out	Can you pick up two baguettes on the way?

TAB. 2.1: The 13 conceptual classes of questions proposed by (Lehnert, 1978), together with a description of each class and an example for it.

has a population of 12 million." in text and ⟨ Île de France , population , 12,000,000 ⟩ as a KG RDF triple).

On the other hand, varying definitions exist for what constitutes a **complex question**, which is dependent on the natural language processing domain being considered. For instance, in QG for tutoring (Gong et al., 2022) and in the discourse community (Jahangir et al., 2024), questions are often deemed complex (also termed as 'deep') when they correspond to causal antecedent, causal consequent and goal orientation questions in Lehnert's taxonomy. The term complex question is also often used in QA to refer to multi-hop questions (Talmor and Berant, 2018), that is, questions which require some inferencing or reasoning over multiple pieces of evidence (e.g. sentences or documents) in order to arrive at the answer. More specifically, the complexity of a question is tied to the number of relations (also termed as 'hops') or constraints it involves (Talmor and Berant, 2018; Usbeck et al., 2018). As KG information can be seen as an aggregation of the knowledge around an entity, which is often distributed over multiple sources, there is some correspondence between such a view of complex questions and that of multi-hop questions. Other definitions of question complexity used have been empirically defined – e.g. (Seyler et al., 2017) based it on quasi-annotated data such as questions from a popular television game show, Jeopardy!, whereas (Gao et al., 2019) used the answerability by MRC models as a yardstick for complexity. For our studies, since we work on QG and QA from both KG and text and use datasets such as WebNLG and KELM (see Section 2.8) that have a correspondence between the two, we adopt the definition tying question complexity to the number of KG triples it relates to.

2.1.3 Terminology and notation

We establish here the definitions of some of the terms and notations we use throughout this thesis. We use the term *graphs* (denoted by g) to refer to subgraphs of the Wikidata KG (Vrandečić and Krötzsch, 2014) and *texts* (t) to refer to texts (in English if not specified otherwise). A KG graph is a set of triples (also called facts) of the form ⟨ subject , predicate , object ⟩. We write X to denote the (text or graph) context of a question and X' to denote its semantically equivalent counterpart in the other modality; g' is a subgraph of g that corresponds to a question q and its answer; nf is the number of facts related to a given q (i.e. the size of its corresponding subgraph $|g'|$); $\overrightarrow{q}$ is a list of NL questions; and a_X is an answer in X whereby a graph answer a_g is either a subject or an object entity in g whereas a text answer a_t is a span in t.

2.2 Natural language generation with autoregressive language models

As we have noted in the introduction, QG is a subtask falling under NLG, which is focused on generating natural language text given some conditioning input. Modern text generation approaches are based on the **autoregressive language modelling** (LM) paradigm, which involves learning a model to be able to predict the next token based on the previously predicted tokens. The generation task is then to select either the most probable token in the vocabulary at that point (greedy decoding), or sample from the most probable subset of the vocabulary – the latter, using strategies such as top-k and top-p/nucleus sampling (Fan et al., 2018; Holtzman et al., 2020), or with beam search (Lowerre, 1990; Reddy, 1977) by maintaining multiple decoding branches and selecting the overall most probable sequence.

Initially, LM tokenisation was done at a word level, but this resulted in the need to truncate

the vocabularies so as to manage model parameter sizes; words in the training data that are not in the model vocabulary are then represented with a token for all such unknown words. This limited the possible words a model could generate; and one characteristic of early autoregressive LMs was the appearance of such tokens in the generated output thereby impacting its fluency. Subsequent innovations in tokenisation – by defining vocabularies using frequency-based character-sequence schemes e.g. WordPiece (Devlin et al., 2019) and BPE (Sennrich et al., 2016), permitted the representation of words composed at the sub-word level and significantly alleviated this constraint.[6]

Early autoregressive LM models (Mikolov et al., 2010) were based on the recurrent neural networks (RNN) architecture (Rumelhart et al., 1986), which generates the next token given a computed hidden state that is a fixed-sized vector accumulating information about all the preceding tokens generated. This single-vector hidden state, effectively a coarse summary of the preceding tokens, is limited in its ability to model long contexts where long-range dependencies are present, thereby limiting the performance of these early RNN-based models (Murphy, 2023). Attention mechanisms made prominent by (Bahdanau et al., 2016)'s work that provided a solution for addressing this by computing an attention vector across the entire preceding context, which is effectively a representation of the relative importance of each preceding token towards the current token that is to be generated.

Current autoregressive LMs are implemented with the **Transformer** (Vaswani et al., 2017) architecture which gathers these innovations – subword tokenisation and multi-head attention, with engineering to allow efficient processing of all tokens in parallel at training time. This combination permitted better modelling over longer contexts as well as efficient training over large amounts of training data, giving rise to the **pretrain & fine-tune paradigm**. This involved a first stage of pretraining (with an unsupervised language modelling objective) on text data that is a filtered crawl of the web, followed by a second phase of supervised fine-tuning (SFT) on downstream tasks such as for natural language inference, QA, translation and summarisation, sentence similarity and coreference resolution. The first few generative pretrained LMs with promising performances, such as BART (Lewis et al., 2020a) and T5 (Raffel et al., 2020) were trained with between 160 gigabytes to 750 gigabytes of data drawn from the web, and then fine-tuned with between 10 and 20 tasks (primarily from the GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a) benchmarks), whereas recent models such as GPT-4 (OpenAI et al., 2024) and Llama 3 (Dubey et al., 2024) are pretrained with data with sizes in the terabytes, as well as SFT with more than hundreds of tasks (both human annotated and synthetically generated) that is followed by preference tuning (Ouyang et al., 2024; Longpre et al., 2023) to better align the model's generated outputs with human preferences for form and content.

This training and fine-tuning enable these models to learn broad knowledge about the world (Petroni et al., 2019; Roberts et al., 2020), going beyond linguistic knowledge about structure, form and meaning to include factual knowledge, abstract concepts, and some semblance of reasoning-related knowledge (commonsense (Li et al., 2022), arithmetic (Imani et al., 2023), spatio-temporal (Li et al., 2024) etc). This parametric knowledge is useful for addressing the semantic gap (Li et al., 2021) between KG graphs (in the form of RDF triples that can be seen as an underspecification of a corresponding NL utterance) and natural language text that is expected of data-to-text generations, and we investigate the use of such pretrained models from Chapter 3 for the task of KGQG and continue to leverage these pretrained models for the

[6]For example, instead of having to represent the words "dog", "doggy", "dogged" and "doggedly" with individual tokens, a model – depending on its subword tokenisation scheme – might instead use tokens for "dog", "_ged", "_gy", "_ly" where the last three may be shared with other lemmas, giving a significantly reduced vocabulary.

remainder of our studies.

2.3 Models for generating and answering questions

The task of question generation is to learn a function that generates the sequence of tokens that make up the question by conditioning on the input context. When using an autoregressive pretrained language model (see Section 2.2) to do so, such a model can be expressed in the following form:

$$p(q_1, ..., q_n) = \prod_{i=1}^{n} p(q_i|X, A, q_1, ..., q_{i-1}) \; ; q_i \in V \tag{2.1}$$

where q_i denotes the tokens that make up the question and are drawn from a vocabulary V, and X denotes the input context provided to the model, which can be either a paragraph of text (Du et al., 2017) (see also Sections 2.3.1), a KG graph (Serban et al., 2016) (see also Sections 2.3.3 & 2.4), meaning representations (Deng et al., 2022) or an image (Mostafazadeh et al., 2016). A denotes the expected answer to the question to be generated.

Most QG models generate a single question given a (X, A) tuple, which is largely due to the fact that benchmark QG/QA datasets such as SQuAD and HotpotQA have been constructed to have a single question for a given input. Some work (Lopez et al., 2021) including ours in Chapters 4 & 5 generate $\overrightarrow{q}$, a sequence of questions given the input. Other work (Shakeri et al., 2021; Ushio et al., 2023a,b) have also been done to generate both the question and answer given only the context, in which case the task is termed question and answer generation (QAG).

On the other hand, the task of question answering is to learn a model that can return the answer to the question given the context. When using an autoregressive pretrained language model to generate the answer given some question and context, such a model can be written as:

$$p(a_1, ..., a_n) = \prod_{i=1}^{n} p(a_i|Q, X, a_1, ...a_{i-1}) \; ; a_i \in V \tag{2.2}$$

where a_i denotes the tokens that make up the answer that is conditioned on the question Q and the context X. Within the QA task are a number of specialised areas of focus with their unique requirements and task set-up, these include machine reading comprehension (MRC), where the goal is to derive the answer to a question from a given text, open-domain QA (ODQA) (Voorhees, 2000) that typically relies on modules to retrieve relevant contexts in order to answer the question, closed-book QA (Roberts et al., 2020) where it is expected of a model to answer question-based on its parametric knowledge, and conversation QA (Reddy et al., 2019) where the question and the answer to be given are part of the dialogue. In this thesis, we focus on generating and answering questions from/on the text and KG modalities.

Research on automating QA predates those for QG, and the former continues to enjoy more research attention over the latter. Systems were already being developed for QA as early as the 1960s and 70s (Green et al., 1961; Woods, 1977), whereas QG as a field started to consolidate only after the TREC-led series of Question Generation workshops and shared challenges (Rus et al., 2010, 2008). The pioneering QA systems (Sections 2.3.2 & 2.3.4) were rules-based, as were the early QG approaches that came after (Sections 2.3.1 & 2.3.3). These rules-based approaches were based on written grammars and templates and mostly focused on narrow domains (e.g. baseball statistics and data on lunar geological specimens), which were brittle and failed to generalise well. They gave way to machine learning and early neural network-based models that allowed some

flexibility in modelling. In recent years, the success of the Transformer architecture (Vaswani et al., 2017) and the pretrain & finetune (Devlin et al., 2019) paradigm over vast amounts of text data brought about substantial improvements in QG and QA and are currently the *de rigueur* model architectures of choice for these tasks.

The following sections provide an overview of the tasks in each of these modalities, briefly describing the main approaches for them and outlining the main challenges/limitations found in each. We structure the following sections by starting with QG, as it is the primary focus of this thesis, and from the text modality, as one of the earliest works on QG was focused on generating questions from text for educational purposes. The rest of the sections are structured to position QG of a particular modality alongside QA of that modality.

2.3.1 Generating questions from text

Initial QG efforts were motivated towards automatically generating questions for educational purposes and were based on **rules-based transformations** on the surface forms of assertions to convert them into questions. This was typically done by removing certain lexical information in the assertion and inserting question-related words (for e.g. a simple example involves the replacement of the named entity "*Charlemagne*" with a "*Who*" and the period with that question mark in the statement "*Charlemagne was the first king of the Carolingian Empire.*"). An early system by (Mitkov and Ha, 2003) for generating multiple-choice educational questions relied on key term extraction using syntactic parsing followed by a set of rules-based transformations to generate the generation. Their evaluation of the system showed that about 60% of the generated questions were of acceptable quality and that though the remainder required some manual post-editing, the effort to do so was nearly five times less than having a teacher generate the question manually from scratch.

Subsequent work for QG from text sought to extend rules-based approaches to handle more complex linguistic phenomena, such as those with clauses or embedded structures. (Heilman and Smith, 2010)'s system was designed for use on texts found in Wikipedia and used rules to first simplify complex constructions in order to more easily transform them into questions. For instance, the example sentence they give: "*During the Gold Rush years in northern California, Los Angeles became known as the 'Queen of the Cow Counties' for its role in supplying beef and other foodstuffs to hungry miners in the north.*" was transformed into "*Los Angeles become known as the 'Queen of the Cow Counties' for its role in supplying beef and other foodstuffs to hungry miners in the north*". By carrying out this intermediate transformation of the statement, the sequence of transformation required to go to "*What did Los Angeles become known as the 'Queen of the Cow Counties' for?*" is simplified. To improve the generalisability of their system, instead of writing specialised transformation rules for each type of linguistic phenomena, they used a set of general-purpose question transformation rules and relied on over-generation and ranking with a logistic regression model to identify the most suitable generation candidate.

Despite this early progress for QG, such initial rules-based models were brittle and would break down when their use was extended to constructions not covered by the grammar or when applied to a new domain. QG methods then began shifting towards **sequence-to-sequence (seq2seq) autoregressive modelling** (Equation 2.1) using recurrent neural networks – such as with Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (Cho et al., 2014) architectures, after breakthrough performances were achieved in machine translation (Sutskever et al., 2014) with them. (Du et al., 2017) trained such a model with an added attention mechanism as well as using pretrained GloVe word embeddings (Pennington et al., 2014a) for encoding the context. Their model, trained on the SQuAD (Rajpurkar

et al., 2016) dataset, achieved significantly better performance in automatic and human evaluation over (Heilman and Smith, 2010)'s rules-based system. One limitation of these early neural approaches lies in their fixed vocabularies, which are fixed at the top-k most frequent words in the training dataset. As a result, the long tail of rare tokens (in particular for named entities, which is important for factoid QG) is either represented with a token for unknown words or else is difficult for a model to learn a good representation of. To address this, subsequent work (Song et al., 2018; Qiu and Xiong, 2019) incorporated copy mechanisms (Gulcehre et al., 2016) to facilitate better handling of such rare words. Success in machine translation with the Transformer architecture (Vaswani et al., 2017) and development in the pretrain & fine-tune paradigm (Peters et al., 2018; Radford et al., 2018) were extended into QG to give rise to increasingly capable QG models. The focus also began to shift from generating simple factoid QG models to more complex and/or specialised questions such as multi-hop (Fei et al., 2022; Cheng et al., 2021), deep (Pan et al., 2020), conversational/dialogue (Wang et al., 2018), clarification (Rao and Daumé III, 2018) questions.

2.3.2 Answering questions from text

A closely related task to QA is information retrieval (IR), where a user query (which need not be a fully-formed NL question, for e.g. a sequence of terms in an Internet search engine query) is addressed by matching the query over a large collection of documents (in the tens of thousands to millions) to find the most relevant documents (typically from 10 to 100). In other words, the information needs of a user's query is met by returning him/her a set of documents where the user may find the answer to the query. These approaches represent the document as a collection of vectors (either using weighted word-count methods such as TF-IDF (Sparck Jones, 1988) and BM25 (Robertson et al., 1995), or dense embeddings such as those from BERT-like models (Khattab and Zaharia, 2020; Karpukhin et al., 2020)), and use similarity-based measures such as cosine similarity to identify the set of documents most similar to the query (and therefore most relevant to addressing it).

The TREC-8 Question Answering Track Evaluation (Voorhees and Tice, 2000) helped usher (Roy and Anand, 2022) a new research direction that went from document retrieval-based answering towards extracting spans of text in a document as answers for queries. Initial rules-based systems for such general ODQA depended on pipelines, the one proposed by (Harabagiu et al., 2000) first parses and then classifies the question, followed by extraction of the set of entities from the text (on the assumption that questions tend to seek information on entities within the text) before handwritten templates with the help of ontologies like WordNet (Miller, 1994) are used to extract the answer.

Over time machine learning (ML) techniques were introduced to improve parts of the rules-based pipeline approaches. For instance, (Moschitti et al., 2007) focused on improving question and answer classification and utilised a Support Vector Machines (SVM) (Boser et al., 1992) model for the task to obtain better answer and question matching, which led to improved overall QA performance. While bringing about improved performances, these early ML-based methods still required extensive feature processing and struggled with the limited amount of training data available then; for e.g. (Moschitti et al., 2007)'s SVM model required syntactic and argument structure parsing of the input before building and selecting features from these parses. The release of the SQuAD dataset (Rajpurkar et al., 2016) (see Section 2.8) with more than 100,000 extractive QA pairs, provided a meaningful amount of data to enable the development and testing of QA using ML techniques and neural models. Accompanying their dataset release, (Rajpurkar et al., 2016) proposed a logistic regression model that uses manually built features

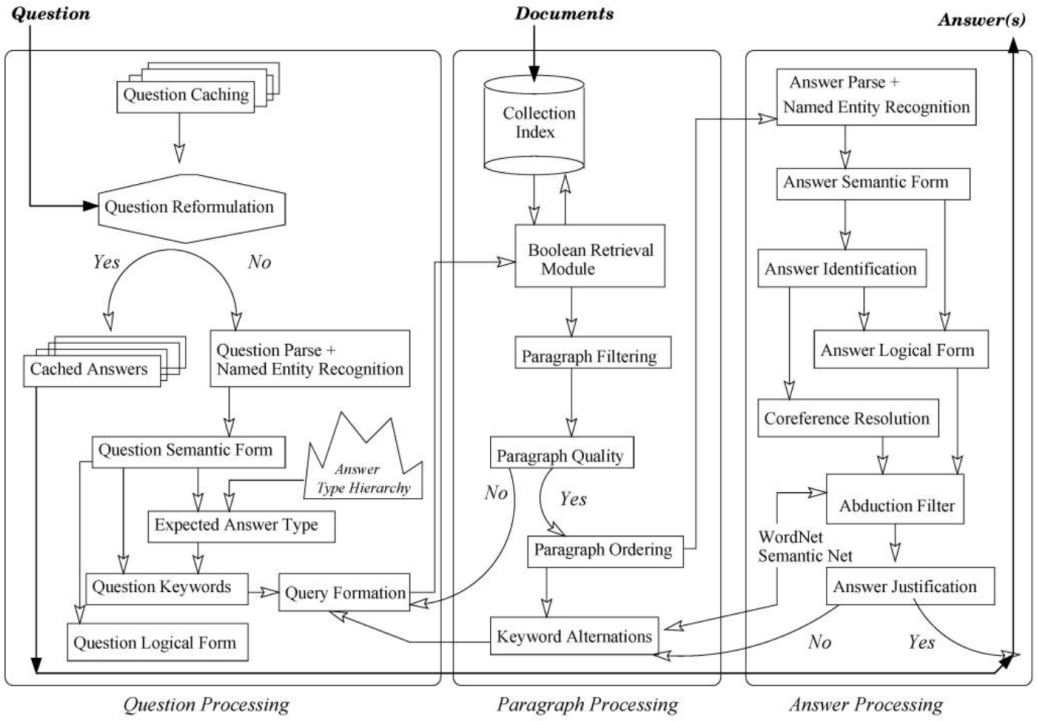


FIG. 2.1: Graphical overview of the components in the FALCON (Harabagiu et al., 2000) rules-based pipeline QA system.

from n-gram frequencies and dependency tree paths to predict the answer to the question from the set of constituent parses in the context. Shortly after, methods using neural networks (Xiong et al., 2017; Wang and Jiang, 2017) were proposed for the SQuAD dataset that were capable of obtaining major improvements over (Rajpurkar et al., 2016)'s approach.

The advances from the pretrain & finetuning paradigm significantly shifted the way QA – especially ODQA, was carried out. The authors of the BERT (Devlin et al., 2019) model showed that it was possible to use their model (a pretrained encoder) and do away with the need for heavy pre-processing to achieve significant improvements on SQuAD. They obtained results better than previous baselines, as well as over human performance (87.4 vs 82.3 exact match, 93.2 vs 91.2 token F1). Since the QA task in SQuAD is extractive (i.e. the answer is exactly a span in the context document used for answering), such BERT-based QA models cast the answering task as token-level classification (i.e. whether a token is within the span of text that corresponds to the answer) whereby the question and text are encoded to give a contextualised representation of the entire input, which is then passed through an output layer that gives a binary prediction of whether the token is inside or outside the answer span.

Shortly after the BERT model's release, investigations (Radford et al., 2019; Petroni et al., 2019) suggested that BERT and similar pretrained models such RoBERTa (Liu et al., 2019b) which is trained with 10 times more data, and GPT-2 (Radford et al., 2019) a decoder-only causal LM, showed promise in answering questions without having the need for access to a context. This formulation of the QA task is termed **closed-book QA** in the sense that, when given a question, an answer to the question could be obtained from the model just from the knowledge stored in its parameters; knowledge acquired after having been trained over these large amounts of text.

In parallel to the research on closed-book QA, another direction (Guu et al., 2020; Lewis

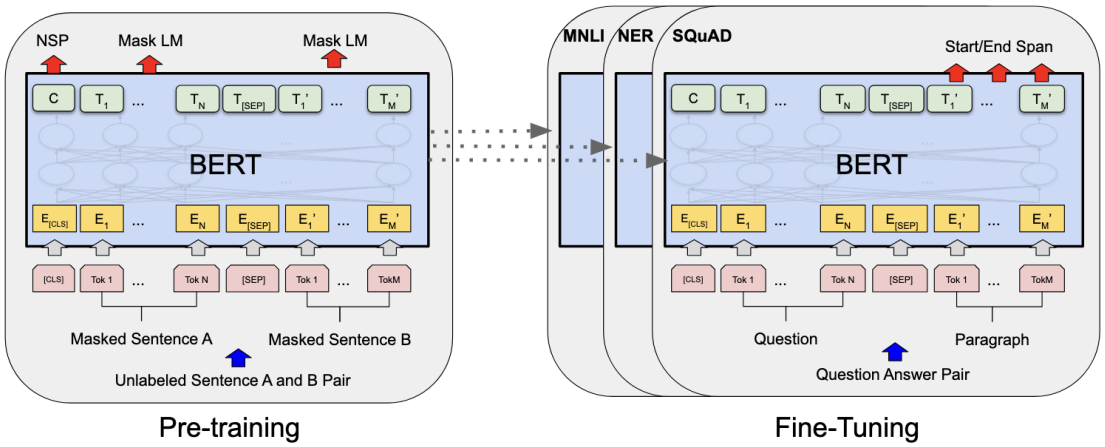


FIG. 2.2: Overview of the BERT (Devlin et al., 2019) model's pretrain and finetune approach.

et al., 2020c; Borgeaud et al., 2022) investigated ODQA using IR methods to retrieve relevant contexts for an NL question and to go on to use those contexts to answer the question. An example of this is the Fusion-in-Decoder (FiD) system first proposed by (Izacard and Grave, 2021), which achieved state-of-the-art on the TriviaQA and NaturalQuestions datasets. Their system combined the dense passage retrieval model from (Karpukhin et al., 2020) and the T5 encoder-decoder pretrained model (Raffel et al., 2020). The system encodes each of the relevant retrieved passages using the T5 model and concatenates each of their encoded representations together with that of the question before giving this as input to the T5 decoder to condition the generation of the answer to the question. In this generative form of QA, the answer is generated token by token in contrast to the extractive QA approach (see above). Subsequent work by (Asai et al., 2021) extended this approach to multiple languages.

A continued limitation of pretrained autoregressive LMs – from BART, T5 and up to state-of-the-art billion-parameter models such as GPT-4 and Llama 3 – lies in their being prone to introduce hallucinations or non-factual information in their outputs. This is usually most severe when the model only has its parametric knowledge to rely on (i.e. the closed-book QA setting); as such arguments have been made for augmenting these LMs generative abilities with retrieval when addressing knowledge-intensive tasks (Lewis et al., 2020c). This direction, termed **retrieval-augmented generation** (RAG), has become an established method for QA with the latest large LMs, especially for use in specialised domains (Siriwardhana et al., 2023), whose data may not have been heavily represented in the training data for the large LMs, thereby increasing their propensity for hallucinations. Our downstream QA evaluations in Chapters 4-5 make use of models that are part of these RAG approaches.

2.3.3 Generating questions from KG

The task of generating questions from KG falls under data-to-text generation in NLG, which covers the generation of text from some structured information such as KG graphs but also tables and meaning representations such as Abstract Meaning Representation (Banarescu et al., 2013; Fan and Gardent, 2020) for text and Discourse Representation Structures (Kamp, 2004; Liu et al., 2021) for discourse and dialogue as well as query languages such as SPARQL (Lecorvé et al., 2022).

Initial KGQG efforts (Olney et al., 2012; Seyler et al., 2015; Song and Zhao, 2017; Seyler

et al., 2017) were similarly **rules-based** and focused on generating **simple questions** from KG graphs. They leveraged the observations that (i) generating factual questions from a given KG graph typically involves copying mentions of relational and entity references that appear together with the answer in the text; and (ii) generic syntactic forms of NL questions for a given relation are shared between entities of similar types, e.g. "*What is X of Y?*". Many of these approaches, therefore, made use of hand-crafted templates, which required significant human effort, generalised poorly and were difficult to scale up.

The neural-based models that came after, brought improvements over rules-based approaches by being able to learn patterns for lexicalisation from training on large datasets of (KG triple, NL question) pairs. They also do not require as much manual intervention as rules-based ones. Within the semantic web community, (Kumar et al., 2019) introduced a neural question generator over KG where the complexity of the output can be controlled. (Serban et al., 2016) first trained a recurrent encoder-decoder network with attention on SimpleQuestions dataset (Bordes et al., 2015). To handle unseen entities, they used a placeholder for the subject entity in the question and trained on the delexicalised data. These early neural models, however, still often required complex processing such as de-lexicalisation and template-writing that are also challenged when faced with unseen relations and types.

In **simple KGQG**, (Elsahar et al., 2018) focused on generalisation in a zero-shot setting. To handle unseen entities and properties, they enriched the KG input with a lexicalisation of the input KG property obtained through distant supervision and with the Freebase type of the input subject and object entities. They also delexicalised the data by replacing matching terms in this additional information and the output questions with placeholders, replacing these by their value after inference. The RDF triples are initialised with learned TransE (Bordes et al., 2013a) embeddings. Two separate encoders are used for the RDF and the textual context, and the decoder attends to both. (Liu et al., 2019a) expanded the contextual information used by (Elsahar et al., 2018) with information about the domain and the range of the input property. To improve question specificity, they propose an answer-aware loss by optimising the cross-entropy between the generated question and the answer type words. (Bi et al., 2020) developed an encoder-decoder question generator, which also enriches the input facts with additional information and constrains the decoder with word types to preserve the adequacy of the generated question.

An early work on **complex KGQG** sought to leverage a compositional approach for generating CQs. (Zhao et al., 2019), (i) to address the limited availability of CQ data which makes directly training a complex KGQG model difficult, and (ii) by recognising the promising performance of neural models for simple QG (Serban et al., 2016). They proposed an LSTM-based model that seeks to generate a CQ given a query graph.[7] They supplement their model with an encoder trained on NL SQs, with the intuition that SQs (or approximates, i.e. SQs that are similar) – effectively decompositions of the CQ – can be matched to parts of the input CQ query. By concatenating the representations for SQs with the input, the model is allowed to learn to attend to and copy information from the SQs when verbalising the CQ from the query graph. Later complex KGQG models sought improvements by using meta-learning and graph retrieval (Zhang et al., 2022) and graph neural networks for better capturing the structural information from the input graphs (Chen et al., 2020, 2023).

[7] A form of structured representation (Yih et al., 2015; Lan and Jiang, 2020; Qin et al., 2021) for an NL question, containing entities, relations and variable nodes for implicit entities and the answer entity.

Representations for KG facts

There are several methods for representing a KG fact when training models for data-to-text generation. One way is to use KG embeddings such as TransE and DistMult, where the unique representations for each entity relation in the KG embeddings can be directly leveraged; this was the method used in (Serban et al., 2016)'s system that we introduced above. One limitation of such KG embeddings-based models (including when used for KGQA, see Section 2.3.4) is that the KG embeddings that they rely on require intensive resources for training and for inference; furthermore, once trained, it is not trivial to update KG embeddings with new entities and relations that appear subsequently.

Another direction involved the use of early word embeddings like word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014a) for the labels of KG entities and relations to initialise a representation for them, as was done in data-to-text generation by (Marcheggiani and Perez-Beltrachini, 2018; Moryossef et al., 2019).[8] Finally, recent work that leverages pretrained LMs and LLMs for data-to-text generation (Kale and Rastogi, 2020; Ribeiro et al., 2021; van der Lee et al., 2023,?) treat the data-to-text task as a sequence-to-sequence transformation of the graph input into NL text. This approach places the (subject, relation, object) triples in the KG graph in a linear sequence, which has the effect of bringing the graph input closer to the form of data (NL text) seen at training by these pretrained LMs/LLMs.[9] This is also the approach taken by (Oguz et al., 2022) in their work proposing a unified representation across modalities (KG, tables and lists) to solve multimodal QA over text and structured information.

2.3.4 Answering questions from KG

Some of the earliest QA work involved answering questions over a structured knowledge source such as databases containing baseball statistics or information about rocks from the moon (Green et al., 1961; Woods, 1977). The task of KGQA, that is, answering questions by identifying answers in a knowledge graph, is a specific form of QA over structured information that has arisen from the development of KGs (Section 2.4). Approaches for KGQA can be found from at least five broad families, namely: (i) semantic parsing, (ii) information extraction, (iii) KG embeddings, (iv) graph neural networks, and (v) generative approaches with pretrained LMs and LLMs. Much of the research into KGQA has been focused on answering simple questions, although recent efforts (see the end of this subsection) have started shifting towards addressing complex QA.

Semantic parsing (Kate et al., 2005; Lu et al., 2008) approaches work by parsing NL questions into query languages (e.g. SPARQL) so that these can be executed as a search against a KG to obtain the answer. A key goal in semantic parsing was to learn some mapping between the NL phrases in the question and components to compose a logical form. This is usually done by maintaining a lexicon of NL utterances together with the KG relations they correspond with. Context-free grammars such as Combinatory Categorial Grammar (CCG) (Steedman, 1987) are

[8] Despite these word embeddings not taking a word's context into consideration e.g. the word "*bank*" has the same representation regardless of whether it was used to denote a financial institution or the side of a river, since they were trained over a relatively large amount of text, they still provide a useful means for initialising the representation for a KG entity or relation than learning this from scratch with smaller amounts of data generally available for the actual data-to-text task. This method was also used in RDF2Vec (Ristoski and Paulheim, 2016), which sought to learn KG embeddings by representing entities with the paths obtained for them from graph walks across their neighbourhood.

[9] A simple linearisation may, however, see a linearised input with repeated information, especially in the case of a 'star-shaped' graph where a central entity may hold relationships with multiple other entities.

then used to compose a logical form meaning representation from the matched components (Yao et al., 2014). It is worth noting that KGQA research in the early 2010s was mostly carried out on the Freebase KG, which was the largest and state-of-the-art among publicly accessible KGs at the time. It, however, held information in a less structured manner compared to traditional ontologies and more modern publicly accessible and maintained KGs such as DBpedia and Wikidata.[10] Therefore, a meaningful part of the work in this period (Berant et al., 2013; Reddy et al., 2014) was focused on addressing the challenge of entity and relation linking over the noisy information due to Freebase's schema.

Information extraction-based approaches such as (Yao and Van Durme, 2014) use entity-linking (Milne and Witten, 2008) on a query to identify the key entity (or entities) of interest in the query and use them to query a k-hop subgraph from the KG, which is used as the candidate set for finding the answer to the query. These approaches are based on training models to produce a score-based judgement; for example, a logistic regression model in (Yao and Van Durme, 2014) on the match for the candidates to the answer. Another class of approaches (Huang et al., 2019; Mohammed et al., 2018) is based on **KG embeddings** such as TransE (Bordes et al., 2013b) and DistMult (Yang et al., 2015), which are obtained by training over an entire KG (or a large portion of one) to have embeddings for its entities and relations: (i) that place similar ones closer to each other in the embedding space, and (ii) where a rotation of the embedding for a subject entity s using a relation embedding should give the embedding for the entity that holds the relationship with s in the KG. (Huang et al., 2019) uses an LSTM model – at training, their model learns a representation for a simple NL question, which is optimised to match to find the TransE embeddings for the subject entity and relation mentioned in the question; at inference time, answering the question is then merely finding the closest fact in the KG that corresponds to the matched embeddings. We use the models of (Huang et al., 2019; Mohammed et al., 2018) for downstream evaluation in our work in Chapter 3.

These earlier methods mainly focused on the answering of simple questions i.e., questions which verbalise a single KG fact. More recently, some researchers have started to address complex KGQA i.e. questions on more than one KG fact (Saha et al., 2018; Christmann et al., 2019; Perez-Beltrachini et al., 2023). One way of addressing complex KGQA is to find better ways to model the graph structural information that is inherent in KG graphs of which one is using **graph neural networks** that carry out message passing (Gilmer et al., 2017; Kipf and Welling, 2017) between nodes and edges to obtain a structurally-informed representation of the KG graph. Another direction is to include the use of **generative LLMs** when answering the question – for instance, (Sen et al., 2023)'s system uses existing methods for KG graph retrieval for a given NL question, and then linearise the KG graph (using the text labels for its entities and relations) so that it can be provided as input together with the NL question to an LLM to return the answer in a zero-shot manner. Another model proposed by (Yasunaga et al., 2021) for multiple-choice KGQA uses both a pretrained LM as well as KG embeddings for KGQA to construct a joint graph containing embedding constructed from the LM (for the NL question and entities in the graph) as well as retrieved nodes for the entities in a retrieved KG graph for the question. They use a GNN architecture over this joint graph to identify the entity that is the answer to the question.

[10] This is because Freebase employed a schema that organised information on 'type-property-value' model and allowed relatively free creation of entity types and properties. This led to instances of synonymous properties and even conflicting information (Bollacker et al., 2008). More recent KGs such as DBpedia and Wikidata have tighter management of the KG schema and established procedures for property creation (see, for example `https://www.wikidata.org/wiki/Wikidata:Property_proposal`).

2.4 Knowledge graphs

In each of our studies for this thesis, we carry out QG and QA from the Wikidata KG (see Figure 2.4 for an example subgraph from Wikidata), which is currently the largest public KG. Wikidata has accumulated more than 100 million pages[11] that have been contributed, edited and maintained by volunteers across the world, making it a powerful epistemic resource. The ability to generate and answer questions on a KG like Wikidata facilitates many uses – for instance, for use in educational exercises, for verification of information (as we investigate in Chapters 4-5)

A KG is defined as being a data structure/collection that is "*intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities*" (Hogan et al., 2021). More specifically, KGs are designed and populated with information that allow the representation of knowledge about the relationships between entities (such as persons, organisations, events, objects, places, concepts etc...), as well as frequently, the attributes possessed by some of these entities (e.g. the date of birth of a prominent figure). More concisely, a KG can be expressed as having the following form:

$$T = \{\ (h, r, t) \mid h, t \in E,\ r \in R\} \tag{2.3}$$

where E and R are, respectively, the set of entities, and the set of relations in the KG, i.e. (h, r, t) is a triple fact denoting a directed relationship r between the entities h and t.

KGs gained popularity in the early 2010s, with the disclosure[12] of Google's construction of its Knowledge Graph, which it has used to serve results on its search engine. An early public KG Freebase (Bollacker et al., 2008), was acquired by Google and folded into its proprietary graph in 2010 before Google re-released Freebase knowledge to the public again by offering to map Freebase schema and data to the then-nascent but growing Wikidata (Vrandečić and Krötzsch, 2014) (Pellissier Tanon et al., 2016). Concurrently, other private enterprises such as Airbnb, Amazon, eBay, Facebook, IBM, LinkedIn, Microsoft and Uber have also built and maintained their own KGs to serve their business needs (Hogan et al., 2021). Besides Wikidata, there also exist other modern large-scale publicly accessible KGs such as DBpedia (Lehmann et al., 2015), as well as ConceptNet(Speer et al., 2017) and BabelNet (Navigli and Ponzetto, 2012). In the European Union, the EU Knowledge Graph[13] has been constructed to hold information about EU institutions, legislation and regulations as well as grant information.

KGs rely on three Semantic Web technologies that are based on standards deliberated and recommended by the World Wide Web Consortium, namely the Resource Description Framework (RDF)[14], the Web Ontology Language (OWL)[15], and the SPARQL query language [16]. RDF defines a graph data structure with nodes (entities)[17] that are linked by directed edges (i.e. a relation between node E_i to E_j does not directly imply the same relationship from E_j to E_i, unless the semantics of the relation is defined to be so). OWL provides a standard to represent the ontological knowledge about concepts, classes and groups of entities, the relations that hold amongst them and therefore, the ability to carry out reasoning on entities based on their class

[11] https://stats.wikimedia.org/#/wikidata.org/content/pages-to-date
[12] https://blog.google/products/search/introducing-knowledge-graph-things-not/
[13] https://linkedopendata.eu/wiki/The_EU_Knowledge_Graph
[14] https://www.w3.org/RDF/
[15] https://www.w3.org/OWL/
[16] https://www.w3.org/TR/sparql11-query/
[17] There are three types of nodes – (i) resource; (ii) literal; and (iii) blank nodes – that allow (i) things; (ii) values; and (iii) as-yet unspecified resources/values to be represented in the KG.

and relational knowledge. Four relations are primarily used in Wikidata to capture ontological information between the entities and concepts, namely: "class" (Q16889133), "entity" (Q35120), "instance of" (P31), and "subclass" of (P279).[18]

The RDF and OWL standards facilitate the merging and mapping of initially different KGs – for example, once a resource node of a given KG is resolved as referring to the same entity E as that of a node in a different KG, work on merging, propagating and/or de-conflicting the relational knowledge about E from both KGs can be carried out. Notably, two of the largest public KGs, DBpedia and Wikidata, were partially developed in this manner. In Chapter 3, we work on mappings to port datasets of RDF triples and their lexicalisations originally from Freebase and DBpedia into the Wikidata KG. The entities and relations in KGs are tagged with labels (primarily in English, and occasionally in other languages), and we use these labels (in the form of $\langle$ SUBJECT , RELATION , OBJECT $\rangle$ triples) as input for QG and QA from KG.

2.5 QG/QA-based evaluation of generated text

Automatic evaluation for generated text in NLG (such as summaries or data-to-text outputs) has for a long time been based on evaluating the similarity of the generated text (using either surface or contextual embedding similarity, see Section 2.9 above) against one or more human-written reference(s). As the capabilities of QG and QA models improved, proposals (Eyal et al., 2019; Wang et al., 2020) were made to use such models to automatically evaluate the quality of machine-generated summaries. The intuition is based on the observation that a good summary should only consist of a subset of the most relevant information in the original document; as such, there should be some level of correspondence between the questions that can be answered on the summary as well as on the source document.

In the system proposed by (Eyal et al., 2019), a set of QA pairs that have been defined as being key in the document being summarised are posed together with the generated summary to an off-the-shelf QA model; the quality of the summary is then assessed based on the proportion of the questions that can be answered correctly using the summary. This QG/QA evaluation paradigm for summarisation was not entirely new, albeit the automation was – work from as early as the 1990s (Jing et al., 1998; Narayan et al., 2018; Chen et al., 2018) already sought to evaluate machine-generated summaries by assessing whether human annotators are able to answer a set of questions from the source document against the summary.

The arrival of pretrained/large language models such as BART (Lewis et al., 2020a) and T5 (Raffel et al., 2020) brought about further improvements in the capabilities of QG and QA models and a body of research on QG/QA-based evaluation of summary quality were developed (Wang et al., 2020; Honovich et al., 2021; Scialom et al., 2021; Fabbri et al., 2022) as well as extended into other NLP tasks such as for text simplification (Trienes et al., 2024), knowledge-grounded dialogues (Honovich et al., 2021). It was also extended for evaluating NLG outputs conditioned on information across domains, such as for evaluating image captions (Lee et al., 2021) and data-to-text generations (Rebuffel et al., 2021; Zhang et al., 2023b).

Our work in Chapters 4-5 seeks to improve multi-modal QG and QA to aid QG/QA-based evaluation for data-to-text generation. Prior work by (Rebuffel et al., 2021) created synthetic multimodal-QA/QG datasets to train the QG/QA models used for cross-modal evaluation of KG-to-Text outputs. They show that the models can be used to evaluate KG-to-Text generation models and report better correlations between their measure, the Data-QuestEval metric, with human judgements of semantic adequacy than existing automatic metrics. Our work, which

[18] https://www.wikidata.org/wiki/Wikidata:WikiProject_Ontology/Modelling

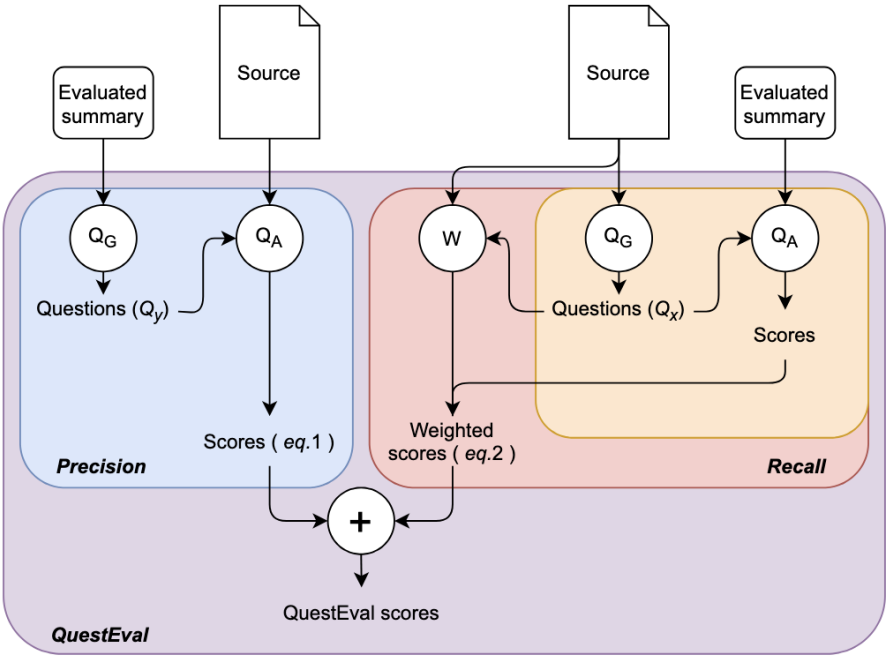


FIG. 2.3: Overview of (Scialom et al., 2021) QG/QA-based framework for evaluating the quality of generated summaries.

extends (Rebuffel et al., 2021)'s, is based on the intuition that a QG/QA-based measure of the semantic quality of NLG generations should be supported by a set of QG and QA models that can systematically generate and answer questions across as wide a swath of information in all modalities to allow a more confident assessment about the quality of generated output.

2.6 Bridging modalities: QG/QA across Text and KG

An area of focus in this thesis (our second research question, see Section 1.1) is the ability to generate questions based on input from one modality, which can be answered by a context that is in another modality. Figure 2.4 provides an illustration of this. We refer to these as 'cross-modal questions' here. Bodies of work exist for the text-vision modalities, in visual question generation (VQG) (Mostafazadeh et al., 2016; Zhang et al., 2017; Krishna et al., 2019), which is the task of generating NL questions from an image, and visual question answering (VQA) (Antol et al., 2015; Zhang et al., 2016; Yu et al., 2020), which is the task of answering NL questions over an image.

With regards to QG, there exists a historical distinction between QG from text and QG from KG. Due to the fact that information that can be simply expressed in a piece of text might have to be captured in a KG via multiple relations because of the KG's schema[19], the distribution of the types and forms of questions likely to be asked from text tend to be different from the questions likely to be asked from KG, and is especially for complex questions. As a result, this is reflected in the forms of the questions collected for the QG/QA training data in each of the

[19] As a simple example, "*A is the grandparent of C*" in text might be captured in a KG as ($\langle$ A , PARENT , B $\rangle$, $\langle$ B , PARENT , C $\rangle$).

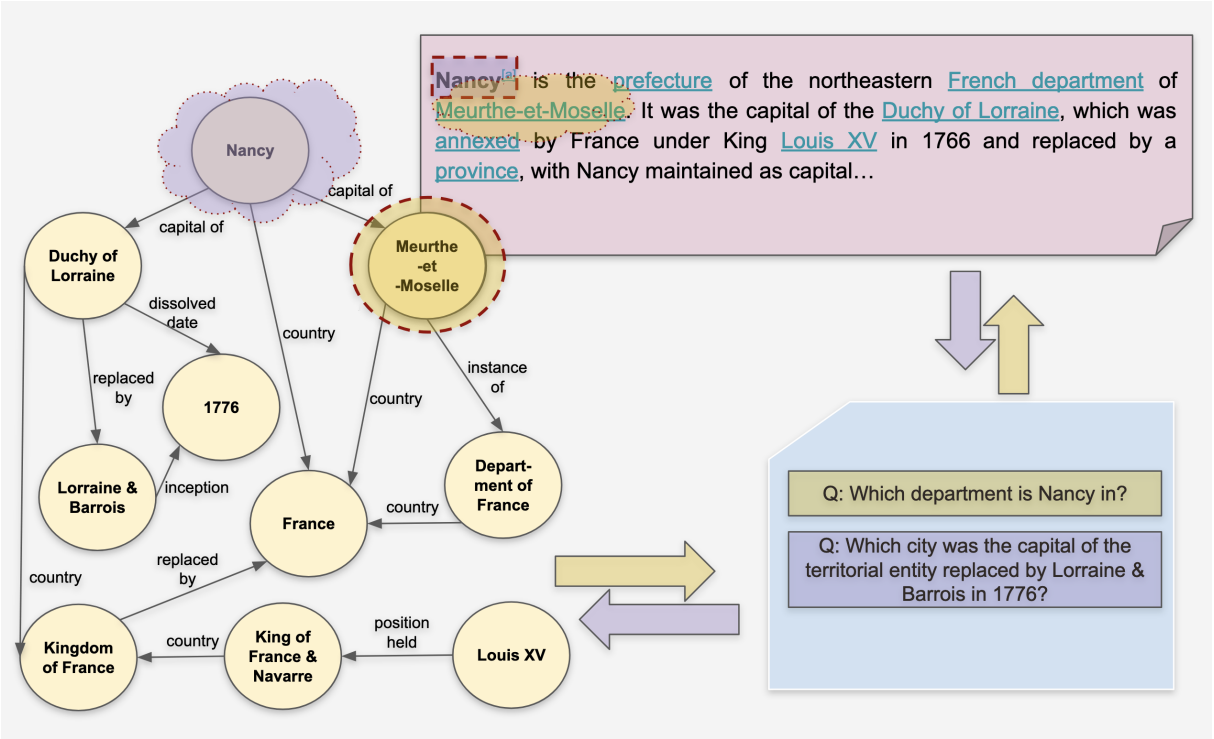


FIG. 2.4: Generating and answering questions from text and from KG. On the top is a snippet of the Wikipedia page for a city, and on the left is a subgraph of Wikidata corresponding to the information (entities and relations) mentioned in the text.

modalities, and models trained on such data naturally have a distinction in their distribution when they come from different modalities. To our knowledge, there was no previous work done on QG from text with the view that the generated questions would be answered with a KG input, or vice-versa.

On the other hand, efforts have been made in QA to leverage information from different modalities to answer a question, and some of such recent methods can easily be applied to answering questions across modalities. Initial efforts to bridge QA between modalities focused on supplementing limited KG coverage by extracting relational information from more abundant texts. For instance, (Fader et al., 2014; Das et al., 2017) leveraged both structured (KB, tables, lists etc) and unstructured (text) information and used information extraction methods such as OpenIE (Banko et al., 2007) and UniversalSchema (Yao et al., 2012) so as to employ semantic parsing- or rules-based KGQA methods. More recent work instead casts structured information as text to access their knowledge through textual QA methods. (Agarwal et al., 2021) constructed KELM as a verbalisation of a large KG (Wikidata) to add to a retrieval LM corpus, obtaining performance improvements on benchmark QA datasets. (Oguz et al., 2022) obtain improvements by adding Wikipedia tables and lists to the data mix; they also propose a unified representation of structured information in a text-like format so that they may take better advantage of pretrained LMs and LLMs that are trained over large amounts of text. We take a similar approach (Oguz et al., 2022)'s in our work (see Section 2.3.3).

A closely related but distinct task is the use of inputs from more than one modality to help with QG or QA (Wang and Baraniuk, 2023; Talmor et al., 2021; Lehmann et al., 2024). IBM's DeepQA ODQA system (Ferrucci et al., 2010) is one such example of this for QA; it has gained

prominence for beating human champions in the long-running Jeopardy! quiz show. In QG, recent work by (Dong et al., 2024) proposed a GNN-based architecture that seeks to leverage information from both text and graph to improve the quality of questions generated from data in WebQuestions (KGQG) and SQuAD (TextQG). While useful for improving either the QG or the QA task individually, such approaches are not suitable for certain downstream applications such as fact verification, NLG data-to-text evaluation and tools to aid knowledge graph completion where it is important to establish what can be asked/answered specifically by the information in each modality.

2.7 Multilingual QG/QA

Another area of interest in this thesis is the extension of multimodal QG (and QA) into languages other than English (Chapter 5). Being able to do so helps enable the equitable availability of NLP tools for populations across the world by allowing access to such tools for speakers of other languages. Moreover, models that can generate and answer questions in local languages cater better to the information needs of users. It can also be useful for extending QA-based evaluations (Section 2.5) into these other languages and provide a means for helping to automate the verification of generated information.

A key limitation to having such multilingual QG and QA models is the lack of non-English data for training, which in turn also limits the availability of models for these languages. One important source of data for KGQA came from the Question Answering over Linked Data (QALD)[20] series of annual challenges, but even there the data available numbers in the hundreds[21] and insufficient for training neural network-based models. Additionally, since open KGs such as Wikidata remain English-centric, the setting we examine has to involve cross-linguality when carrying out QG and QA in the graph modality. For QA over text, it was only very recently that annotated multilingual QA/QG datasets (Lewis et al., 2020b; Artetxe et al., 2020) became available and even then, these datasets were limited in size (and coverage) and mainly useful for evaluating models only. Furthermore, none of these QA/QG datasets have alignment of QA pairs across modalities, as well as across languages (besides QALD).

Due to these data limitations, efforts for multilingual KGQA were mainly based on QALD data and started with rules-based methods using grammars capable of parsing for multiple languages (Zimina et al., 2018; Perevalov et al., 2023; Pellissier Tanon et al., 2018). Another early approach added an off-the-shelf (grammar-based) machine translation model to translate NL questions from other languages into English before using available models for KGQA in English to answer the question (Ahn et al., 2004). The use of MT, together with advances from neural network-based models, continues to enable KGQA over unseen languages (Perevalov et al., 2022), and we leverage it in our work in Chapter 5 as a means to augment the available Russian and Brazilian Portuguese text-graph aligned data (Section 5.5) for use.

Similar to the advances in QG and QA in English due to the capabilities of pretrained generative LMs (Section 2.2); recent work – including ours in Chapter 5 use similar multilingual models such as mT5 (Xue et al., 2021) as well as mT0 and BLOOMz (Muennighoff et al., 2023) to address the task. Using a similar approach as us in Chapter 5 (Zhang et al., 2023a) verbalise a multilingual KGQA dataset to utilise textual QA methods for KGQA; their work does not, however, examine QG, nor cross-modal QA over text and graph.

[20] https://qald.aksw.org

[21] The ninth edition of the challenge (Usbeck et al., 2018) provided training data over 11 languages but these were limited to 408 question-answer pairs in each language.

On the other hand, QG across languages is a relatively new area of study. To alleviate the issue of data scarcity, (Riabi et al., 2021) also used a combination of synthetic QG[22] and MT to obtain QA pairs for training cross-lingual QA (i.e. the question and context are in different languages). However their use of SQuAD and similar datasets (with one question per context passage) does not provide wide-coverage QG. In an effort to address this limitation (as well as for more efficient computation), (Ushio et al., 2022, 2023a,b) explored paragraph-level and multilingual QAG methods in a series of work, but they train only on SQuAD data and acknowledged that their models are "fine-tuned on questions that require one-hop reasoning only, so they are unable to generate multi-hop reasoning questions" (Ushio et al., 2023b).

2.8 Data

Since QG is an inverse task of QA (and vice-versa), datasets originally constructed for QA can be repurposed for QG with relative ease, and the same applies vice-versa. QG models and approaches (as described above), have been proposed using benchmark QA datasets such as SQuAD (Rajpurkar et al., 2016) for training. On the KG side, datasets from the long-running LC-QuAD (Dubey et al., 2019) and, as mentioned above, QALD (Usbeck et al., 2023) KGQA challenges have also been used for QG.

2.8.1 QG/QA data for KG

As such, our KGQG work in Chapter 3 leverages the **SimpleQuestions** (Bordes et al., 2015) dataset that was originally intended for KGQA and became accepted as a benchmark of KGQG models. SimpleQuestions is a set of 108K natural language QA pairs that were collected by crowdsourcing. It is based on one-hop triples from the Freebase KG; the questions were collected by showing crowd workers a single KG triple and asking them to write a question that mentions the subject and the relation of the triple and which is asks for the object of the triple as the answer.

Although SimpleQuestions is meaningfully large enough as a starting point for training neural-based KGQA/KGQA models, the Freebase KG it is based on has not been actively maintained or updated since the early 2010s. To update it and increase the diversity of available KGQG training data, we collected additional NL questions on the one-hop KG triples in the **WebNLG** dataset where the semantic content of text and graph are aligned. We did this through crowdsourcing and asked annotators to write questions asking for the subjects of the triples as answers, as well as questions doing the same for the objects. WebNLG is the eponymous dataset used in a series of NLG challenges that started in 2017. The initial version[23] contains 9,674 sets of RDF triples that are paired with crowd-sourced texts in English that lexicalise the triplesets. Each tripleset is paired with between three and seven texts, giving a total of 25,298 texts.

At the same time, we also converted the **ZeroshotRE** (Levy et al., 2017) KGQA corpus for use as KGQG training data. This dataset – part of the KILT benchmark (Petroni et al., 2021) that was intended to evaluate pretrained language models' ability to answer knowledge-intensive tasks, is made up of about 160,000 questions that were generated by slot-filling. The questions were instantiated from 1,192 crowdsourced question templates (e.g., "*Where did X*

[22]Using a QG model trained on SQuAD to generate new questions on texts from a different dataset.

[23]An updated and enlarged version of the WebNLG dataset has been released and was used in subsequent editions of the WebNLG challenge (Castro Ferreira et al., 2020).

graduate from?", "*In which university did X study?*" and "*What is X's alma mater?*") for 120 Wikidata properties.

2.8.2 QG/QA data aligned across modalities

As we noted in Section 2.6, there is, however, no dataset that is closely aligned for questions, answers and context over both text and KG, which is vital for work on our aims for cross-modal Text-KGQG and KGQA in Chapters 4-5. While some work has used distant supervision to pair KG graphs with text scrapped from the Internet, the approach is approximate and often results in texts that either add, omit or alter the information contained in the KG graph they are supposed to serve as lexicalisations of. To resolve this, in our work, we leverage the WebNLG dataset where the texts are human-written (collected via crowdsourcing) to be aligned with the information in KG graphs.

Synthetic data Other work has investigated the use of synthetically-generated data with round trip filtering techniques and shown improved textual QA performance (Alberti et al., 2019; Puri et al., 2020; Kwiatkowski et al., 2019). (Riabi et al., 2021; Agrawal et al., 2023) have also examined multilingual synthetic QA/QG data generation. Similarly, we used data augmentation and round trip filtering to improve generalisation; however, these other works are aimed at improving textual QA only, unlike ours, which aims to improve multilingual QG and QA jointly, which is also cross-modal for text and graph while also ensuring wide QG coverage.

While WebNLG is of meaningful size, it only covers a limited set of entities and relations. The latter is especially important as models often struggle to plug the semantic gap between KG relation labels and natural language. We therefore include the **KELM** (Agarwal et al., 2021), which originally consists of 15 million (Wikidata KG graphs, English texts) pairs. The English texts were synthetically generated from the Wikidata graphs using a T5 (Raffel et al., 2020) pre-trained model that was itself fine-tuned on TekGen, a large dataset of (graph, text) pairs that was created using distant supervision. By using off-the-shelf Text QG and QA models to generate and answer questions on the texts in WebNLG and KELM together with filters such as round-trip filtering (Alberti et al., 2019), we obtain a QG/QA dataset with silver-quality alignments between questions, answers, texts and graphs. Details on how we do so can be found in Section 4.2.

2.8.3 Aligned QG/QA data outside of English

On top of the QA datasets (which can be repurposed for QG) mentioned in Section 2.7, the QG-Bench (Ushio et al., 2022) dataset has been released for use as a benchmark dataset for evaluating paragraph-level (i.e. generating multiple questions from a single context paragraph) multilingual and multidomain QG from text. All of these datasets, however, do not possess all the needed components that are aligned between them (i.e. non-English text that is aligned with graph, questions and answers over the text and graph).

On the other hand, WebNLG has been expanded into other languages; besides English, WebNLG has been translated into Brazilian Portuguese (Almeida Costa et al., 2020), German (Castro Ferreira et al., 2018), Russian (Shimorina et al., 2019) as well four low-resourced languages Breton, Irish, Maltese, Welsh (Cripwell et al., 2023). Therefore, we leveraged the Brazilian Portuguese and Russian versions of WebNLG to generate multilingual cross-modal QG/QA data to use for evaluations in our study in Chapter 5. Since WebNLG only contains pairings of text and KG graphs, in that work, we rely on machine translation of WebNLG and KELM together

with off-the-shelf QG and QA models to produce silver-quality question, answer, text and graph instances for training our cross-modal cross-lingual models.

2.9 Evaluation

2.9.1 Evaluation for QA

Evaluation for QA is focused on assessing whether a model can identify or produce the correct answer given the question and context.[24] In our first work (Chapter 3), we use **Accuracy@1** when we carry out downstream evaluation for QG using QA (Section 2.9.2). This is a measure that is used in KG embedding-based KGQA systems (Section 3.5.4) that return a list of possible candidate answers. Accuracy@1 is strict as it only counts the question correctly answered if the reference answer is predicted as the most probable answer candidate by the KGQA system.

Two other metrics, **exact match** and **token F1**, have also traditionally been used in QA evaluation (Jurafsky and Martin, 2024). Exact match is the percentage of answers produced by a QA system that exactly matches the gold reference answer in a test dataset, whereas token F1 is the average amount of token overlap between the predicted and gold reference answer for each of the questions in a test dataset. These two metrics were, however, developed at a time when QA technologies were primarily extractive and carried out in one modality, for e.g. QA datasets (e.g. SQuAD (Rajpurkar et al., 2016)) were constructed by having human annotators select spans in the text as the answers to questions.

In recent years, QA approaches have evolved towards returning generated answers as models became increasingly based on pretrained/large language models. In the generative case, models often give answers to questions that are similar to the gold reference answer and, therefore, semantically correct. However, the generated answer may not be found exactly in the text (e.g "*Joe Biden*" instead of "*Joseph R. Biden*"), and therefore shares little surface similarity with the gold reference answer, which results in a lower EM and token F1 scores that do not accurately reflect the model's performance. Similarly, in the cross-modal case, the way in which an answer is lexicalised in text is often different from how it has been captured in the KG label. As such, the use of contextual similarity-based metrics such as BERTScore has been extended to evaluate QA systems; in our two subsequent works on cross-modal QG, our approaches rely on generative QA (Fusion-in-Decoder and MGEN, see Sections 4.5.3 & 5.7.5) for both text and KG modalities and we use BERTScore for evaluating the correctness of the answers generated.[25]

2.9.2 Evaluation for QG

Evaluation for QG, on the other hand, is focused at a fundamental level on three main aspects of the question: (i) its **answerability** using the input context it was conditioned on, and if the answer was part of the input, that it is appropriate for the answer; (ii) its **semantic adequacy** – in the case of KQGG, whether it appropriately verbalises the information in the KG input, and in text, whether it reflects the information in the input text and does not add or omit information; and (iii) its **fluency and grammaticality** i.e. whether the question is syntactically well-formed and sounds natural. In this section, we focus mostly on describing the more complex evaluation protocol required for QG.

[24] In the case of open-domain QA, which is not the focus of our work, the context is not provided.

[25] We do measure the exact match and token F1 scores nonetheless as a point of verification and report these in the appendices, see Appendix C.

There are, however, different aspects of QG, and depending on the research focus, a more specific set of evaluation criteria is called for. As an example, efforts focused on generating challenging multi-choice questions require not only that the question can be answered by the context, but that the answer candidates chosen must be plausible and potentially semantically similar amongst them (so as to serve as strong confounds). For our work on KGQG (Chapter 3, we were also interested in the ability to generate a diverse set of questions in a controllable manner for a given one-hop triple, and that therefore necessitated evaluating how different a generated question is from the reference question in our tests sets. For our work on multi-modal QG for improving QG/QA-based evaluation of data-to-text generation, an important objective is to generate questions that ask about as many parts of the input text (or KG graph) as possible; therefore, it necessitated evaluating the semantic coverage of the generated questions. Therefore in Chapter 4, to verify that our models were generating complex questions as intended, we also include this aspect in our evaluation protocol for that work.

Some efforts (Gollapalli and Ng, 2022; Wang et al., 2022; Mohammadshahi et al., 2023) have been made to develop **specialised automatic metrics for QG** that provide a single numerical score which captures different aspects expected of good QG; these efforts however often involve complex processing or workarounds and have yet to reach the ability to produce judgements of QG quality that is close to par with human judgements. To account for the answerability of generated questions (that standard surface metrics such as BLEU do not account for), initial work by (Nema and Khapra, 2018) proposed QBLEU by adding to BLEU an answerability measure that is approximated by the question having "*relevant content words, named entities and question types and function words*". A learnable weight, tuned on the dataset and tailored for the different QG tasks of KGQG, text QG and visual QG, is then used to combine the surface-based metric with the answerability score.

Recent work on a reference-free metric for QG by (Mohammadshahi et al., 2023) use a question-answering model (UnifiedQA-v2 (Khashabi et al., 2022)) to first extract an answer span for the question from the context, then a pre-trained RoBERTa encoder that is tuned to predicts a score of between one and five for the answer span. They report that their metric **RQGUE**, which, while surpassing previous work, achieves less than 0.5 correlation (i.e. demonstrating only correlation that is moderate or less) (Spearman rho and Kendall tau) or less with human judgements across aspects of grammaticality, answerability and relevance.

As such, a viable evaluation strategy for QG to derive meaningful signal on the quality of the questions generated could be to use a multi-prong evaluation strategy that directly assesses the aspect we wish to evaluate. For instance, in our studies, we use combinations of surface-based and contextualised embedding metrics to assess the similarity of the generated question to a reference, downstream QA to assess the answerability of the question and, where feasible, human evaluation to further assess the naturalness and semantic adequacy of our generated questions. Finally, our work in the last two chapters of this thesis (Chapters 4 & 5) was intended to improve QG/QA-based reference-less evaluations of data-to-text outputs. The evaluation for NLG evaluation metrics is usually based on computing how well a proposed metric correlates with human judgements for the aspects the metric is designed for (e.g. semantic adequacy, fluency, etc).

It is established practice in the many NLP fields to use automatic measures to assess the quality of decisions or generations produced automatically, and this applies to NLG as well. While human evaluations continue to serve as the gold standard for evaluating generated text, the resources required for executing such evaluations over large amounts of generated text and across multiple models are prohibitive. Furthermore, it has been shown that, in the face of the continuously improving quality of machine-generated text, there are limitations to using human

evaluation protocols to assess the quality of such texts. In their study, (Clark et al., 2021) found that human evaluators performed poorer than random chance (i.e. 50%) when asked to choose which of a human-written text and one generated from a large language model (GPT-3 (Brown et al., 2020)) was written by a human. In our work, we use human evaluations for QG where meaningful, and use it as a complement to automatic metrics to obtain a more comprehensive assessment of the quality of the generated questions. In Chapters 4 & 5, we use a form of cyclic QA consistency measure to derive confident support that our approach leads to improvement over a baseline. In the following paragraphs, we introduce the automatic metric and human evaluation protocols we use in our works.

Automatic metrics

BLEU (Papineni et al., 2002) is an established and widely-used metric within the NLG as well as machine translation (MT) field to assess if the quality of a generated (or translated) sentence is of good quality. It is based on string matching and approximates the semantic quality of a generated text by assessing how much of its words are present against that of a reference sentence, i.e. the precision of generated text. As the metric was initially designed for MT systems, where early models often generated repeated words, the BLEU metric computes a "*modified unigram precision*" where precision count is capped at the "*total count of each candidate word by its maximum reference count*". BLEU can be measured at different n-gram levels from one to four, with the intuition that matches at higher n-gram levels provide a stronger signal about the semantic correspondence between the generated text and the reference sentence. It is common to report the 4-gram (BLEU-4) count when comparing the results of different systems, and we have done so in our work as well. While the BLEU authors carried out tests conducted that showed it to correlate well to human judgements of generation quality, there are, however, limitations in relying on n-gram overlaps to determine the quality of a candidate as it is often unable to assign a high score to a perfectly acceptable paraphrase of the reference sentence (Zhou et al., 2006). We ameliorate these by using a suite of other surface-based automatic measures as well as a contextualised similarity-based measure, which we describe below.

Another similar metric, **ROUGE** (Lin, 2004), was initially developed for automatically assessing the quality of machine-generated summaries of text but has since been adopted for use in evaluating other NLG tasks. Similar to BLEU, it is based on string matching and seeks to identify how much of the generated text can be found in the reference text(s); although ROUGE differs from BLEU in that it is the ratio of matched n-grams divided by the total sum of the number of n-grams appearing in the reference(s). This makes it a recall-oriented metric and a suitable complement to BLEU's focus on precision. Like BLEU, ROUGE can be measured at different n-gram levels, and there is also a form that computes ROUGE, the longest common subsequence between generated text and reference. The latter (ROUGE-L) permits flexibility in matching, especially when there is paraphrasing (i.e. a smaller n-gram span could reasonably fall within another larger n-gram span and would have captured similar semantics) instead of a strict n-gram-level match. When using ROUGE in our work, we report the ROUGE-L form.

METEOR (Banerjee and Lavie, 2005) is another surface-based automatic metric; however, it seeks to go beyond simple n-gram matching in BLEU and ROUGE by taking into consideration paraphrases. The metric aligns the words (using exact match, stem, synonyms and paraphrase matching) against the words found in a reference text. In doing so, the METEOR metric more closely measures the quality of text in light of variability that can be found in human-written text.

BERTScore (Zhang et al., 2020) provides an alternative to surface-based automatic evalua-

tion metrics (BLEU, METEOR and ROUGE); it embeds candidate sentences with a pre-trained BERT (Devlin et al., 2019) model (or similar models[26]) and uses the hidden representation from the model at a given layer to obtain a contextualised representation of the candidate sentence. This is done by producing token alignments – in a greedy manner to maximise the cosine similarity, between the generated text and the reference. The authors report a strong correlation between BERTScore assessments of semantic similarity compared with human judgements, which has been broadly verified by subsequent investigations (Hanna and Bojar, 2021; Xiao et al., 2023). As a result, the use of BERTScore has become widespread in NLG. We use BERTScore in all of our work in Chapters 3-5.

Human evaluation

QG typically shares a similar human evaluation protocol with those used for other NLG tasks (see (van der Lee et al., 2019) for a position paper on best practices for NLG evaluation). It usually involves multiple human annotators (either recruited for their expertise and/or familiarity with the task being evaluated or through a larger pool of crowd annotators). Evaluation involves showing the raters the generated question and soliciting a judgement for it for the aspects of interest (see above). Depending on the task set-up and the QG aspects being evaluated, the input context and the answer may be shown; generated questions from two or more systems may also be presented to the rater at the same time. Evaluation is usually done with respect to the generated questions from a baseline or a competitive QG system and the judgements solicited from the raters are for gauging whether a proposed model or approach leads to better QG performance for the aspects being evaluated. In some cases, as we do in Chapter 3, annotators are also shown a reference question that is available in the dataset and asked to judge the generated question with respect to this reference. In our case, this was because we wished to understand the ability of our proposed approach to controllably generate questions that are diverse (i.e. different from the reference).

These judgements may be collected in a number of ways. Raters could be asked to give a numerical score to a generated question, e.g. using a Likert scale with integer values or on a sliding scale with real numbers, with the scores being averaged over all the questions from a given system and compared to the same for another system to identify the better-performing system. They could also be presented with the output questions for two or more systems and asked to express their preference for one; with the system selected the most often deemed the better performing. NLG systems evaluation with human judgements is prone to their own set of limitations, most notably bias that may arise from annotators (variance in annotator judgements, rating fatigue, annotator inconsistencies) or the collection protocol. Recommendations (e.g. (Howcroft et al., 2020; Schoch et al., 2020)) have been made and are typically taken in the collection procedures to mitigate these, some of which measures include recruiting qualified raters, presenting the NLG output to them in an anonymised and randomised order and providing neutral prompts and example ratings.

Downstream evaluation

Besides using automatic metrics to compare generated text against a reference and human evaluation, another evaluation method to assess the quality of QG is to carry out **downstream evaluation**. By using question answering with QA/MRC models on the generated questions and taking their level of answerability, we obtain useful signal about the quality of the generated

[26]See the spreadsheet on `https://github.com/Tiiiger/bert_score` for a list of models.

questions (i.e. if the question can be answered by the QA/MRC model, it informs us that the generated question quite likely contains sufficient information relevant to obtain the answer). We use this approach in all of our studies in Chapters 3-5.

3

Generating Questions from Wikidata Triples

Sommaire

3.1 Introduction . . . 32
3.1.1 Task and Terminology . . . 33
3.2 The WkdQG Dataset . . . 34
3.2.1 (RDF, Question) Datasets . . . 34
3.2.2 Adding typing and lexicalisation information . . . 36
3.3 Approach . . . 37
3.4 Experiments . . . 38
3.4.1 Training Details . . . 38
3.4.2 Elsahar's Model. . . . 38
3.4.3 Automatic Evaluation . . . 38
3.4.4 Human Evaluation . . . 39
3.5 Results and Discussion . . . 39
3.5.1 With and without additional NL information on Seen data . . . 39
3.5.2 Zero-Shot Learning . . . 40
3.5.3 Additional data . . . 40
3.5.4 Downstream QA Evaluation . . . 43
3.5.5 Ablation . . . 44
3.6 Conclusion . . . 44

To handle rare entities and generalise to unseen properties, previous work on KGQG resorted to extensive, often ad-hoc pre- and post-processing of the input triple. In this chapter, we revisit KGQG – using pretraining, a new (triple, question) dataset and taking question type into account – and show that our approach outperforms previous work both in a standard and in a zero-shot setting. We also show that the extended KGQG dataset (also helpful for knowledge graphs question answering) we provide allows not only for better coverage in terms of knowledge graphs (KB) properties but also for increased output variability in that it permits the generation of multiple questions from the same KG triple. Our code and dataset can be found at: https://gitlab.inria.fr/hankelvin/wikidataqg

This chapter is structured as follows: we start with an overview of the work in Section 3.1, the relevance of the simple QG from KG task, as well as, the reasons for revisiting it and for it to be controllable. We then go on to describe the data we used for our experiments in Section 3.2, followed by the description of our approach in Section 3.3. In Section 3.4 we describe our experimental set-up, which is followed in & Section 3.5 with the results we obtained and an analysis of the results. Finally, in Section 3.6, we conclude with a summary of our key findings.

3.1 Introduction

Question generation from knowledge graphs (or knowledge graphs question generation, KGQG) is the task of generating questions from structured database information, typically in the form of triples representing facts. RDF (Rich Description Framework, (Lassila and Swick, 1999)) is a semantic web standard for encoding knowledge. In an RDF KB, facts are encoded as triples of the form (s, p, o), where s and o are RDF entities (also called resources), and p is a property.

With the rise of large-scale knowledge graphs (KGs) such as Wikidata (Vrandečić and Krötzsch, 2014), DBpedia (Auer et al., 2007), and Cyc (Lenat and Guha, 1993), large amounts of factual data have become available, which can be used to answer factual questions. In that context, teaching machines to generate a question from a KG item (question generation from KB, KGQG) has become an important issue with multiple potential applications. By translating a KG fact (e.g., ⟨ HENRY_POINCARÉ , BIRTHPLACE , FRANCE ⟩) into a natural language (NL) question (e.g., *Where was Henri Poincaré born?*), KGQG facilitates access to KGs by non-experts. It could help improve the ability of dialogue models to ask factual questions and support the development of tutoring systems that ask the user a series of questions about some KG entity. Finally, it is useful for creating or augmenting the sets of (KG content, NL question) pairs necessary to train Question Answering (QA) systems on KGs (KGQA). However the scale of these knowledge graphs, the high number of rare entities they contain and the lack of NL aliases for KG relations still leave this task a challenging problem.

The state of the art in KGQG has mainly focused on how to address these rare entity and unknown relation issues. Typically, the KG input is enriched with lexicalisation information extracted from the KG (semantic type of the entities, domain and range of the relations) or/and using distant supervision from comparable KB/NL data (Elsahar et al., 2018; Liu et al., 2019a; Serban et al., 2016). Delexicalisation has also been commonly used, where KG entities are replaced with placeholders both in the input and in the output text (see table A.2). The model is trained on the delexicalised data, and at inference time, post-processing replaces placeholders with the corresponding values (Elsahar et al., 2018; Liu et al., 2019a; Serban et al., 2016).

Yet even if these approaches have yielded good results, they require extensive, often ad-hoc, pre- and post-processing techniques to be effective, increasing the complexity of the model. Moreover, these additional methods might not be generic enough to scale up to new databases with other schema and broader signature. Delexicalisation for instance, which requires matching KG entities (e.g., Barack Obama) in the input with their corresponding NL mentions in the output text (e.g., the former President of the United States) may be quite complex and may also result in incorrect or incomplete delexicalisations when applied to a new KB. Similarly, distant supervision is only possible given some comparable data and might only provide partial information. In fact, (Liu et al., 2019a) notes that (Elsahar et al., 2018)'s distant supervision approach only provides textual information for 44% of the predicates present in the SimpleQuestion dataset they use for training. Finally, the presence and coverage of type, domain and range information that are relevant for the generation of NL questions vary depending on the database and might

not be sufficient to support the verbalisation of unknown entities or relations (i.e., entities and relations which have not been seen at training time).

1.	**Input**	rdf: 〈 HENRI_POINCARE , BIRTHPLACE , FRANCE 〉	qfocus-pos: obj qtype: which
	Output	Which country was Henri Poincaré born in ?	
2.	**Input**	rdf: 〈 HENRI_POINCARE , BIRTHPLACE , FRANCE 〉	qfocus-pos: obj qtype: where
	Output	Where was Henri Poincaré born ?	

FIG. 3.1: Input/Output Examples: 1 and 2 show how the same input triple may map to multiple questions with different question types.

From around 2019, pretraining and fine-tuning have been shown to be effective for providing neural models with additional information about the structure of natural language and improving generative tasks (Dong et al., 2019; Song et al., 2019; Lawrence et al., 2019). In this chapter, we leverage pretraining to provide a model for KGQG, which requires neither delexicalising the training and test data nor enriching the KG input with additional information. We use BART (Lewis et al., 2020a), a Transformer-based encoder-decoder pretrained using a denoising objective on large quantities of text, and we propose an approach to the KGQG task which differs from previous work in two ways. First, we use the question type (e.g., *what, which, where, when, etc.* see Section 3.2.2) to guide generation. This helps capture the fact that, as illustrated by Examples 1 and 2 in figure 3.1, a given KG fact may give rise to multiple questions. Second, we provide a novel dataset for KGQG using Wikidata (Vrandečić and Krötzsch, 2014) as a KG and deriving (KG fact, question) pairs from three existing datasets, namely, SimpleQuestions (Bordes et al., 2015), ZeroShotRE (Levy et al., 2017) and WEBNLG (Gardent et al., 2017a). This novel dataset is both more up-to-date (replacing Freebase, which is no longer available with Wikidata, which has become one of the largest and most prominent collections of open data on the web) and linguistically richer (contrary to the SimpleQuestions dataset, which maps each input to a single question, the dataset derived from the WEBNLG data allows for a one-to-many input/question mapping).

We show that our approach outperforms previous approaches in a zero-shot setting for KG properties and entity types (i.e., for KG facts whose property/entity type does not occur in the training data); that additional data increases coverage (more KG properties can be accounted for); and that controlling generation using question type helps improve diversity (one KG triple can be used to produce multiple questions).

3.1.1 Task and Terminology

Given an RDF triple of the form (s, p, o), the KGQG task consists of generating an NL question about the object o (or the subject s) of the triple. Some examples of input and output are shown in figure 3.1.

We call the questioned part of the RDF triple the **question focus**, and we refer to the semantic type of the question focus, which can be extracted from the KG as the **question focus type**. We use the term **question focus position** to refer to the position (subject or object) of the question focus in the RDF triple.

3.2 The WKDQG Dataset

To evaluate our approach using the BART model, we created WKDQG, which is the compilation of three KGQG datasets: SimpleQuestions (Bordes et al., 2015), ZeroshotRE (Levy et al., 2017) and WebNLG-Q. Its creation was motivated by the lack of standard between the three datasets, since each one verbalises natural questions based on different knowledge graphs, some not even maintained anymore. To unify these datasets, WKDQG maps their knowledge graphs and aligns their natural questions to the Wikidata format (Vrandečić and Krötzsch, 2014), currently one of the largest and most prominent collections of open data on the web. Moreover, we also enrich the standardised datasets with additional typing and lexicalisation information that is not present in the original versions to allow comparison with previous approaches.

In this section, we first describe the creation and content of these three datasets and how they were mapped to Wikidata. We then explain how these datasets were enriched with additional typing and lexicalisation information to help guide generation.

	# (RDF,Q) pairs	**# qtype/RDF** (avg/min/max)	**# RDF** prop.	**# RDF** ent.
SQ	53,624	1.0/1.0/2.0	196	62,088
WQ	10,272	2.26/1.0/4.0	184	2,713
ZQ	282,543	1.22/1.0/4.0	120	193,576
TOTAL (union)	346,439	1.19/1.0/4.0	342	247,720

TAB. 3.1: Datasets statistics - 1.

	S/O/S&O	**Vocab.** Size	**Question Size** (avg/min/max)
SQ	0.77/0.20/0.03	16,602	7.96/1.0/36.0
WQ	0.11/0.71/0.18	6,084	10.08/5.0/42.0
ZQ	0.74/0.24/0.01	22,970	8.99/4.0/38.0
TOTAL (union)	0.75/0.22/0.03	30,607	8.86/1.0/42.0

TAB. 3.2: Dataset statistics - 2. (S/O/S&O denotes the proportion of entities occupying the subject, object or subject as well as object positions of the triples in the data. Vocab. Size is the number of distinct subword-tokens in the NL question part of the data, Question Size is the number of word tokens in each NL question.)

3.2.1 (RDF, Question) Datasets

SQ. We transformed the original SimpleQuestions (SQ) dataset (SQ-FB) into its Wikidata version (SQ hereafter) by mapping its entity pairs into Wikidata using the P646 property (Free-Base ID). We noticed that a small set of Freebase entities map to more than one entity in Wikidata; to resolve these subject entity ambiguities we used token overlap with the target question.[27]

[27]For instance, the Freebase entity 08fc1w is linked to two Wikidata entities, Q14798562 and Q153441 as at January 2022. They have entity labels 'Margarita Luti' and 'La fornarina', respectively. The entity Q153441 was selected given its label's uncased word token overlap with the question in the SQ sample: "*What artist created*

Next, we identified the set of all Wikidata properties between each entity pair. The entity pairs (in Wikidata format) are then grouped by their Freebase property (FB cluster). First, we consider all the one-to-one relations; if all entity pairs in a Freebase cluster with Freebase property p^i_{fb} share the same WKD property p^j_{wkd}, we map p^i_{fb} to p^j_{wkd}. If, on the other hand, not all pairs of the FB cluster are related by the same WKD property,[28] we check whether there exists a Wikidata property shared by at least 75% of the FB cluster. If this is the case, we assign this Wikidata property to all pairs of the FB cluster. Otherwise, we manually inspect the set of Wikidata properties associated with the FB cluster to decide whether to keep (assign one property to the group or from a set of properties for members of the group) or discard the found Wikidata properties.

This was done first in the forward direction, and then we repeated the process in the backward direction for the remaining unassigned Freebase entity pairs. Finally, for the remaining entity pairs without any Wikidata relations between them, we used the Freebase-to-Wikidata property mappings already found.

We note that in the original corpus, the question focus is always the object of the triple. When converting from Freebase to Wikidata, it sometimes shifted from being the object of the RDF triple to being the subject. We ensured that the relevant information in such samples are appropriately reversed. The final Wikidata RDF triple is represented by the English-language labels for the entities and property of the triple.

- **ZQ.** Although the ZeroshotRe dataset is already in the Wikidata format,[29] its test set comprises of only RDF triples, whose question focus is missing. For example, the question 'Which award did Hrant Melkumyan get?' is paired with the incomplete RDF triple 〈 HRANT MELKUMYAN , AWARD RECEIVED , _ 〉. To circumvent this, we used SPARQL queries to retrieve the answer set for these questions in Wikidata. For those with more than one possible answer, we instantiated new samples in the corpus.[30].

- **WQ.** The WEBNLG dataset (Gardent et al., 2017b) is another popular data-to-text benchmark, with natural language assertions to 3,790 unique RDF triples as well as their combinations. We collected a *data-to-question* version of this corpus, comprising 11,664 NL questions to 3,625 (95.6%) of the single RDF triples of version 2.1 of the original corpus.

The questions were collected using the AMT crowdworking platform. Participants were asked to produce a question for an RDF triple given a question type and a question focus to ensure wide coverage of such questions, which is our aim for WKDQG. In terms of question focus, they were given both subject and object parts of the RDF triples to ask for (e.g., 〈 **ALBENNIE_JONES** , ACTIVEYEARSENDYEAR , 1950 〉 → *Which singer closed out her career in 1950?* and 〈 ALBENNIE_JONES , ACTIVEYEARSENDYEAR , **1950** 〉 → *When did Albennie Jones' career come to a close?* respectively) .[31]

"La Fornarina"?"

[28] For example, in the Wikidata KB, the FB property `www.Freebase.com/music/artist/origin` sometimes map to the Wikidata relation "place of birth" (P19) and sometimes to "location of formation" (P740).

[29] We used the version of it that is included in the KILT benchmark (Petroni et al., 2021), publicly available in the HuggingFace `datasets` library.

[30] A small set of 217 of these incomplete RDFs (534 samples) remained without answers (from the time ZeroshotRE was created to present, the subject in the RDF no longer holds the property); they are marked "NoAnswer" in the dataset and excluded from our experiments

[31] The question focuses were classified using a coarse-grained set of KG types (Location, Organization, Person, and Event/Date, Measure, Number as well as an Other category) and mapped to the corresponding question types from *What, When, Where, Which, Who, How many.*

Question type	SQ	WQ	ZQ
What	58.3%	42.3%	57.7%
When	<0.1%	1.0%	3.0%
Where	9.6%	7.7%	1.6%
Which	14.1%	38.3%	20.8%
Who	13.1%	8.5%	15.5%
How many	-	2.2%	-
Other	4.9%	-	1.4%

TAB. 3.3: Distribution of question types in the SQ, ZQ and WQ datasets.

As the WEBNLG triples come from the DBpedia knowledge graphs, we mapped their entities and properties into Wikidata using a process similar to that for the SQ dataset. Details are given in the appendices. Due to differences in the data models of DBpedia and Wikidata, 412 out of the 3,625 WEBNLG original single triples could not be mapped into Wikidata as their properties have no or multiple counterparts in Wikidata. Another set of 90 single triples maps into a smaller set of 45 Wikidata triples.[32] As a result, only 10,272 RDF-Q pairs remained in WQ after the mapping.

3.2.2 Adding typing and lexicalisation information

For all datasets, we enrich each RDF-question pair with question type and the semantic type of its question focus. This information was obtained from the Wikidata public SPARQL endpoint in the second half of 2021. For SQ, we also include the additional information released by (Elsahar et al., 2018).

▪ **Question type.** SQ, WQ and ZQ contain *What, When, Where, Which and Who* questions, whereas WQ also contains quantity-seeking questions (e.g. 'How many pages is the novel A Long Long Way?'). We detect these question types with regular expressions/string match.[33] Besides these question types, SQ and ZQ contain 'inform-me' questions (e.g. 'The date of birth of Glyn Pardoe is? and 'Name a modern jazz singer.') and polar questions ('Is highly refined pirates a post-rock album?') as well; in our work, we label these questions as being of the *Other* type.

▪ **Question focus type.** For SQ, we use the Freebase entity type information contained in the version of the original dataset released by (Elsahar et al., 2018) (see Section 3.2.2) to allow for a comparison with Elsahar et al's model. For WQ and ZQ, we retrieved the set of Wikidata supertypes for every entity in the dataset from Wikidata using the 'instance of' (P31) and 'subclass of' (P279) properties. If an entity has multiple possible supertypes, we select the one that is the most common across the training split of the dataset for our experiments. Table 3.2 shows some statistics about each dataset, and Table 3.3 about the question type distribution.

[32]For instance the DBpedia triples ⟨ BACON EXPLOSION , MAIN INGREDIENT , BACON ⟩ and ⟨ BACON EXPLOSION , INGREDIENT , BACON ⟩ both map to ⟨ BACON EXPLOSION , HAS PART , BACON ⟩ in Wikidata.

[33]First with regular expressions at the start of the question, and if no question type word was detected there, a fallback to string match inside the question (for embedded questions, e.g. ''Name an artist who plays rock music.').

▪ **SQ additional information.** When comparing our approach with (Elsahar et al., 2018) on the SQ dataset, we use the additional lexical information they released. This comprises verbalisations of the RDF property, obtained by distant supervision, as well as the Freebase semantic type of the RDF entities.[34] For SQ instances without such additional information, we used a special token to represent the missing information.[35]

3.3 Approach

Instead of enriching the input with NL information as was done in previous work, we leverage advances in pretraining and use the WKDQG dataset to adapt the BART pretrained model to KGQG. BART (Lewis et al., 2020a) is a Transformer-based encoder-decoder using sub-word tokenisation (byte-pair encoding) and was trained using a generative denoising objective on a combination of news, Wikipedia and books.

In adapting BART for question generation from RDF input, we explore two main options: one where only the RDF is used as input (BART$_{rdf}$) and another (for the SQ dataset only) where the RDF input is enriched with the additional NL information provided as a support for the lexicalisation of the RDF content by (Elsahar et al., 2018) (BART$_{rdf+nl}$).

We also explore variants regarding the question type: (BART$_{rdf}$), (BART$_{rdf,qt}$) and (BART$_{rdf,mtl}$), which we describe below; Table A.2 in the appendices provide examples of the inputs provided to each of these models.

▪ BART$_{rdf}$. This is a BART model without modification, which takes as input an RDF triple and a token indicating the question focus position. The RDF is represented linearly in the (s, p, o) order with a special token separator ('|') between each of them.

▪ BART$_{rdf+nl}$. This is the same as BART$_{rdf}$ except that we enrich the input with lexicalisation information for the RDF property provided by (Elsahar et al., 2018).[36] We separate this additional information from the rest of the input using a special token.

▪ BART$_{rdf,qt}$. The input to BART$_{rdf,qt}$ is the concatenation of the input RDF triple, a token representing the question focus position, the semantic type of the question focus (e.g., musical artist, location) and a special control token for the target question type. We separated each of these four fields with markers in the input. The addition of the question type information is equivalent to an oracle setting when evaluating on the SQ test set. Our interest in it is motivated by its usefulness for generating varied questions (see Section 3.5.3).

▪ BART$_{rdf,mtl}$. BART$_{rdf,qt}$ requires that the type of the question that can be generated from an RDF triple be known (since the question type is part of its input). We also explore a setting where this requirement is lifted by using multi-task learning where QG is the main task and predicting the question type is an auxiliary task. The input to both tasks is the concatenation

[34] The semantic type information for the entities in SQ were obtained by (Elsahar et al., 2018) from the FB5M version of Freebase using the 'fb:type/instance' property.

[35] As indicated in Section 3.1, Elsahar's distant supervision approach only provides lexicalisation information for 44% of the predicates present in SQ.

[36] For e.g. the RDF ⟨ Adler School of Professional Psychology , country , United States of America ⟩ and type information for the question focus, we reused the lexicalisation "is location of" from (Elsahar et al., 2018) and inserted the entities in their correct argument position to give "*United States of America is location of Adler School of Professional Psychology*".

of the triple with the question focus position and the question focus type. The model is trained by minimizing the weighted sum of the loss from the main KGQG task and this auxiliary task. We found that a 0.3 weight for the auxiliary task and 0.7 for the QG task to perform best.

3.4 Experiments

3.4.1 Training Details

For all our experiments, we used the `bart-base` model from the HuggingFace `transformers` library. The `bart-base` model has six layers each in its Transformer encoder and decoder blocks with a hidden size of 768. We used the bart-base tokeniser with a vocabulary size of 50,282 tokens (having added special tokens for the RDF separator, question and question focus types). We fine-tune all of the models by minimising the standard cross-entropy loss of the outputs against the targets. We use the ADAMW optimiser with a learning rate of 0.0002 and a learning rate scheduler with a linear warm-up of 10% of the training steps. All of the `bart-base` models were trained for 10 epochs and tuned on the BLEU-4 score of the development set.

3.4.2 Elsahar's Model.

We compare our approach with the BART model with (Elsahar et al., 2018), an RNN-based encoder-decoder model with attention as well as delexicalisation. The model is trained with the delexicalised output, at inference time, the decoder output is re-lexicalised. In the `RDF-only` setting, only information about the RDF (in the form of TransE pretrained embeddings) is provided to the model. Word-based tokenisation was used, and word tokens were represented with pretrained 100-D GLOVE embeddings (Pennington et al., 2014b). For the RDF triple, pretrained Wikidata embeddings released by (Han et al., 2018)[37] were used to represent the elements of the triples.

In the `RDF+NL` setting, the additional lexicalisation information about the property and the entities (see Section 3.2.1) is added to the RDF input using separate encoders and GLOVE embeddings for the lexicalisation part of the input. We used the publicly released code by (Elsahar et al., 2018), making only changes to load the pretrained Wikidata KG embeddings and ensuring that their and our decoders are not constrained by a max length in order to allow comparability.

3.4.3 Automatic Evaluation

To evaluate the models' outputs, we used the BLEU-4 (Papineni et al., 2002), ROUGE-L (Lin, 2004), and METEOR (Denkowski and Lavie, 2014) automatic metrics. These n-gram-based measures are widely used as indicators of the surface similarity (BLEU-4 and ROUGE-L) and paraphrase (METEOR) between a model's generated output and a reference. We also report the BERTScore (Zhang et al., 2020) metric, which using a BERT model, provides an indicator of the embedding-based semantic similarity between output and reference. To ensure direct comparability between the outputs from the BART model (subword-based tokenisation) and Elsahar's (word-based), we applied the MOSES detokeniser on all the models' outputs and references, as well as a set of regular expression rules on Elsahar's model outputs to detokenise contractions and possessives (e.g. don't) not handled by the MOSES detokeniser, before scoring with the automatic metrics.

[37] http://139.129.163.161/index/toolkits#pretrained-embeddings

3.4.4 Human Evaluation

We also conducted a human evaluation to assess to what extent the question type control token and the variability present in the WQ dataset (one triple can be mapped to multiple questions with different question types) can help generate questions of different types from the same triple. In this evaluation, we compare our best model trained on SQ only with the same model trained on all three datasets (SQ, WQ, ZQ). Our hypothesis is that WQ variability will help the second model learn to generate different types of questions for the same triple. We collect 50 randomly selected input triples covering different properties from the output of both models. We show the annotators the input triple, the reference question and the output of the two models, and we ask them which of the two outputs verbalises the input property best (A: Adequacy); and differs most from the reference question (D:Difference); is most natural (N:Naturalness) and verbalises the entities best (E:Entity). A third option is possible for cases where both outputs score similarly. We measure the percentage of time one model was chosen over the other, taking the majority agreement between three evaluators.

3.5 Results and Discussion

We first compare our approach with Elsahar's on SQ considering four settings: with and without NL information, on seen data (RDF properties present in the test data have been seen at training time), and in a property zero-shot setting (RDF properties present in the test data have not been seen at training time). We also examine the seen and zero-shot settings for entity types. We then examine the impact of the additional datasets, in particular, the WQ dataset, which allows for the same triple to be mapped to multiple questions with different question types.

3.5.1 With and without additional NL information on Seen data

Table 3.4 shows the results for the standard setting, where RDF properties present in the test data have been seen at training time.

▪ **Pretraining helps bridge the gap between RDF properties and their lexicalisation.** Our approach outperforms Elsahar's in both settings (with and without additional lexicalisation information in the input). Moreover, BART$_{rdf}$, which uses no additional information, yields comparable results to Elsahar$_{nl}$'s model, which uses additional NL information. This indicates that pretraining and word pieces suffice to bridge the gap between the name of the RDF properties and the way they are lexicalised in text. It also shows that despite the high ratio of named entities in RDF triples (the subject and object, which make up two-thirds of an RDF triple, usually are named entities), delexicalisation (used by Elsahar's) can successfully be replaced by these two methods: both pretraining and word pieces help the model generate names that might not have been seen at training time.

▪ **Question type helps improve performance.** Whether it is implicitly learned through multi-task learning (BART$_{rdf,mtl}$, BART$_{rdf+nl,mtl}$) or explicitly input to the model (BART$_{rdf,qt}$, BART$_{rdf+nl,qt}$), informing the model with the target question type improves performance.

Model	B-4	BSc	R-L	M
RDF-only				
Elsahar	34.01	64.85	61.51	32.67
BART$_{rdf}$	37.05	69.42	65.12	34.22
BART$_{rdf,mtl}$	37.91	69.68	65.20	34.55
BART$_{rdf,qt}$	**41.95**	**73.51**	**71.21**	**36.78**
RDF+NL				
Elsahar$_{nl}$	38.13	68.63	65.48	34.74
BART$_{rdf+nl}$	38.38	70.00	65.67	34.87
BART$_{rdf+nl,mtl}$	38.10	70.17	65.57	34.73
BART$_{rdf+nl,qt}$	**42.67**	**73.78**	**71.50**	**37.28**

TAB. 3.4: Results on the SQ dataset under a **SEEN** setting, i.e. no zero-shot constraints (B-4: BLEU-4, BSc: BERTScore, R-L: Rouge-L and M: Meteor).

3.5.2 Zero-Shot Learning

We used the same cross-validation approach as (Elsahar et al., 2018) to approximate a zero-shot property setting. Specifically, we split the SQ dataset into 10 folds, with mutually exclusive sets of RDF properties (no fold contains RDF properties found in another fold) and draw two of these folds in turn for use as the test set in each cross-validation run. At the start of each run, we reload the original parameter weights for the pretrained BART model. Following Elsahar, we repeat the same zero-shot setting on entity types (which is stricter than a zero-shot entity set-up), taking care to account for the fact that a SimpleQuestions triple in Freebase may have a different order when mapped to Wikidata.

Table 3.5 includes the mean and standard deviation of the automatic metrics from cycling through these data splits for a zero-shot property setting. Table 3.6 shows the same but for the zero-shot entity type settings.

▪ **Pretraining outperforms a delexicalisation model whose input is enriched with lexicalisation information.** Regardless of a zero-shot property or zero-shot entity type setting, our approach outperforms Elsahar's whether or not the input is enriched with lexicalisation information. Notably, the BART model **without** NL information (BART$_{rdf}$) performs on par with Elsahar's model **with** NL information (Elsahar$_{nl}$). This illustrates the capacity of pretrained decoders based on subword units to handle unseen units: while the RDF properties of the test data have not been seen at training time, their subword units probably have been and can be used by the decoder to generate the corresponding NL expressions. There is, however, a 10-BLEU point gap between the zero-shot property and zero-shot entity type settings for the top-performing model (BART$_{rdf+nl,qt}$), indicating the importance of lexicalisation data for KG properties.

3.5.3 Additional data

The main contribution of the extended dataset WKDQG (in particular WQ) is that it helps generate multiple questions from the same KG triple. In SQ, each RDF is only paired with a question of a single type as well as a single question focus (either the subject or the object). Accordingly, a model trained (and evaluated to perform well) on SQ data alone is unlikely to be

Model	B-4	BSc	R-L	M
RDF-only				
Elsahar	14.24	47.94	44.30	24.43
	(±2.48)	(±2.23)	(±2.66)	(±0.92)
BART$_{rdf}$	22.63	59.12	53.14	27.02
	±2.85	±2.55	±2.64	±1.35
BART$_{rdf,mtl}$	26.63	62.24	56.52	28.91
	±3.03	±1.20	±2.21	±1.19
BART$_{rdf,qt}$	**28.35**	**64.40**	**60.84**	**30.46**
	±3.33	±2.98	±2.67	±1.62
RDF+NL				
Elsahar$_{nl}$	20.54	55.73	51.11	27.32
	(±3.68)	(±2.34)	(±2.96)	(±1.45)
BART$_{rdf+nl}$	23.42	59.52	53.74	27.46
	± 3.38	±2.66	±2.73	±1.39
BART$_{rdf+nl,mtl}$	25.25	61.29	55.04	28.27
	±3.35	±2.49	±2.42	±1.34
BART$_{rdf+nl,qt}$	**28.74**	**64.18**	**60.62**	**30.38**
	±3.38	±3.27	±2.90	±1.45

Tab. 3.5: Results on the SQ dataset under a **zero-shot** setting for RDF properties.

Model	**Sub-type**		**Obj-type**	
	B4	R-L	B4	R-L
RDF-only				
Elsahar	29.96	58.46	23.94	53.54
	(±2.10)	(±2.29)	(±4.34)	(±3.23)
BART$_{rdf}$	32.90	61.59	30.40	60.05
	(±1.90)	(±1.79)	(±3.09)	(±2.34)
BART$_{rdf,mtl}$	33.21	62.11	31.07	60.49
	(±1.50)	(±1.74)	(±2.65)	(±2.33)
BART$_{rdf,qt}$	**37.30**	**67.48**	**35.05**	**66.11**
	(±1.68)	(±1.38)	(±3.03)	(±1.97)
RDF+NL				
Elsahar$_{nl}$	32.92	61.43	28.58	58.42
	(±2.77)	(±1.91)	(±4.48)	(±2.98)
BART$_{rdf+nl}$	34.23	62.29	30.96	59.87
	(±2.33)	(±2.28)	(±3.27)	(±3.02)
BART$_{rdf+nl,mtl}$	34.32	62.31	31.24	60.13
	(±2.01)	(±1.98)	(±3.23)	(±2.55)
BART$_{rdf+nl,qt}$	**38.51**	**68.08**	**35.76**	**66.62**
	(±1.68)	(±1.51)	(±3.12)	(±2.13)

Tab. 3.6: Results on the SQ dataset under a **zero-shot** setting for RDF entities (subject/object for Elsahar, question focus and the other entity in the triple for the BART models).

able to generate questions of a different form (paraphrased or with a different question type). On the other hand, although WQ is five times smaller in size, it has a wide coverage of question types for each RDF in it and there is also variation in the question focus. While the questions in ZQ were instantiated from templates, and their question focus is all on the object of the RDF, all of its questions are of a high quality in terms of specificity. We included them in WKDQG for these reasons.

Using the additional parallel data created with WKDQG, we also explored different ways of combining it (training on all data or training and fine-tuning) but did not find it to improve results over training and testing on each of the three datasets (cf. Table A.1 in the appendices) likely because the datasets have very different properties in terms of question type/input ratio and vocabulary size.

Model | Metric	B-4	BSc	R-L	M
$Test_O$				
BART$_{rdf,qt}$	41.95	73.51	71.21	36.78
BART$_{rdf,qt,wkdqg}$	41.31	72.63	70.28	36.62
$Test_A$				
BART$_{rdf,qt}$	26.60	60.15	49.53	29.27
BART$_{rdf,qt,wkdqg}$	26.37	59.48	49.29	29.30

TAB. 3.7: Results on the SQ dataset under a **SEEN** setting. BART$_{rdf,qt,wkdqg}$: model fine-tuned on the WKDQG data. `Test`$_A$ (for alternative) is the SQ test set with a different question type provided to the model. `Test`$_O$ (for original) is the SQ test set with the original question type.

Choice | Measure	D	A	N	E
BART$_{rdf,qt}$Test$_0$	14%	12%	18%	2%
BART$_{rdf,qt,wkdqg}$Test$_A$	76%	4%	10%	4%
Same	8%	80%	62%	92%
No Majority Vote	2%	4%	10%	2%

TAB. 3.8: Human evaluation on 50 outputs (D:Difference from the reference, A:Semantically Adequate, N:Naturalness, E:Entity Lexicalisations). For each criterion, the first two lines of the columns indicate which model is preferred. E.g., BART$_{rdf,qt,wkdqg}$'s output is judged more different from the reference 76% of the time than BART$_{rdf,qt}$'s.

▪ **Finetuning with varied data permits generating questions with different question types from the same triple.** Using SQ as test set, we show that BART$_{rdf,qt}$ fine-tuned on the WKDQG data (BART$_{rdf,qt,wkdqg}$) is able to generate questions that are paraphrases of the reference. We do this by replacing the question type in the input with one for a different but semantically plausible question type.[38]

We found that in this setting (`Test`$_A$), BART$_{rdf,qt,wkdqg}$ is able to almost always faithfully generate a question of this new type: the model output only deviated from the provided question

[38]We used a BERT model to predict the distribution of question types given an entity type and a question focus position as input. This model is trained on data from the train and development sets of SQ, WQ and ZQ.

type seven times (out of the 10,725 instances in the test set).[39] Tables 3.7 (automatic evaluation) and 3.8 (human evaluation) show the results of this experiment, and Table 3.10 contain examples of the generated outputs here. While the automatic metrics show significantly lower scores for the $\texttt{Test}_A$ setting since the question type of the generated questions is different from the reference, the human evaluation indicates that the $\text{BART}_{rdf,qt,wkdqg}$ model generates questions with a comparable level of adequacy (A), naturalness (N) and entities (E) than in a $\texttt{Test}_O$ setting but a greater difference with the reference. The agreement among the three annotators was 0.521 (Fleiss' kappa). This demonstrates that by controlling for the question type and using training data with greater variability, KGQG models can be used to generate high-quality questions that differ from the reference.

Model	B-4	BSc	R-L	M
$\text{BART}_{rdf,qt}$	41.95	73.51	71.21	36.78
-question type	37.73	69.72	65.41	34.51
-question focus	41.43	73.17	70.81	36.54

TAB. 3.9: Ablation Study: each line indicates the (non-cumulative) removal of the corresponding component from $\text{BART}_{rdf,qt}$on the SQ dataset

3.5.4 Downstream QA Evaluation

We evaluated the utility of our generated varied questions on the performance of two downstream QA systems – KEQA (Huang et al., 2019) and BuboQA (Mohammed et al., 2018), leveraging the approach and code of (Han et al., 2020). While we generate questions using Wikidata triples to enrich the SQ dataset, all of the QA experiments described here use Freebase data. The results of these experiments are summarised in Table 3.11 here and details are provided in Table 3.12.

Using the same $\texttt{Test_}A$ set as in Section-3.5.3 while leaving the train and development sets unchanged, we show that simply changing the distribution of the question types at inference time results in a 3.5% drop in top-1 accuracy (see **SQ_w1** and **SQ_w2** in Table 3.12).

We were able to reverse this drop in performance on $\texttt{Test_}A$ by using an enriched training and development set. We do this using the same question type prediction model above,[40] and using the obtained set of plausible question type tokens as controls for the BART model, we generated the set of paraphrased questions of (different question types) for each SQ sample.

We also show that this enrichment approach enables robust QA performance – in the face of a shift in the distribution of the question types in the test data. The same models trained on the enriched training and validation set perform as well on the original test set (i.e. $\texttt{Test_}O$ in Section-3.5.3, compare (**SQ_o+e**:**SQ_o** and **SQ_w3**:**SQ_w0**). Consistent with the findings of (Liu et al., 2019a), we find that enriching the training data with generated questions leads to a minor decline ($\approx$ 0.5 percentage points) in top-1 accuracy.

[39] In all seven cases, the new question type provided to the model was 'Other', and the generated questions were of the 'What' (4) and 'Which' (1), and 'Where' (2) types.

[40] Except that, instead of picking a single question type for each sample, we returned the set of all possible question types capped at four (including the original).

	SimpleQ Sample	
1.	Reference	what category of celestial object is 7624 gluck
(O)	Input	⟨ 7624 GLUCK , INSTANCE OF , ASTEROID ⟩, **WHAT**, ANSOBJ, category
	Generated	what type of celestial object is 7624 gluck
(A)	Input	⟨ 7624 GLUCK , INSTANCE OF , ASTEROID ⟩, **WHICH**, ANSOBJ, category
	Generated	which type of celestial object is 7624 gluck
2.	Reference	what is maurizio calvesi's profession
(O)	Input	⟨ MAURIZIO CALVESI , OCCUPATION , CINEMATOGRAPHER ⟩, **WHAT**, ANSOBJ, profession
	Generated	what is maurizio calvesi's profession
(A)	Input	⟨ MAURIZIO CALVESI , OCCUPATION , CINEMATOGRAPHER ⟩, **OTHER**, ANSOBJ, profession
	Generated	is maurizio calvesi a cinematographer or a technician
3.	Reference	who was born in compton
(O)	Input	⟨ CLARENCE DUREN , PLACE OF BIRTH , COMPTON ⟩, **WHO**, ANSSUBJ, american football player
	Generated	who was born in compton, queens
(A)	Input	⟨ CLARENCE DUREN , PLACE OF BIRTH , COMPTON ⟩, **WHAT**, ANSSUBJ, american football player
	Generated	what former football player was born in compton, illinois

TAB. 3.10: Examples of the outputs of $\text{BART}_{rdf,qt}$(O) on Test_O compared with those of $\text{BART}_{rdf,qt,wkdqg}$ on Test_A for the same sample (except for a different question type provided to the model). In boldface are the question-type markers provided to the model.

3.5.5 Ablation

We also performed an ablation study using the SQ dataset (cf. table 3.9) and found that removing the question focus from the input has limited negative impact (-0.52 BLEU-4) but that removing the question type control token leads to a strong decrease in performance across all automatic metrics (-4.22 BLEU-4). This indicates that the benefits brought about by pretraining, through knowledge about the question focus embedded in the model, are limited relative to question type information, at least for the goal of generating questions faithful to the type in the reference.

3.6 Conclusion

To summarise, for the work in this chapter, we revisited the task of KGQG and introduced a novel approach of generating questions from KG triples using a fine-tuned large language model, without the ad-hoc processing and reliance on complicated-to-update KG embeddings required of prior work. Experimental evidence on WKDQG revealed that pretraining and question type control contribute to improved performance both in a standard and in a zero-shot setting. We investigated the capacity of the BART model and extended dataset to generate questions whose type is distinct from that of the gold truth, and found that it leads to better or similar quality questions of varied types in our human evaluation. Additionally, we quantified the decline in

Model	**OOO**	**OOA**	**EEA**	**EEO**
BuboQA Acc @ 1	**85.12**	81.42	85.08	84.57
KEQA Acc @ 1	**86.85**	81.15	83.79	86.44

Tab. 3.11: Results of QA systems with and without questions generated with our approach. The column headers denote the train-dev-test set compositions. **O** denotes original, **A** alternative and **E** enriched sets of questions.

Split/Model	**SQ_o**	**SQ_o+e**	**SQ_w0**
Train	O, (75,722)	O+E, (173,063)	O_w, (37,521)
Dev	O, (10,815)	O+E, (24,664)	O_w, (5,360)
Test	O, (21,687)	O, (21,687)	O_w, (10,726)
BuboQA Acc@1	74.63	74.03	**85.12**
KEQA Acc@1	75.30	74.76	**86.85**

Split/Model	**SQ_w1**	**SQ_w2**	**SQ_w3**
Train	O_w, (37,521)	O_w+E, (149,710)	O_w+E, (149,710)
Dev	O_w, (5,360)	O_w+E, (21,380)	O_w+E (21,380)
Test	O_w-A, (10,726)	O_w-A, (10,726)	O_w (10,726)
BuboQA Acc@1	81.42	85.08	84.57
KEQA Acc@1	81.15	83.79	86.44

Tab. 3.12: Results of QA system performance on SQ with and without generated varied questions in the train, dev and/or test sets. In the table above, **O** denotes the use of original SQ questions, **E** denotes enrichment (of O) with generated varied questions in train and dev, **A** denotes a question of an alternative (to O's) question type for each sample in the test set. **_w** denotes the set of SQ samples successfully mapped to Wikidata. In brackets are the sizes of the dataset splits.

current QA systems' performance at inference time when the question type distribution is shifted from that seen at training, and showed that by enriching the training data with the set of possible questions generated by our approach, these systems' performances are restored.

4

Generating and Answering Simple and Complex Questions from Text and from KGs

Sommaire

4.1 Introduction . . . 48
4.2 Method and data . . . 49
4.2.1 Associating Graphs and Texts with Questions and Answers (Step 1) . . 49
4.2.2 Training QG Models on Q-KELM (Step 2) . . . 50
4.2.3 Extending Q-WebNLG^0 (Step 3) . . . 52
4.3 QTT, a multimodal QG-QA Model . . . 52
4.4 Experiments . . . 55
4.4.1 Evaluation data . . . 55
4.4.2 Evaluating QG Coverage . . . 55
4.4.3 Evaluating QA Consistency . . . 55
4.4.4 Downstream: QA with FiD . . . 57
4.4.5 Downstream: Data-QuestEval metric . . . 57
4.4.6 Evaluation settings . . . 57
4.5 Results . . . 58
4.5.1 QG Coverage . . . 58
4.5.2 QA Consistency . . . 58
4.5.3 Downstream Evaluations . . . 60
4.5.4 Ablation . . . 60
4.5.5 Human evaluation . . . 62
4.5.6 Question generation examples . . . 63
4.6 Conclusion . . . 64

In the previous chapter, we focused on the controllable generation of simple questions using a single RDF triple as input. Seen alongside already existing methods for generating questions from either text or from KG, we could envision approaches that are able to generate questions on information that reside in either modality. Contemporaneously, methods (Scialom et al., 2021; Rebuffel et al., 2021) that combine and leverage both QG and QA systems were showing

promise for use to verify the contents of two inputs (notably for text summarisation and data-to-text generation). However, while both text and KG may be used to answer a question, most existing QA and QG models were designed to work only on a single modality and on either simple or complex questions, rarely both. For this chapter, we introduce a multi-task model that can generate and answer questions from both KG and text input modalities. The model has wide coverage and handles both simple (one KG fact) and complex (more than one KG fact) questions. Extensive internal, cross-modal and external consistency checks and analysis of the quality of the generated questions show that our approach outperforms previous work. Our data and modelling also lead to improvements in downstream tasks, including a better performance with fine-tuning Open-Domain QA architectures and better correlation with human judgments than the Data-QuestEval metric, which was previously proposed for evaluating the semantic adequacy of KG-to-Text generations. They, however, did not provide a systematic evaluation of their models; so we have provided a detailed evaluation of our model and compare it against theirs. We also examine how using our model – instead of (Rebuffel et al., 2021)'s original, impacts the Data-QuestEval metric's correlations with human judgments.

In this chapter, we outline rationales for building multimodal QG-QA in Section 4.1 and the key challenges to being able to do so. We then give in Section 4.2 a high-level overview of our proposed approach for the task, followed by a discussion on the data we used. Details of how we approach the modelling of the task are in Section 4.3. Section 4.4 contains details of our experimental set-up, including the baseline we compare with, as well as how we evaluate our model against it. This is followed by our results and analysis in Section 4.5, before we summarise our findings and highlight our conclusions in Section 4.6.

4.1 Introduction

Previous work on question generation (QG) and question-answering (QA) mainly focused on a single modality such as Natural Language (NL) (Lyu et al., 2021), Knowledge Graphs (KG) (Hu et al., 2019) or images (Shah et al., 2019). Although QG and QA should be able to operate consistently across semantically equivalent sources regardless of their modality, previous work is hampered by the lack of large-scale aligned cross-modal QA-QG data that also ensures wide QG coverage. As argued in (Rebuffel et al., 2021), such cross-modal QG-QA models can also be used to assess semantic consistency between KG and Text in KG-to-Text generation (i.e. Do questions on the generated text yield the same answer on the input graph and vice-versa?). Finally, cross-modal KG/NL models are key for interacting with KGs in natural language.

One key challenge to building multimodal QG-QA, however, is the lack of annotated QG-QA data aligned between modalities available for training and evaluation of such models. Furthermore, to be useful for RDF-to-Text output evaluation, a multimodal QG-QA approach has to be robust to surface variations since the answer to a question likely has different surface forms in the input graph and the output text. Its QG and QA also have to be cross-lingual so that questions from multilingual texts can be generated and answered from English-centric KGs. Additionally, the set of questions generated for a given (graph, text) pair should be sufficiently large, and the model(s) should also perform QA with accuracy and consistency so as to be able to provide a fair assessment of the semantic consistency between graph and text.

Building on datasets pairing KG graphs with text, we develop a multitask, KG/NL model (QTT) that, given a text in English or a graph from the Wikidata KG, can generate and answer simple and complex questions across the two modalities.

We evaluate the model in terms of QG coverage and internal, cross-modal and external QA

consistency. We also examine the quality of the generated questions using human evaluation. The results show that our approach outperforms previous work across the board. We further demonstrate that our approach also brings improvements to two downstream tasks, namely, better performance with fine-tuning Open-Domain QA architectures and better correlations with human judgements when used for the Data-QuestEval metric (Rebuffel et al., 2021). Our code, data and pretrained models are available at `https://gitlab.inria.fr/hankelvin/quartet_qgqa`.

4.2 Method and data

Our approach comprises the creation of graph-text aligned QG-QA datasets covering simple and complex questions (Section 4.2); and multimodal, multi-task training of a generative QG-QA model (Section 4.3) – we call this model QTT for the number (4, or **Q***uar***T***e***T**) of its main fine-tuning tasks.

To train our QTT model, we derive a dataset of $(g, a_g, t, a_t, g', nf, q)$ tuples from two existing datasets pairing KG triples with their natural language verbalisations, KELM (Agarwal et al., 2021) and WEBNLG(Gardent et al., 2017a).[41] (see Chapter 2).

Our training data is derived from KELM and WEBNLG in three steps. First, we create $(g, a_g, t, a_t, g', nf, q)$ tuples by applying text-based QG and QA on the texts and heuristically aligning text answers with the corresponding graph answers – we call the resulting datasets Q-KELM and Q-WEBNLG^0. Second, we use Q-KELM to train two general multimodal QG models. Thirdly, we apply those models to WEBNLG and add to Q-WEBNLG^0, thereby extending the coverage of the data for training QTT. Figure 4.1 illustrates the process, and we describe the three steps below.

4.2.1 Associating Graphs and Texts with Questions and Answers (Step 1)

Given (g, t), which is an instance of any aligned KG-to-Text dataset, we create synthetic Multimodal QG-QA Data by: (i) generating a question q from t using text-based QG; (ii) extracting the text answer a_t using QA to obtain (t, a_t, q);[42] and heuristically aligning a_t with the corresponding graph answer a_g. To improve quality, we filter out any questions that QA found unanswerable, or whose text answer cannot be aligned with a graph entity. We also heuristically align each generated question with the matching subgraph $g' \subseteq g$ and label it with its size nf i.e., the number of facts each question denotes. The size information permits distinguishing between simple (SQ) and complex (CQ) questions which allows us to take a differentiated approach in Step 2 and generate more QA-QG data. Further details about the implementation details for these steps can be found in Appendix B.2.1.

Our filtering above of unanswerable questions, however, comes at the cost of question coverage – when our full data generation procedure is applied to WEBNLG, only 2,044 CQs remain (Table 4.2). Nonetheless, we keep these (as stated earlier, we call this set Q-WEBNLG^0) to have as wide coverage as possible.[43] Applying the procedure on KELM, we get much larger sets of SQs

[41] In WEBNLG, the graphs are from the DBpedia KG. Here we use a version where some of the DBpedia graphs have been mapped to Wikidata (Han et al., 2022), or else removed of underscores and camelcase to align with the Wikidata format.

[42] In our work, we used `t5-base-e2e-qg`, a T5-base QG model fine-tuned on SQuAD 1.0 data and the deepset RoBERTA-based QA model fine-tuned on SQuAD 2.0.

[43] This was not necessary for SQs since SQ-GEN in our next step (3) generates SQs across varying q_{type} and facts.

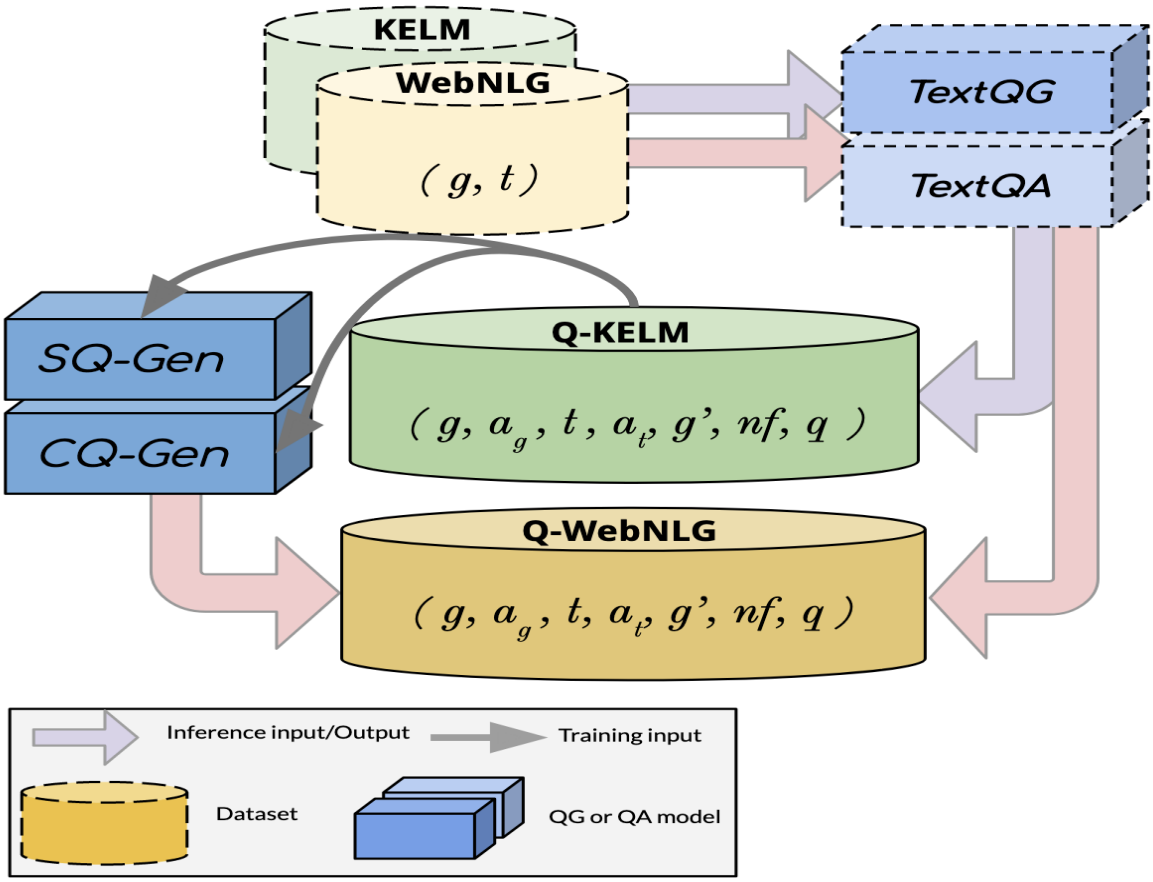


FIG. 4.1: Procedure for generating Q-WebNLG.

and CQs, which we call SQ-KELM and CQ-KELM, and which together form Q-KELM. Some examples of Q-KELM instances are in Tables 4.1 . Table 4.2 shows the sizes of the intermediate datasets resulting from this step.

4.2.2 Training QG Models on Q-KELM (Step 2)

We use Q-KELM to train two multimodal QG models, one (SQ-GEN) fine-tuned on SQ-KELM for simple questions and another (CQ-GEN) tuned on CQ-KELM for complex ones. Both are based on the T5-base public checkpoint, and are able to generate questions from text and from graph.

Using a set of textual prompts (see Table 4.3) at the start of the input sequence to train the CQ-GEN model allows us to controllably generate a complex question from input of the form (X, a_X, a_{type}, nf). Here, a_{type} is the semantic type of the answer entity detected by `ELQ` (Li et al., 2020) or `Duckling`[44]; it is retrieved from the 2021-12-29 Wikidata RDF dump (for entities) or the prediction from `Duckling` (for values). a_{type} is added to improve QG. SQ-GEN is similar to CQ-GEN except that the question type (q_{type}) is used in the input to increase the number and variety of the generated questions (see Table B.2 in Appendix B.2.1 for the textual prompt used here).

[44] https://github.com/facebook/duckling

KELM		
Instance	**Description**	**Example**
Graph g	▪ g: a Wikidata graph	⟨JNR Class DE15, subclass of, diesel-hydraulic locomotive⟩, ⟨JNR Class DE15, instance of, locomotive class⟩, ⟨JNR Class DE15, manufacturer, Kawasaki Heavy Industries⟩, ⟨JNR Class DE15, service entry, 00 1967⟩
Text t	▪ t: an English text generated from g	The JNR Class DE15 is a diesel-hydraulic locomotive class made by Kawasaki Heavy Industries, which entered service in 1967.
Q-KELM		
Instance	**Description**	**Example**
Text (CQ) (t, q, a_t)	▪ q: generated from t using Text QG (Step 1) ▪ a_t: answer derived from (t, q) using Text QA (Step 1)	What is the name of the diesel-hydraulic locomotive class made by Kawasaki Heavy Industries? JNR Class DE15
Text (SQ) (t, q, a_t)	▪ q: generated from t using Text QG (Step 1) ▪ a_t: answer derived from (t, q) using Text QA (Step 1)	When did the JNR Class DE15 enter service? 1967
Graph (CQ) (g', q, a_g)	▪ $g' \subseteq g$: a graph of g heuristically aligned with q ▪ q: generated from t using Text QG (Step 1) ▪ a_g: graph answer i.e. entity in g' heuristically aligned with a_t ▪	⟨JNR Class DE15, subclass of, diesel-hydraulic locomotive⟩, ⟨JNR Class DE15, instance of, locomotive class⟩, ⟨JNR Class DE15, manufacturer, Kawasaki Heavy Industries⟩ What is the name of the diesel-hydraulic locomotive class made by Kawasaki Heavy Industries? JNR Class DE15
Graph (SQ) $(g'^{\lvert 1\rvert}, q, a_g)$	▪ $g'^{\lvert 1\rvert} \in g$: a graph of g, of size one ▪ q: generated from t using Text QG (Step 1) ▪ a_g: graph answer i.e. entity in $g'^{\lvert 1\rvert}$ heuristically aligned with a_t	⟨JNR Class DE15, service entry, 00 1967⟩ When did the JNR Class DE15 enter service? 00 1967

TAB. 4.1: Q-KELM **Dataset.** For each (g, t) pairs in the filtered version of KELM (see Section 4.2), QA pairs are created for both t and $g' \subseteq g$. The question q is generated from t, heuristically aligned with the corresponding graph g', and both the text and the graph answer are extracted from t and g', respectively.

# Facts	**Q-KELM**	**Q-WEBNLG^0**	**QTT-DATA**
1	544,464	–	19,467
2	341,082	1,858	61,346
3	234,170	175	25,170
4	22,607	11	13,918
5	7,031	-	-
TOTAL	1,149,354	2,044	119,901

TAB. 4.2: **Data Statistics.** Number of questions in the QA datasets; nf: the size of the question (no. of facts). Training multimodal QG models on Q-KELM and applying them to WEBNLG drastically enlarges the Q-WEBNLG^0 training data.

Modality		
Text	Input	generate 1 complex question of 2 facts from text [ANS] 1988 [Sp2] value duration | value distance [INP] Peter Vermes represented the United States, where he began his career in 1988 and ended it in 1997.
	Target	cq 1 2 text[ANS] 1988 [QST] When did Peter Vermes begin representing the United States?
Graph	Input	generate 1 complex question of 3 facts from rdf [ANS] SoundApp [Sp2] software [INP] [SUB] SoundApp [PRP] instance of [OBJ] Software [SUB] SoundApp [PRP] operating system [OBJ] System 7 [SUB] SoundApp [PRP] license [OBJ] Freeware
	Target	cq 1 3 rdf[ANS] SoundApp [QST] What is the name of the freeware software program for the operating system of System 7?

TAB. 4.3: **CQ-GEN** Examples of inputs and targets for complex questions training instances, for text and for graph. [ANS], [INP], [QST], [Sp1], and [Sp2] are special tokens we use to demarcate parts of the input/target. [SUB], [PRP] and [OBJ] are used in graph inputs to demarcate subject, property and object elements of a triple.

4.2.3 Extending Q-WebNLG^0 (Step 3)

By applying the controllable QG models from Step 2 to WEBNLG, we can extend Q-WEBNLG^0 from Step 1. This gives us the final training data for QTT, and we call it QTT-DATA.

Our QG models (CQ-GEN, SQ-GEN) generate a question given a context and an answer. Hence, a set of answers must first be selected from the context. For a given X in WEBNLG, we use the same answer selection method as (Rebuffel et al., 2021). For g, the set of possible graph answers is comprised of the subjects and objects in g. For t, it is the set of named entities (NEs) and noun phrases (NPs) detected in t using the `spacy` package.

For SQs from graphs, we follow our previous work on SQ generation from RDF triples (Han et al., 2022) and use the q_{type} prediction model, which returns the set of plausible q_{type} for an answer given its position in the triple and its semantic type.

Finally, we add an answerability+consistency filter on the generated questions by posing them to two QA models[45] and keep only questions where both QA models return an answer which (i) has a confidence score $\geqslant 0.7$, and (ii) shares at least a token overlap with the other model's answer and with the answer used to condition QG.

In sum, by iterating over possible answers, nf from 1 to 4, and q_{type} for SQ-GEN, our controllable approach to QG drastically increases the number of generated questions. A breakdown of QTT-DATA's composition can be found in Table 4.2, and the number of questions associated with the various input and nf size is shown in Table 4.5 .

4.3 QTT, a multimodal QG-QA Model

Our model (QTT) is trained in a multi-task manner to handle both QA and QG. It is based on the T5-small checkpoint (60.5M parameters), allowing for direct comparison with (Rebuffel et al., 2021). We fine-tune on QTT-DATA using four main and four auxiliary tasks, all of which are

[45]The deepset QA above and one based on DeBERTaV3 https://huggingface.co/deepset/deberta-v3-base-squad2.

WebNLG Data	
Graph	[A] ⟨Akita Museum of Art, floor count, 3⟩ [B] ⟨Akita Museum of Art, opening date, 2013-09-28⟩ [C] ⟨Akita Museum of Art, address, 1-4-2 Nakadori⟩ [D] ⟨Akita Museum of Art, floor area, 3746.66 (sqm)⟩
Text	[E] The Akita Museum of Art at 142 Nakadori has 3 floors with a total area of 3746.66 square metres and was inaugurated on 28th September 2013.
QTT Data	
X = graph a_X = g ent	[B], [C] Akita Museum of Art
Complex Questions	{What museum opened in 2013-09-28 in Nakadori? What is the name of the museum that opened in 2013-09-28 in Nakadori?}
X = graph a_X = g ent	[B] 2013-09-28
Simple Questions	{In what year was the Akita Museum of Art opened? Which year was the Akita Museum of Art opened?}
X = text a_X = t span	[E] {The Akita Museum of Art}
Complex Questions	{What is the name of the museum that has 3 floors with a total area of 3746.66 square metres?}
X = text a_X = t span	[E] {2013}
Simple Questions	{What year was the Akita Museum of Art inaugurated? Which year was the museum inaugurated?}

TAB. 4.4: QTT-DATA instances derived from WEBNLG Data. Enclosed letters refer to the triple/text above.

	Text		**Graph**	
nf	avg/min/max	# Qs	avg/min/max	# Qs
1	2.9 / 1 / 7	10,205	2.9 / 1 / 10	9,262
2	2.2 / 1 / 9	52,517	1.6 / 1 / 11	8,829
3	1.5 / 1 / 6	17,734	1.9 / 1 / 17	7,436
4	1.4 / 1 / 6	8,841	1.9 / 1 / 21	5,077

TAB. 4.5: **QTT DATA**. Average, minimum and maximum number of questions for text and graph inputs of size nf (the size is the number of facts matched by the question)

cast in a sequence-to-sequence manner. Using a single Nvidia A40 GPU, it takes approximately 20 hours to fine-tune QTT. The four main and four auxiliary tasks are:

- ***QG from text/graph*** Given (X, a_X, nf), generate a set of questions $\overrightarrow{q}$. We obtain this set by first gathering together questions in QTT-DATA that were generated from a given context X, and which share the same size nf and answer, and then adding to these the questions

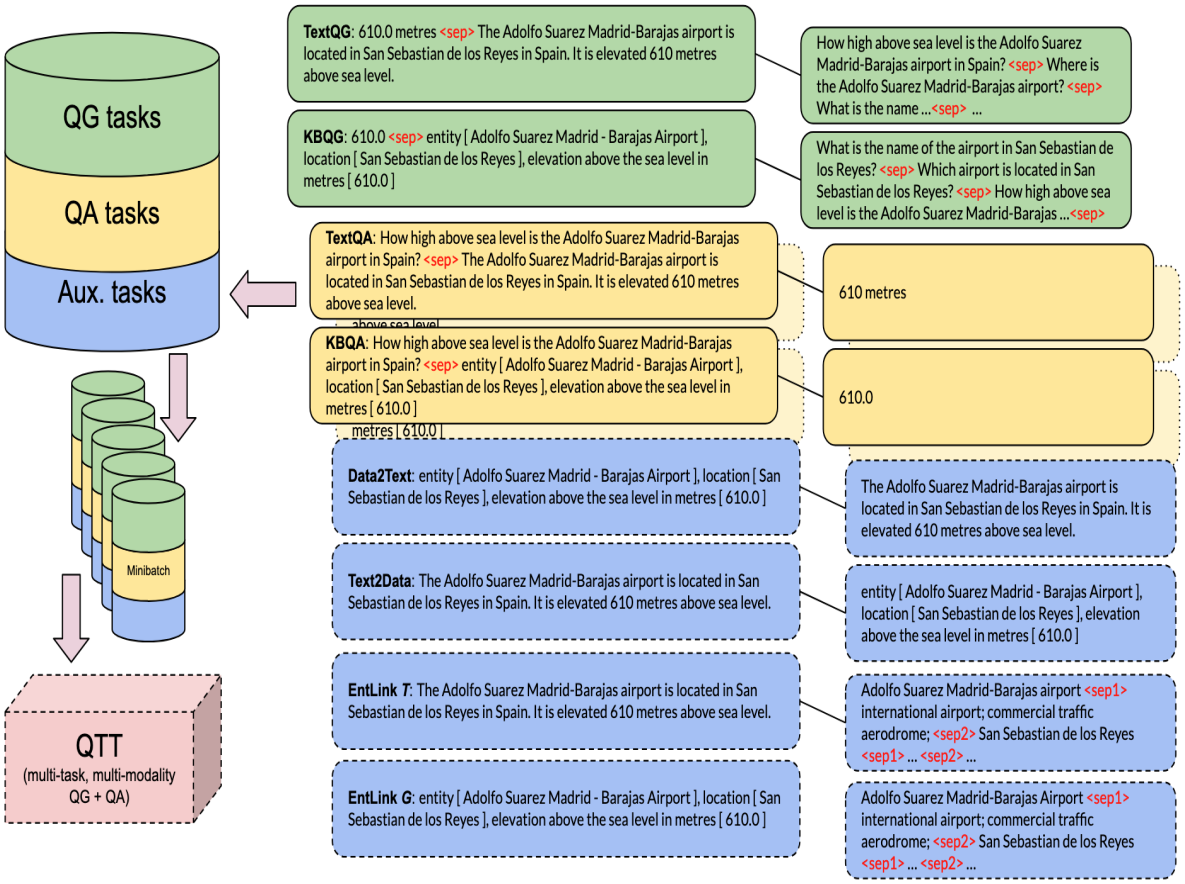


FIG. 4.2: Fine-tuning tasks used for QTT.

generated from other "smaller" pieces of contexts (whose information is fully contained in X), and also sharing the same attributes (nf and answer). This gathering process is detailed in Appendix B.3.1.

▪ ***QA from text/graph*** Given (q, X), generate an answer $\hat{a}_X$. We leverage sets of (X, a_X, q) from QTT-DATA for training these tasks. Additionally, to maximise the use of the data for training QA, we also associate questions answerable by a text t to larger pieces of text that semantically contain t (details in Appendix B.3.1). To allow QTT to abstain from an answer if the question cannot be answered from the context, we use two strategies (details in Appendix B.3.1) to generate negative unanswerable $(q, \neg X)$ pairs.

▪ ***KG-to-Text/Text-to-KG*** These auxiliary tasks consist in either verbalising a graph or deriving a graph from a text. We instantiate each of the WEBNLG graph-text pairs as training instances.

▪ ***Entity typing on graph/text*** These auxiliary tasks combine entity detection and typing. Given a context X, the task identifies the entities/values mentioned in X, and their semantic types. We use `BLINK` (Wu et al., 2020), `Duckling` and Wikidata to obtain the target information for the tasks.

As the number of instances varies across QTT-DATA, we upsampled between modalities and tasks to balance them. Details of each task and upsampling can be found in Appendices B.3.2 and B.3.1. Also, when the input is a graph, we linearise it using the same format for data-to-text tasks in the GEM benchmark (Gehrmann et al., 2021). During training, we use cross-entropy loss as the objective. At inference, we generate with greedy search.

4.4 Experiments

We compare QTT to the original models (DQE, hereafter) used in the Data-QuestEval metric (Rebuffel et al., 2021) in terms of QG coverage, QA accuracy and consistency as well as performance in two downstream tasks. In the following, we describe DQE, our evaluation data and methodology.

▪ **Baseline: the DQE models** DQE comprises four T5-small (Raffel et al., 2020) models fine-tuned for QG-QA from graph and text. For text, their QA model `DQE-TextQA`[46] was fine-tuned on SQuAD 2.0 and the QG model `DQE-TextQG`[47] on SQuAD 1.0. For graphs, both their QG `DQE-KGQG`[48], `DQE-KGQA`[49] and QA models were fine-tuned on a synthetic QG dataset of (g, a_g, q) triples created by applying `DQE-TextQG` to a (g, t) corpus.

4.4.1 Evaluation data

We reserved the test part of WEBNLG, which comprises 1,779 (graph, text) instances, for evaluation. The parallel (g, t) data here ensures that a question can be answered using graph or text, allowing us to check the models' cross-modal consistency (Section 4.4.3). We apply both DQE and our model to the test set and generate questions for every graph and text.

4.4.2 Evaluating QG Coverage

We compare the coverage of QTT against DQE by measuring the number of unique questions they each generated on the WEBNLG test set. We also compare the semantic coverage of the questions using BERTScore (BSc) (Zhang et al., 2020), by taking one model's question for a given entry as prediction and the other's generated questions for the same entry as multi-references. This is repeated with both approaches swapped. The intuition is that if approach A scores higher with approach B's questions as references than vice-versa, A's questions are "contained" in B's, and conversely, B has wider semantic coverage.

4.4.3 Evaluating QA Consistency

In what follows, we refer to a_X, the answer used to condition the generation of q_X, as the ground truth (GT) or the reference answer. We use $\hat{a}_X$ to denote a generated answer. For a given question, $\underline{\hat{a}_X}$ is the answer derived from modality X and $\underline{\hat{a}_{X'}}$ is from modality X'. We use superscripts (e.g. $\underline{\hat{a}_X^A}$ and $\underline{\hat{a}_X^B}$) to distinguish answers generated by different models for the same question q and input context X.

[46] https://huggingface.co/ThomasNLG/t5-qa_squad2neg-en
[47] https://huggingface.co/ThomasNLG/t5-qg_squad1-en
[48] https://huggingface.co/ThomasNLG/t5-qg_webnlg_synth-en
[49] https://huggingface.co/ThomasNLG/t5-qa_webnlg_synth-en

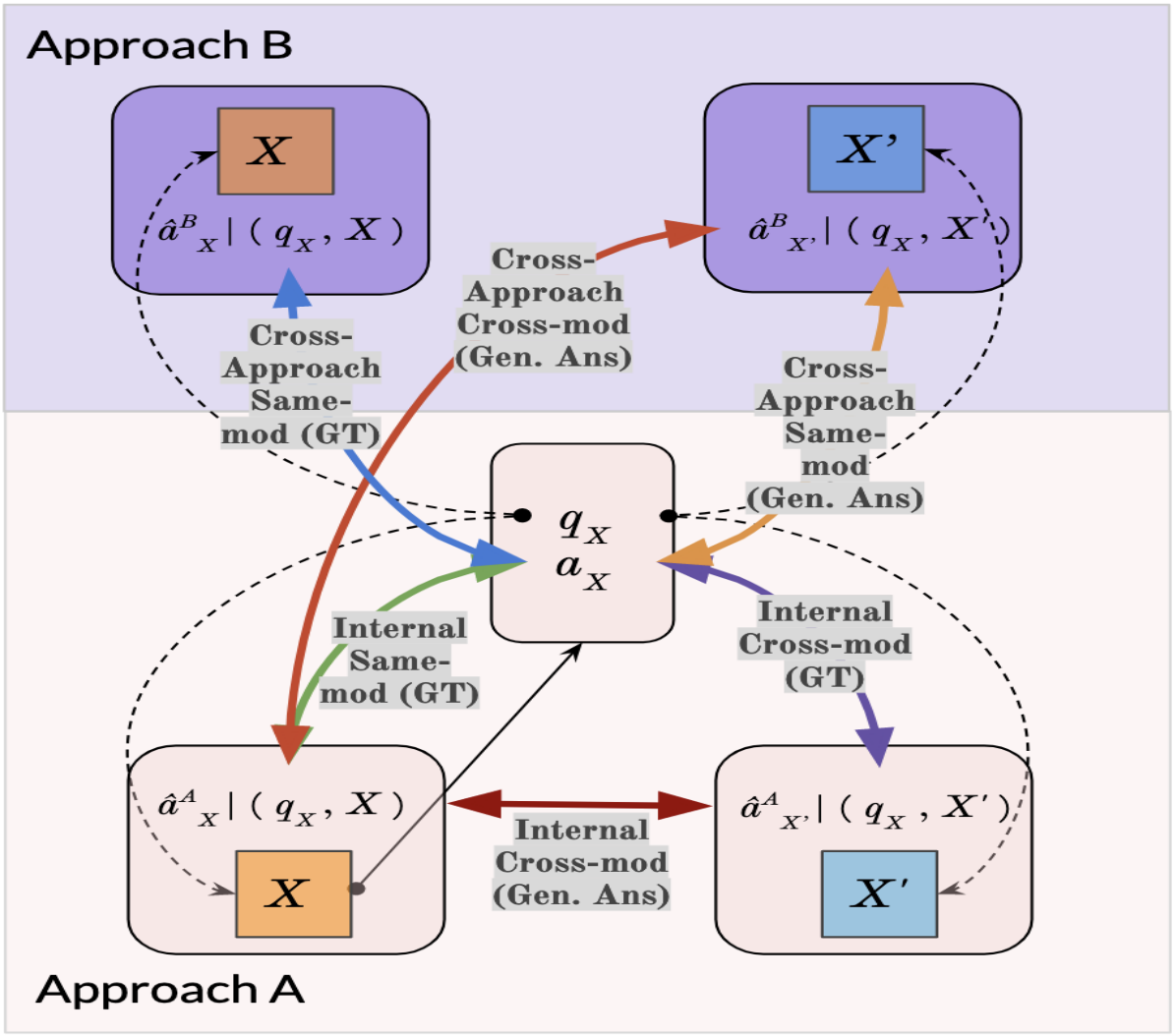


FIG. 4.3: **QA Accuracy**. Bold lines denote QA comparisons within/between modalities and/or approaches. Dotted arrows indicate the context X or X' that the question (q_X) is posed against to obtain the answers.

Accounting for the various ways in which a question can be answered (i.e. $a_X, \hat{a}_X, \hat{a}_{X'}$), we evaluate the quality of multimodal QG-QA models by computing three consistency metrics.[50] *Internal Same-mod (GT)* compares the generated answers $\hat{a}_X$ against the reference answers a_X, indicating the approach's self-consistency. *Internal X-mod (GT)* compares the ground truth a_X with the answer derived from the other modality $\hat{a}_{X'}$. Finally, *Internal X-mod (Gen Ans)* compares $\hat{a}_{X'}$ and $\hat{a}_X$, the answers derived from each modality.

We also investigate QA across approaches, allowing an external indication of each's QG-QA capabilities. Here, we examine the two answers that can be generated by approach B ($\hat{a}^B_X$ and $\hat{a}^B_{X'}$) when given q^A_X, a question generated by A. We do this on three levels: (i) *X-Appr Same-mod (GT)*, by comparing $\hat{a}^B_X$ against a^A_X on the same modality that q^A_X came from; (ii) *X-Appr*

[50] QTT and DQE generates differing numbers of questions for a given X; to ensure a fair evaluation, when an Approach A generates more questions for X, we randomly sample from its set as many questions that Approach B generates for X.

X-mod (GT) comparing B's generated answer with the reference answer across modalities (i.e. $\hat{a}^B_{X'}$ vs a^A_X); and (iii) *X-Appr X-mod (Gen Ans)* where $\hat{a}^B_{X'}$ is compared against $\hat{a}^A_X$. A graphical overview of these QA consistency comparisons can be found in Figure 4.3.

4.4.4 Downstream: QA with FiD

For another external verification of QTT's (and DQE's) QG, we conducted experiments with Fusion-in-Decoder (FiD) (Izacard and Grave, 2021). We use a checkpoint[51] that was trained on TriviaQA (Joshi et al., 2017) as the questions there are factual in nature, i.e. compatible with the texts in WebNLG. To investigate the quality of QTT-Data, we also fine-tune the same FiD checkpoint using either QTT-Data or DQE's training data and use these for QA.[52] Similar to *X-Appr Same-mod (GT)* above, we compare $\hat{a}^B_X$ against a^A_X, except that B, in this case, is a given fine-tuned (or not) FiD QA model while A is DQE or QTT. Though such a setting gives upper-bound FiD scores,[53] the differences in scores – when varying fine-tuning data and QG – independently validates QTT's QG vs DQE's and QTT-Data too.

4.4.5 Downstream: Data-QuestEval metric

Since DQE was originally used in the Data-QuestEval metric, we also compare the correlation of the resulting Data-QuestEval metric with human judgements when DQE is replaced with QTT. For this, we compute the correlations with the judgements collected on 2,007 outputs from 9 participating systems in the WebNLG Challenge (Gardent et al., 2017b). Following (Rebuffel et al., 2021), we compute Pearson's r , but also report Spearman's ρ.[54]

4.4.6 Evaluation settings

The following describes and explains the settings for our evaluations.

▪ ***Answer selection for QG inference*** To have a direct comparison with (Rebuffel et al., 2021)'s models, we follow their use of the `spacy` pipeline for answer selection on t (see Section 4.2.3), and report results with this setting in Sections 4.4 and 4.5. However, this approach is noisy and often segments NEs with nouns or NPs (e.g., 'English' being extracted from 'English Without Tears'), leading to ill-formed questions. We trained an answer selection model for texts (see Appendix B.3.2) using QTT-Data, ensuring that the answers for QG are meaningful spans in t. We conducted our ablation experiments and human evaluation (Sections 4.5.4 and 4.5.5) using this answer selection method for text.

▪ ***Automatic metric*** Following (Rebuffel et al., 2021), we use BERTScore (BSc) for evaluation to address the restrictiveness of token F1 for cross-modality QA. We use the same settings, except the following for a clearer analysis: (i) BScs were rescaled against the official BSc baseline for a wider spread, (ii) "unanswerable" strings were set to an empty string to avoid non-zero BSc for these and "over-counting" them, and (iii) lowercasing.

[51] https://dl.fbaipublicfiles.com/FiD/pretrained_models/tqa_reader_base.tar.gz

[52] Here, when the context is a graph, we use the linearisation scheme from (Oguz et al., 2022) to utilise FiD for KGQA.

[53] i.e. the information to answer the question is in a single document and this gold document is being provided to FiD.

[54] The latter may be appropriate since the system outputs for the WebNLG 2017 challenge evaluation were selected to cover a spread of automatic scores and are therefore unlikely to be normally distributed.

▪ ***Self-consistency filter*** Since both QTT and DQE are trained on synthetic data, some generated questions may be ill-formed and pose an impact on QA. We, therefore, filter from both QTT and DQE the questions: (i) which cannot be answered from their source context; or (ii) whose generated answer $\hat{a}_X$ has a BSc < 0.7 when compared against the reference a_X. We focus our analysis for QA Consistency and the Downstream Evaluations on the results after filtering as this is the upper bound of the approaches' performance; the impact of removing this filter is included in our ablations (Section 4.5.4). For congruence with our QG Coverage analysis, if the filtering will leave a given approach A – and therefore B as well – with no QA pairs (i.e. no coverage), we keep one QA pair for A.

We note also that the self-consistency filter above is important in the downstream Data-QuestEval evaluation (see above) – given how the Data-QuestEval metric is computed (i.e. the generated answer is compared against the GT answer) if a generated question cannot be answered by the source context and yet still posed to the other modality, it will not capture the factuality comparison accurately (i.e. it will skew the metric and affect its reliability).

4.5 Results

4.5.1 QG Coverage

QTT generates three times as many questions as DQE when the input is a text (14,141 vs 42,959) and 10 times more when it is a graph (7,272 vs 75,906). Figure 4.5 provides a fine-grained view of the QG coverage by the (graph-based) size of the context.

This higher coverage results from three modelling choices differentiating QTT from DQE: (i) QTT is trained to generate multiple questions from a given X; (ii) the enlarged size and coverage of QTT's training data by applying Q-KELM-trained QG models to WebNLG; and (iii) the use of q_{type} controls in SQ-Gen, permitting multiple SQs of various types to be generated from a single X.

The BScs are also higher with QTT's questions as references (89.7 vs 85.3 for text; 90.4 vs 82.5 for graph), suggesting that QTT is not just generating more questions but also ones that "contains" DQE's as well as semantically different ones from DQE's.

4.5.2 QA Consistency

Table 4.6 summarises the comparisons between QTT and DQE on QA accuracy. A finer-grained analysis of QTT's Internal performance across question complexity can be found in Table B.1 in Appendix B.1.

▪ ***QTT outperforms DQE on self-consistency*** Despite QTT and DQE both starting fine-tuning from the same T5-small checkpoint, QTT gains over DQE in *Internal Same-mod (GT)* (+8.0 BSc for text, +3.8 for graph). This shows that using QTT-Data – which provides aligned wide-coverage QA-QG data – for training in a multimodal multi-task manner enables QG and QA with greater internal roundtrip consistency.

▪ ***Our synthetic in-domain data improves performance*** QTT's self-consistency gains over DQE for text also stem from our procedure for creating in-domain data. `DQE-TextQG` and `DQE-TextQA` were fine-tuned on SQuAD only, leading to a drop in scores when `DQE-TextQA` is applied on WebNLG (vs `DQE-KGQA`'s 95.0). This out-of-domain effect also shows when it answers

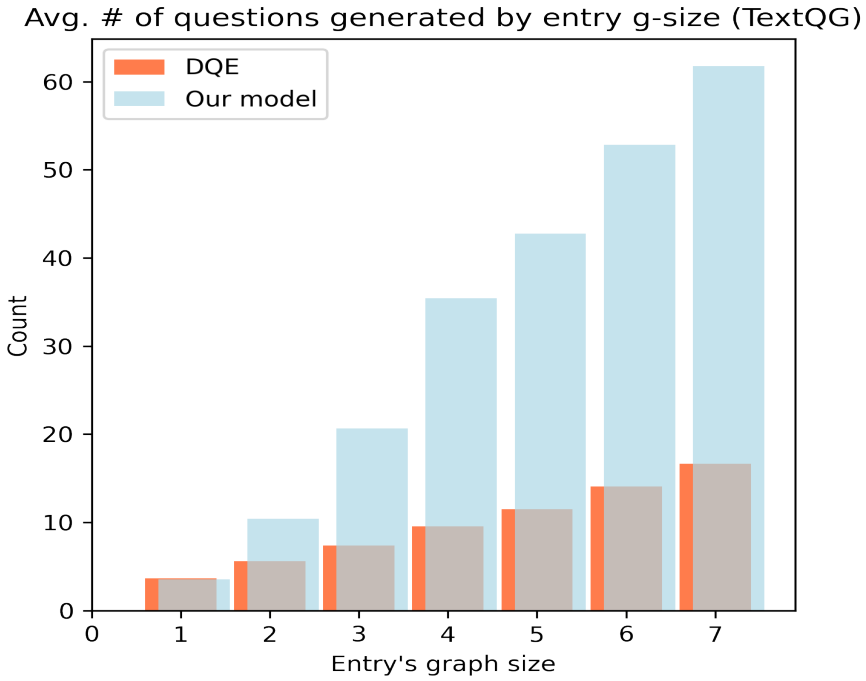


FIG. 4.4: **Comparative QG coverage DQE vs. QTT.** Total and average number of generated questions is much higher for QTT across both modality and input size. The delta increases with the size of the input.

a question QTT generated with text (95.4 to 56.8, Table 4.6); whereas, in the reverse case, the drop for QTT when it answers a DQE question generated is much less (87.4 to 81.4).

▪ ***Q-DATA with multi-task training improves cross-modality performance*** QTT outperforms DQE by at least 9.1 BSc (e.g. 69.7 vs 60.6 for T → G) in *Internal X-mod (GT)* showing that it can more accurately answer questions cross-modally with respect to the reference answer, and/or generate questions that allow this. This is beneficial when using the QG-QA model(s) for evaluation, such as the Data-QuestEval metric where this comparison ($\hat{a}_{X'}$ and a_X) is relied on to assess the semantic concordance between the data input and generated text. Furthermore, a low discrepancy in the *Internal X-mod (Gen Ans)* performances between the cross-modal directions (i.e. $G \rightarrow T$ vs $T \rightarrow G$) is ideal under our parallel data evaluation setting, as it shows that the QG-QA model(s) is able to reflect the agreement between the (g, t) pairs. QTT's 4.3 BSc gap here narrows by nearly two-thirds the 12.1 BSc gap faced by DQE (i.e. 76.4-72.1 vs 51.8-63.9).

▪ QTT***'s performance is consistent across question complexity and externally validated*** We find that QTT also performs consistently for all nf, (the number of facts q relates to) for text and for graph (Table B.1 in Appendix B.1). Our findings above on QTT's internal performance also hold when examined cross-approach (*X-Appr*, i.e. external); whenever QTT is used to answer questions generated by DQE, the drop in QA accuracy is significantly lower (or in some cases a gain) than vice-versa.

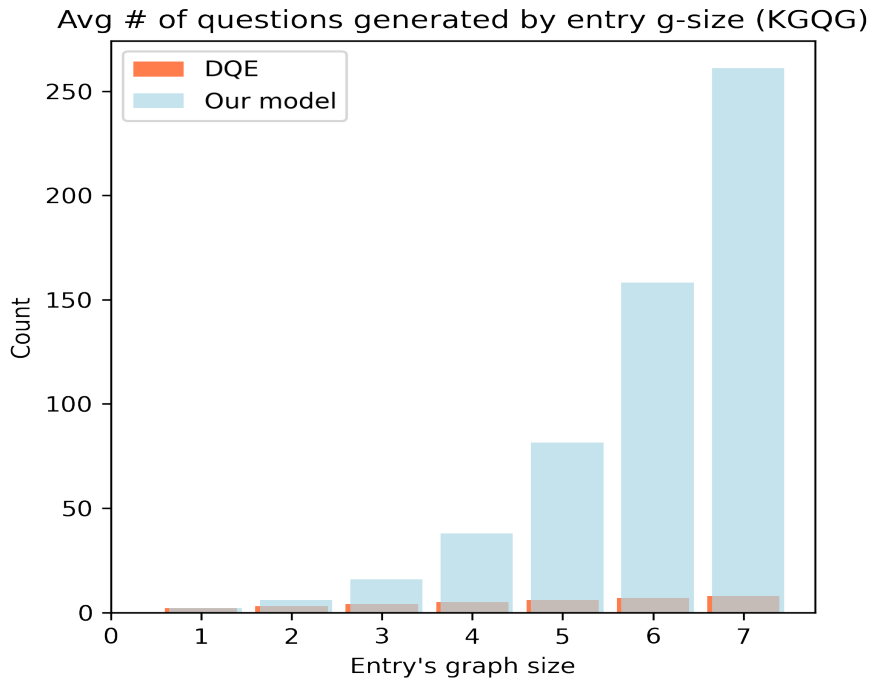


FIG. 4.5: **Comparative QG coverage DQE vs. QTT.** Total and average number of generated questions is much higher for QTT across both modality and input size. The delta increases with the size of the input.

4.5.3 Downstream Evaluations

QTT also outperforms in two downstream tasks. Our FiD experiments (Table 4.7) show that the QTT's questions can be answered with higher accuracy than DQE's for both modalities – likely because QTT's training data permits it to generate more CQs than DQE, which is closer to the forms in TriviaQA. The combination of fine-tuning FiD with QTT-DATA and QTT QG also betters every other combination with DQE and/or its training data. These validate (i) our procedure for creating QTT-DATA, and (ii) the use of it with multimodal multi-task modelling for improved QG. Besides that, using QTT for computing the Data-QuestEval metric also boosts by >10 points the score's correlation (Spearman's ρ) with human judgements of semantic adequacy (Table 4.8). This is likely due to the improved QG (quality and coverage) for both modalities, allowing QA-based evaluation to more accurately assess the information content of the data and the generated text.

4.5.4 Ablation

We also studied the impact of four variations to the data and modelling: (i) *X-Filt* removes the question self-consistency filter (Section 4.4.6); (ii) *Data* uses a similar data setting as (Rebuffel et al., 2021) i.e. synthetic QG-QA data on WEBNLG plus SQuAD, without Q-KELM and Steps 2 & 3 in Section 4.2; (iii) *SPO* uses another graph linearisation, where each (s, p, o) in g is shown as they appear; and (iv) *X-Aux*, removes all auxiliary tasks. All these models were trained for 383,445 steps. These ablation results can be found in Table 4.9.

	Internal		X-Appr	
QG:	DQE	QTT	DQE	QTT
QA:	DQE	QTT	QTT	DQE
	Same-mod (GT)			
T → T	$87.4_{(\pm 0.01)}$	**95.4**$_{(\pm 0.14)}$	$81.4^{(\mathbf{-6.0})}_{(\pm 0.02)}$	$56.8^{(-38.6)}_{(\pm 0.31)}$
G → G	$95.0_{(\pm 0.01)}$	**98.8**$_{(\pm 0.04)}$	$76.2^{(\mathbf{-18.8})}_{(\pm 0.05)}$	$79.4^{(-19.4)}_{(\pm 0.47)}$
	X-mod (GT)			
G → T	$51.4_{(\pm 0.04)}$	**75.8**$_{(\pm 0.35)}$	$60.3^{(\mathbf{+8.9})}_{(\pm 0.04)}$	$48.2^{(-27.6)}_{(\pm 0.52)}$
T → G	$60.6_{(\pm 0.02)}$	**69.7**$_{(\pm 0.35)}$	$59.8^{(\mathbf{-0.8})}_{(\pm 0.02)}$	$51.6^{(-18.1)}_{(\pm 0.27)}$
	X-mod (Gen Ans)			
G → T	$51.8_{(\pm 0.04)}$	**76.4**$_{(\pm 0.35)}$	$58.2^{(\mathbf{+6.4})}_{(\pm 0.04)}$	$48.4^{(-28.0)}_{(\pm 0.51)}$
T → G	$63.9_{(\pm 0.01)}$	**72.1**$_{(\pm 0.38)}$	$61.2^{(\mathbf{-2.7})}_{(\pm 0.02)}$	$53.1^{(-19.0)}_{(\pm 0.28)}$

TAB. 4.6: **Consistency Results** Avg. of BScs between answers. In $_{\text{subscripts}}$ are std. dev. across 5 random runs; $^{\text{superscripts}}$ are the difference between X-Appr and Internal. X/Y indicates the QG/QA model used. QTT betters DQE on all consistency tests and for all modalities.

	Fusion-in-Decoder					
QG:	DQE	DQE	DQE	QTT	QTT	QTT
QA:	FiD^0	FiD^D	FiD^Q	FiD^0	FiD^D	FiD^Q
	Same-mod (GT)					
T → T	68.94	86.24	86.60	77.06	88.89	**91.48**
	(0.01)	(0.04)	(0.02)	(0.42)	(0.26)	(0.18)
G → G	74.25	91.07	88.66	82.14	91.10	**94.99**
	(0.03)	(0.02)	(0.05)	(0.33)	(0.27)	(0.17)

TAB. 4.7: **Consistency Results with FiD** (Similar to Table 4.6.) FiD^0 is the public FiD checkpoint trained on TriviaQA. FiD^D/FiD^Q denotes that checkpoint fine-tuned on training data from DQE/QTT for that modality.

Measure	DQE	QTT
Spearman's ρ	47.9 (1.47e-104)	**58.6** (4.81e-168)
Pearson's r	51.8 (2.69e-125)	**61.8** (1.24e-191)

TAB. 4.8: **Correlations with human judgements.** Comparing when DQE and QTT are used in the computation of the Data-QuestEval metric. All (p-values) « 0.001.

The ablations show that other than the question self-consistency filter and using QTT-DATA, the other variations have relatively limited impact on QTT's *Same-mod (GT)*. It also shows that using our data generation procedure leads to improvements in QA; especially in cross-modal settings (*X-mod (GT)* and *X-mod (Gen Ans)*).

	QTT	X-Filt	Data	SPO	X-Aux
	Same-mod (GT)				
T → T	**97.6**	76.6	95.7	97.4	97.4
G → G	98.8	86.0	**99.5**	97.8	98.9
	X-mod (GT)				
G → T	**75.5**	70.3	71.4	75.3	75.4
T → G	**77.7**	63.8	64.4	74.9	75.8
	X-mod (Gen Ans)				
G → T	76.0	**76.3**	71.7	76.0	76.0
T → G	**78.3**	72.3	66.6	75.5	76.7

TAB. 4.9: **Ablation results.** All models here use the text answer selector. Comp. denotes consistency comparison, and Mod. denotes QG and QA modalities, respectively.

4.5.5 Human evaluation

Modality	Consistency	Naturalness	Complexity
Text	0.76 (0.53)	0.77 (0.45)	0.75 (0.72)
Graph	0.64 (0.56)	0.67 (0.55)	0.93 (0.86)

TAB. 4.10: **Human evaluation for QG.** Each score is the average of all annotators' ratings. Values in brackets are the Fleiss' kappa coefficient for that particular aspect.

We conducted a human evaluation on the quality of QTT's questions since there are no reference questions. To have a broad study, we sampled 10 questions each from bins combining these characteristics: (i) modality - whether QG from graph or text; (ii) complexity - the question's size (nf); and (iii) QA accuracy, where the questions were separated into three quantiles based on their *Internal Same-mod (GT)* score.[55]

Three doctoral candidates in NLP fluent in English were shown the 240 questions and their contexts and asked to rate each question on three aspects with a binary choice: whether it is (i) consistent with the context, (ii) natural-sounding, and (iii) a CQ. For the last aspect, we report whether their rating matches the control (nf=1 or nf>1) used for generation. The results can be found in Table 4.10.

The overall agreement between the annotators is substantial (Fleiss' kappa: 0.65 for KGQG; 0.61 for TextQG). For TextQG, QTT scores $\geqslant$ 0.75 on all aspects. In KGQG, we observe lower scores for *Consistency* and *Naturalness*. This is likely due to the increased challenge when generating from graph (which is under-specified and requires a semantic gap to be overcome), and is compounded by unseen properties in the WEBNLG test set. There is also a difference in the performance for *Complexity* between the graph and text modalities; we found that this can be attributed to the challenge of accurately judging KG facts in text.[56]

[55] e.g. 1 bin is {G, nf=1, 1st quant.} i.e. SQs from graph, with BSc for $\hat{a}_g$ vs a_g in top 33% of all SQs from graph.

[56] For e.g. the question "*Who was born in Los Angeles in California?*" generated from the text "*The birthplace of X is Los Angeles in California.*" corresponds to the single KG fact ⟨ X; born in; Los Angeles, California ⟩ but may be judged as being a CQ of more than 1 fact (e.g. ⟨ X; born in; Los Angeles > and < Los Angeles; located in; California ⟩).

4.5.6 Question generation examples

Modality	Input/Generation
Text	Nord is a post metal album by Year of No Light and was released by Crucial Blast record label.
Question	What is the name of the post metal band that was released by Crucial Blast?
Observations	The question is a complex one (covering multiple facts), and consistent with the information in the input text. However, its phrasing (in blue) is slightly unnatural.
Graph	{ ⟨ Nord (Year of No Light album) , artist , Year of No Light ⟩, ⟨ Nord (Year of No Light album) , genre , Post-metal ⟩, ⟨ Nord (Year of No Light album) , record label , Crucial Blast ⟩ }
Question	What genre of music did Nord perform in the year of No Light?
Observations	The question is a complex one that is consistent with the KG facts. There is a minor typographical error (in blue).
Text	ENAIRE, located in Madrid, is the operating organisation for Adolfo Suarez Madrid-Barajas airport in Alcobendas. The airport is elevated 610 metres above sea level and its runway name is 14L/32R.
Question	Where is the ENAIRE located?
Observations	The question is a simple one. It is consistent with the information in the text and relatively natural sounding.
Graph	{ ⟨ Adolfo Suarez Madrid - Barajas Airport , operating organisation , ENAIRE ⟩, ⟨ Adolfo Suarez Madrid - Barajas Airport , location , Alcobendas ⟩, ⟨ Adolfo Suarez Madrid - Barajas Airport , runway length , 4349.0 ⟩, ⟨ Adolfo Suarez Madrid - Barajas Airport , elevation above the sea level , 610.0 ⟩, ⟨ Adolfo Suarez Madrid - Barajas Airport , runway name , 14L/32R ⟩, ⟨ ENAIRE , city , Madrid ⟩ }
Question	What is the name of the airport operated by ENAIRE that is located at 610.0 and 14L/32R?
Observations	The question is a complex one. It has information that is consistent with the graph (no hallucination), but it is not natural-sounding (it is missing the unit of measurement for elevation above sea level, and a span of text that could possibly have been: "has a runway named")
Text	McVeagh of the South Seas is a film directed by Cyril Bruce and Harry Carey. It's a film that is logged in the IMDb database with the ID of 0004319. Carey was born on 1878 and was the writer of the film while he played a major character in the movie too. This movie was distributed by Alliance Films Corporation. [Question is natural sounding]"
Question	Cyril Bruce and Harry Carey are the creators of McVeagh of the South Seas, which is in what database?
Observations	The question is complex and the information in it is consistent with the text.
Graph	{ ⟨ McVeagh of the South Seas , imdb id , 0004319 ⟩, ⟨ McVeagh of the South Seas , director , Cyril Bruce ⟩, ⟨ McVeagh of the South Seas , director , Harry Carey (actor born 1878) ⟩, ⟨ McVeagh of the South Seas , starring , Harry Carey (actor born 1878) ⟩, ⟨ McVeagh of the South Seas , writer , Harry Carey (actor born 1878) ⟩, ⟨ McVeagh of the South Seas , producer , The Progressive Motion Picture Company ⟩, ⟨ McVeagh of the South Seas , distributor , Alliance Films Corporation ⟩ }
Question	What is the id of the South Seas written by Cyril Bruce and Cyril Bruce
Observations	The question is complex and is consistent with the information in KG input. However, the named entity "McVeagh of the South Seas" is not fully lexicalised (it is missing "McVeagh of"). Depending on the intended use for the question, whether the relation "imdb id" is lexicalised only as "id" may be acceptable (when a context is given or accessible, so that IMDB ID can be established in the ground) or ambiguous (e.g. a search query).

Tab. 4.11: Examples of the questions generated by QTT for both text and graph inputs, as well as observations of the errors present in them.

In Table 4.11 we show some examples of the questions generated from QTT. Some questions have issues with the naturalness of how they have been phrased. This is especially so for complex questions (the question generated from the text input for the first example in the table) as it appears that the model struggles between copying the information in the input or to slightly paraphrase the generation. We observe that some of the questions generated from graph inputs have errors related to the semantic gap inherent in KG inputs, as the model may struggle to fill

the gaps in a way that is consistent with the context, a task which humans might easily infer from the context and world knowledge (see the question generated from the graph input in the second example).

4.6 Conclusion

In this chapter, we proposed an approach (QTT), which we showed generates more questions that cover more information compared to previous work (DQE). Unlike existing approaches, our data and architecture allow us to generate multiple questions for a given input. The extensive internal, cross-modal and external checks we conducted showed that QTT outperforms DQE on QA consistency. The quality of our generated questions was also verified with human evaluation for semantic consistency, naturalness and adherence to our complexity controls. Finally, the use of our approach also leads to improvements against DQE in two downstream evaluations (QA with Fusion-in-Decoder and the Data-QuestEval metric). Our main contributions are (i) a large multimodal general QG-QA dataset (Q-KELM), (ii) a data generation procedure including two general multimodal QG models enabling controllable generation of in-domain synthetic QG-QA datasets, and (iii) a multimodal multi-task QG-QA model that can generate and answer questions from text and from graph.

5

Multilingual Generation and Answering of Questions from Texts and Knowledge Graphs

Sommaire

5.1 Introduction 66
5.1.1 Terminology and notations 67
5.2 Approach 67
5.3 Data 68
5.4 Creating QA/QG Data for English 68
5.5 Creating QA/QG Data for Russian and Portuguese 70
5.6 Multimodal multi-task QG-QA model 70
5.7 Evaluation and results 72
5.7.1 Baseline: single-task models 72
5.7.2 Evaluating Question Coverage 72
5.7.3 Evaluating QA Consistency 72
5.7.4 Data-QuestEval metric 76
5.7.5 Multilingual Retrieval-based QA 79
5.8 Conclusion 79
9 Summary of findings 81
10 Future directions 82

The ability to bridge Question Generation (QG) and Question Answering (QA) across structured and unstructured modalities has the potential for aiding different NLP applications. As we examined in the previous chapter, one key application is in QA-based methods for automatically evaluating Natural Language (NL) texts generated from Knowledge Graphs (KG). However, methods for QG-QA across these modalities have been in English only; in this chapter, we bring multilinguality (Brazilian Portuguese and Russian) to multimodal (KG and NL) QG-QA. Using synthetic data generation and machine translation to produce QG-QA data that is aligned between graph and text, we are able to train multimodal, multi-task models that can perform multimodal QG and QA in Brazillian Portuguese and Russian. We show that our approach outperforms a baseline which is derived from previous work on English

and adapted to handle these two languages. Our code, data and models are available at https://gitlab.inria.fr/hankelvin/multlingual_kg-text_qgqa.

In this chapter, we outline the reasons for extending multimodal QG and QA to a multilingual setting in Section 5.1. We then describe our general approach for doing so in Section 5.2, followed by details on the Brazillian Portuguese and Russian datasets that we use in our work in Section 5.3. Sections 5.4-Section 5.5 contains the procedures for how we generate the QG/QA data aligned between texts and KG graphs for these languages. After describing our modeling approach in Section 5.6 using the generated data, we outline our evaluation approach as well as results and findings obtained in Section 5.7 before concluding in Section 5.8.

5.1 Introduction

The ability to generate and answer questions from both Knowledge Graphs (KG) and Natural Language (NL) text is useful in a number of ways. It permits easing users' access to the knowledge contained in KGs without having to master complex query languages, since an NL query from a user may be directly applied to KG information to derive the answer. It also allows for questions to be generated and answered from both the open domain information contained in NL text and the factual knowledge contained in KGs/knowledge bases, thereby widening the pool of information that is interrogable.

It is also useful for verifying the consistency of information between modalities. In particular, (Rebuffel et al., 2021) recently showed that multimodal KG/NL Question Generation (QG) and Question Answering (QA) can be used for assessing the semantic consistency between an input KG graph and a generated English text, thereby providing a reference-less metric to measure the quality of KG verbalisers (RDF-to-Text models). The intuition is that, for a match between an input and its output, the answers that are extracted for a given set of questions from the input should be consistent with the answers that are extracted from the output (for the same set of questions).

In this work, we investigate the generation and answering of questions from graph and from text for multiple languages besides English. We present models enabling this for Russian and Brazilian Portuguese. In addition, we apply our approach to WebNLG, a dataset of (graph, text) pairs that is used to train RDF-to-Text models, and we show that our approach extends to (Rebuffel et al., 2021)'s reference-less, semantic adequacy metric for RDF-to-Text models for these two languages.

We make the following contributions. First, we create training data where questions are aligned with both their graph and their text answers, allowing the training of models that show cross-modal consistency: i.e. for a given question, the answers obtained from a graph and from its semantically equivalent text are consistent.

Second, we derive silver training data from English using Machine Translation (MT) and we demonstrate that the resulting multimodal, Portuguese and Russian QG-QA models trained on them have high internal consistency: in most cases, applying a Russian or Portuguese question generated from a graph to the same graph returns an answer that matches the graph entity that the question was originally conditionally-generated from.

Third, we show that our approach has much larger question coverage than a baseline adapting (Rebuffel et al., 2021)'s approach to our target languages. Finally, to overcome the lack of any gold QA/QG data that is aligned between text and graph in these two languages, we designed our evaluation suite to include multiple internal, cross-modal, cross-approach and external checks. These checks also demonstrate that our approach significantly improves on the baseline.

5.1.1 Terminology and notations

En	English
$PtBr$	Brazilian Portuguese
Ru	Russian
g	(or *KG graphs/graphs*). A graph of the Wikidata KG (Vrandečić and Krötzsch, 2014); comprised of a set of triples (also called facts) of the form ⟨subject, predicate, object⟩ in English.
t	(or *NL texts/texts*). A text in English/Portuguese/Russian
X	The context of a question in one modality (text or graph)
X'	Semantically equivalent to X in the other modality
q	A question
$\overrightarrow{q}$	A collection of questions
a_X	An answer in X (a graph answer (a_g) is either a subject or an object entity in g; a text answer (a_t) is a span in t)
g'	A graph of g corresponding to a q and its answer
nf	The number of facts related to a given q (i.e. the size of its corresponding graph $\|g'\|$)
t^{PtBr}	A text in Portuguese (the superscript distinguishes languages)

5.2 Approach

Since there is no cross-modal QG/QA data for Brazilian Portuguese and Russian, we approach the task by first creating the data necessary for training multimodal QG-QA models for English. We then machine translate this data to create training data for Brazilian Portuguese and Russian. Finally, we use this automatically translated data to train multimodal (KG/NL), multi-task (QG-QA) models for these two languages. The following outlines our approach and details are provided in the subsequent sections.

- **Creating Question/Answer Data for pairs of English Texts and Wikidata Knowledge Graphs.** We first create a large synthetic dataset (Section 5.3) of QA pairs for graphs and texts in English, as leveraging existing resources (datasets and models) already developed for that language allows us to obtain QA pairs at scale. We do this by applying off-the-shelf QG and QA models to texts from KELM (Agarwal et al., 2021), a large dataset of (Wikidata graph, English text) pairs. We call this data Q-KELM^{En}.

- **Learning Controllable QG Models for English Texts and Wikidata Knowledge Graphs.** Using the Q-KELM^{En} dataset for training, we learn two controllable graph- and text-based QG models, which allow for multiple, varied questions of different graph sizes and question types to be generated from the same (graph, text) pairs. This is essential as it is through controllable generation with these models that we can significantly increase the QA/QG training data coverage (Section 5.7.2).

- **Creating Question & Answer Data for the WebNLG Dataset (from both English Texts and Graphs).** By applying our controllable QG models to the (graph, text) pairs of the

WebNLG (Gardent et al., 2017a) dataset, we create a large dataset of (question, answer) pairs from texts and graphs that can be used to train models used in verifying the semantic match between WebNLG graphs and texts generated from these graphs.

- **Learning Multimodal, Multi-task QG-QA Models for Brazilian Portuguese and for Russian.** By further using MT, heuristics and quality filters (Section 5.5), we obtain silver-aligned, in-domain QA pairs that enable us to train QG-QA models for WebNLG graphs and their texts in Portuguese or Russian.

- **Testing on WebNLG and with an independent QA model.** We apply our Portuguese and Russian, multimodal QG-QA models to WebNLG Portuguese and Russian evaluation data and compute correlation (Section 5.7.4) with human judgement of semantic adequacy (i.e. Does a generated text match the semantic content of its input graph?). We also apply the generated questions to a retrieval-based multilingual QA model (Section 5.7.5) to further verify the quality of the QG (i.e. How well can the generated questions be answered by an external model?).

We show that our approach brings substantial improvements over the baseline, to question coverage, QA consistency, correlations with human judgements of semantic adequacy as well as answerability by a retrieval-based multilingual QA model.

5.3 Data

For this study, we use WebNLG and Kelm (see Chapter 2 alongside two other data-to-text datasets that are in Russian and Brazilian Portuguese, namely:

- WebNLG^{Ru}: (Shimorina et al., 2019) created a Russian version of WebNLG by machine translating 16,522 English texts from the original WebNLG dataset, followed by crowdsourced post-editing. This dataset was used in the 2020 WebNLG Challenge (Castro Ferreira et al., 2020).

- WebNLG^{PtBr}: Similarly, (Almeida Costa et al., 2020), created a Brazilian Portuguese version of the WebNLG test set by using MT and a pool of native Portuguese speakers for post-editing.

We reserve the test sets of WebNLG^{PtBr} and WebNLG^{Ru} — 1,606 (g, t^{PtBr}) and 1,102 (g, t^{Ru}) pairs, respectively — for evaluation as the parallel (g, t) in these test sets means that if a question can be answered by a text (or graph), it can also be answered by the graph (or text) that it is paired with — this allows us to test cross-modal QG and QA for consistency (Section 5.7.3). We use the training portion of WebNLG to produce our QA/QG datasets, which is done in two phases and which we describe in the following sections and illustrate in Figure 5.1.

5.4 Creating QA/QG Data for English

Our first phase generates synthetic (question, answer) pairs forWebNLG graphs and English texts in three main steps. The process is similar to the steps taken in Chapter 4 for generating English QG/QA data across text and KG. Briefly, to recap, this involves the following steps: (i) **Obtaining synthetic QA data**, (ii) **Training English Text/KG QG models**, and (iii) **Obtaining in-domain training data**. In the first step, we derive (t, a_t, q) triplets from KELM

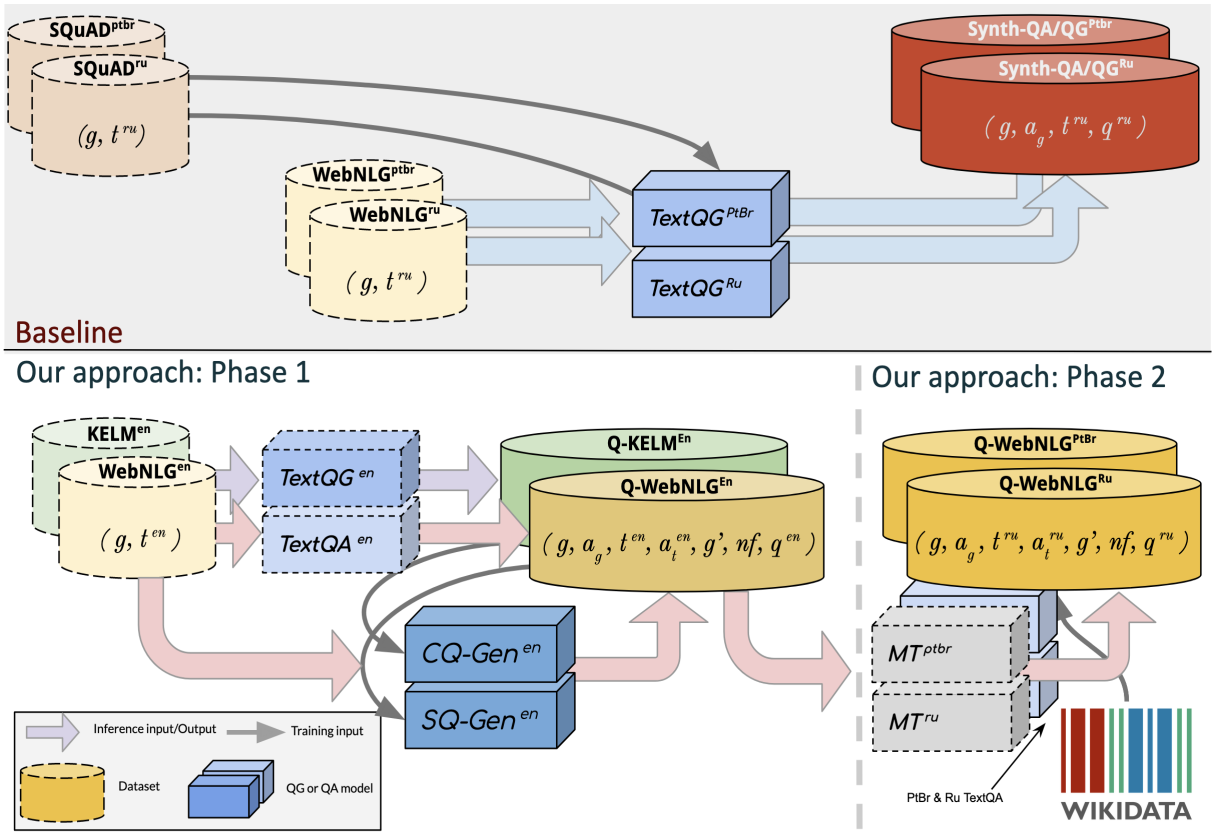


FIG. 5.1: **Comparing our approach against the baseline.** Using KELM and controllable QG increases coverage. Aligning questions with contexts and answers across modalities improves consistency. Machine translation provides multilingual training data.

texts using off-the-shelf models for textual QG and QA[57] and call this resulting dataset Q-KELM^{En}. This is followed by training two QG models using Q-KELM^{En}, that can controllably generate simple (one KG graph fact) and complex (>1 KG graph fact) questions from either text or graph. Finally, these two models are applied to the training portion of WEBNLG.

It is the controllable QG models in step 2 that enable our generation of wide-coverage in-domain QG-QA data that also allows us to generate multiple questions per input (Section 5.6). By cycling through the set of possible graph and text answers in step 3, and leveraging the controls we introduced in Step 2 (question sizes and question types), we ensure wide coverage of the resulting set of questions.

Similar to Chapter 4, we add an answerability & consistency filter on the questions generated from either graph or text of WEBNLG, by posing them to the two `deepset` textual QA models — we then keep only questions where both QA models return an answer which (i) has a confidence score $\geqslant 0.7$, (ii) shares at least a token overlap with the other model's answer, and (iii) has a token overlap with the answer used to condition QG (i.e. a text answer for questions from text; a graph answer for questions from graph). In this way, for all questions that are generated from

[57] https://huggingface.co/valhalla/t5-base-e2e-qg, a T5-base QG model fine-tuned on SQuAD 1.0 data, as well as https://huggingface.co/deepset/roberta-base-squad2 and https://huggingface.co/deepset/deberta-v3-base-squad2 which are RoBERTa/DeBERTa-base QA models that are both fine-tuned on SQuAD 2.0.

a graph, we ensure that they have a matching graph and text answer. We call the resulting dataset — of graph- and text-based (question, answer) pairs, Q-WebNLG^{En}.

5.5 Creating QA/QG Data for Russian and Portuguese

In the second phase, we leverage MT and multilingual KG entity labels from Wikidata to transform Q-WebNLG^{En} into Portuguese and Russian versions.

We describe the process for Portuguese first. We translate both the English texts of WebNLG training data and the set of questions from Q-WebNLG^{En} using a T5 English-Portuguese translation model[58]. We filter these translations using checks (details in Appendix C.1.2) that include back-translation and keeping only those whose automatic scores are above a cut-off. Since we machine translate the questions from English, the semantics of the questions might be affected; as such for each translated question q, we then use a textual QA model trained on Portuguese SQuAD data to obtain the answer to that question from the (automatically translated) Portuguese text (a_t^{PtBr}) and we use heuristics (see Appendix C.1.2) to align this text answer to an entity a_g in the matching graph[59]. We discard any QA pair for which a_g^{PtBr} is different from the graph answer a_g^{en} associated with the original English question (the question q is a translation of).

nf	**Q-K**^{En}	**Q-W**^{En}	**Q-W**^{PtBr}	**Q-W**^{Ru}
1	44,464	19,467	7,557	3,250
2	341,082	61,346	15,463	8,736
3	234,170	25,170	5,446	3,612
4	22,607	13,918	2,411	2,004
5	7,031	-	-	-
TOTAL	1,149,354	119,901	30,877	17,602

Tab. 5.1: **Data Statistics.** Number of questions in the QA datasets; nf: the size of the question (no. of facts)

.

We do the same for Russian, except that we do not translate WebNLG training data to Russian as it is already available (Shimorina et al., 2019). To translate the Q-WebNLG^{En} questions into Russian, we use NLLB-200-1.3B[60] (NLLB Team et al., 2022), an MT model achieving state-of-the-art for 200 languages.

We call the resulting QA datasets, Q-WebNLG^{PtBr} and Q-WebNLG^{Ru}. Table 5.1 summarises their statistics, and example instances of the data can be seen in Table 5.2.

5.6 Multimodal multi-task QG-QA model

We train monolingual models (i.e. one each for Portuguese and Russian) to limit the effect of cross-lingual transfer during training (i.e. only English-Portuguese or English-Russian when generating and answering with graph). Each language-specific model — which is fine-tuned from the public checkpoint of the 300M-parameter mT5-small (Xue et al., 2021) for 10 epochs (to be

[58] https://huggingface.co/unicamp-dl/translation-en-pt-t5
[59] Recall that in WebNLG, each text is paired with a matching graph with semantically equivalent content.
[60] https://huggingface.co/facebook/nllb-200-distilled-1.3B

WebNLG Data			
Graph	[A] < Akita Museum of Art, floor count, 3 > [B] < Akita Museum of Art, opening date, 2013-09-28 > [C] < Akita Museum of Art, address, 1-4-2 Nakadori > [D] < Akita Museum of Art, floor area, 3746.66 (sqm) >		
Text	[E^{En}] The Akita Museum of Art at 142 Nakadori has 3 floors with a total area of 3746.66 square metres and was inaugurated on 28th September 2013.	[E^{Ru}] Акита Музей искусств по адресу 142 Накадори имеет 3 этажа общей площадью 3746,66 квадратных метров и был открыт 28 сентября 2013 года.	[E^{PtBr}] O Museu de Arte de Akita em 142 Nakadori tem 3 andares com uma área total de 3746,66 metros quadrados e foi inaugurado em 28 de setembro de 2013.
Q-W*			
X = graph	[B], [C]		
a_x = *g* ent	Akita Museum of Art		
	{What museum opened in 2013-09-28 in Nakadori?, What is the name of the museum that opened in 2013-09-28 in Nakadori?}	{Какой музей открылся 28 сентября 2013 года в Накадори?, Как называется музей, открывшийся 28 сентября 2013 года в Накадори?}	{O que abriu o museu em 2013-09-28 em Nakadori?, Qual o nome do museu que se abriu em 2013-09-28 em Nakadori?}
X = graph	[B]		
a_x = *g* ent	2013-9-28		
	{In what year was the Akita Museum of Art opened?, Which year was the Akita Museum of Art opened?}	{В каком году был открыт Художественный музей Акиты?, В каком году был открыт Художественный музей Акиты?}	{Em que ano foi aberto o Museu de Arte Akita?, O Museu de Arte de Akita foi inaugurado há algum ano?}
X = text	[E^{En}]	[E^{Ru}]	[E^{PtBr}]
a_x = *t* span	{The Akita Museum of Art}	{Музей искусств Акита}	{Museu de Arte do Akita a 142 Nakadori}
	{What is the name of the museum that has 3 floors with a total area of 3746.66 square metres?}	{Как называется музей, который имеет 3 этажа и общую площадь 3746,66 квадратных метров?}	{Qual o nome do museu que possui 3 andares com área total de 3746,66 metros quadrados?}
X = text	[E^{En}]	[E^{Ru}]	[E^{PtBr}]
a_x = *t* span	{2013}	{28 сентября 2013 года}	{2013}
	{What year was the Akita Museum of Art inaugurated? Which year was the museum inaugurated?}	{В каком году был открыт Художественный музей Акиты?, В каком году был открыт музей?}	{O Museu de Arte Akita foi inaugurado há algum ano?, Qual ano foi inaugurado o museu? }

TAB. 5.2: Q-WEBNLG^{En} instances derived from WEBNLG Data and translated into Portuguese and Russian to give Q-WEBNLG^{PtBr} and Q-WEBNLG^{Ru}. Enclosed letters refer to the triple/text above.

comparable with the baseline) — is trained in a multimodal (graph and text), multi-task (QG and QA) manner, where each training batch contains a mix of all the tasks, i.e. Text QG, Text QA, KG QG and KG QA. This gives us a single unified model that can generate and answer questions from text and from graph.

It takes approximately 20 hours to fine-tune one of our language-specific multimodal multi-task models using a single Nvidia A40 GPU.

▪ **Maximising Coverage** To maximise coverage, we train the QG model to generate multiple questions from the same input, and we extend the set of possible sources for a (question, answer) pair, thereby facilitating question answering. For QG, we gather into a set the questions generated from each X (t or g) in Q-WEBNLG^{PtBr}/Q-WEBNLG^{Ru}, and add to it questions that were generated from other contexts whose semantic content is contained in X. QG coverage is then maximised by gathering all nf-sized questions that shared the same answer in this set — by doing so, each of our QG training instance can then contain multiple questions, enabling the generation of multiple questions from a given (X, a_X, nf) input. For QA, we associate each (q, a_X) pair from a given X to other contexts in the data that encompass X to give $(X, a_X, \overrightarrow{q})$; every (X, a_X, q) triplet in this set is then created as a QA training instance, thereby allowing QA coverage to be increased.

▪ **Handling Unanswerable Questions.** To allow our model to abstain from an answer if the question cannot be answered from the context, we use two strategies (details in Appendix C.2.1) to obtain negative unanswerable $(q, \neg X)$ pairs.

5.7 Evaluation and results

5.7.1 Baseline: single-task models

Dataset	Portuguese	Russian
SQuAD	118,678	50,364
WebNLG	125,957	62,007
TOTAL	244,635	112,371

Tab. 5.3: **Baseline** training data. Number of (t, a_t, q) triplets in the obtained datasets for each language.

The baseline we compare against comprises four different models for each language and is similar to the approach described for the Data-QuestEval metric (Rebuffel et al., 2021) for English, which they have shown to be useful for reference-less evaluation of English RDF-to-Text models. First, textual QG and QA models are trained on SQuAD data for Portuguese and Russian.[61] The textual QG models are then applied to the Portuguese and Russian texts of the WebNLG training data; whereby the entities from the graph that is paired with each text are used to condition QG. This provides (g, a_g, q) triplets that can be used for training the KG QG and KG QA models. Table 5.3 contains statistics about the training data for the baseline. To remain in the same model paradigm as (Rebuffel et al., 2021), every model here is also fine-tuned from mT5-small for 10 epochs.

5.7.2 Evaluating Question Coverage

Using a similar approach to our work in Chapter 4, we use two measures to assess question coverage. We first compare the number of unique questions generated by each approach on the WebNLG^{PtBr}/WebNLG^{Ru} test sets. We further use BERTScore (BSc) (Zhang et al., 2020) to assess their semantic overlap by taking one approach's question for a given X as prediction and the other's generated questions for the same X as multi-references. The intuition is that if approach A scores higher with approach B's questions as references than vice-versa, A's questions are "contained" in B's, and conversely, B has wider semantic coverage.

▪ ***Wider q coverage*** *Our approach has a substantially higher question coverage (up to 4x more for Portuguese and nearly 7x more for Russian) than the baseline (Figure 5.2-5.5). In terms of semantic coverage (Table 5.4), we outperform the baseline in all modalities for Portuguese; we also outperform in graph for Russian, and are on par in the text modality there.*

5.7.3 Evaluating QA Consistency

In a similar vein as in Chapter 4, we evaluate the performance of our approach and the baseline by examining their respective QA consistency performance. While sharing similar approaches, we have to take into consideration evaluation in Russian and Brazilian Portuguese, as well as

[61] For Portuguese we use a version of SQuaD v1.0 (Rajpurkar et al., 2016) we understand is produced with MT and post-edited by native speakers, available at https://huggingface.co/datasets/ArthurBaia/squad_v1_pt_br; for Russian we use the SberQuad (Efimov et al., 2020) dataset.

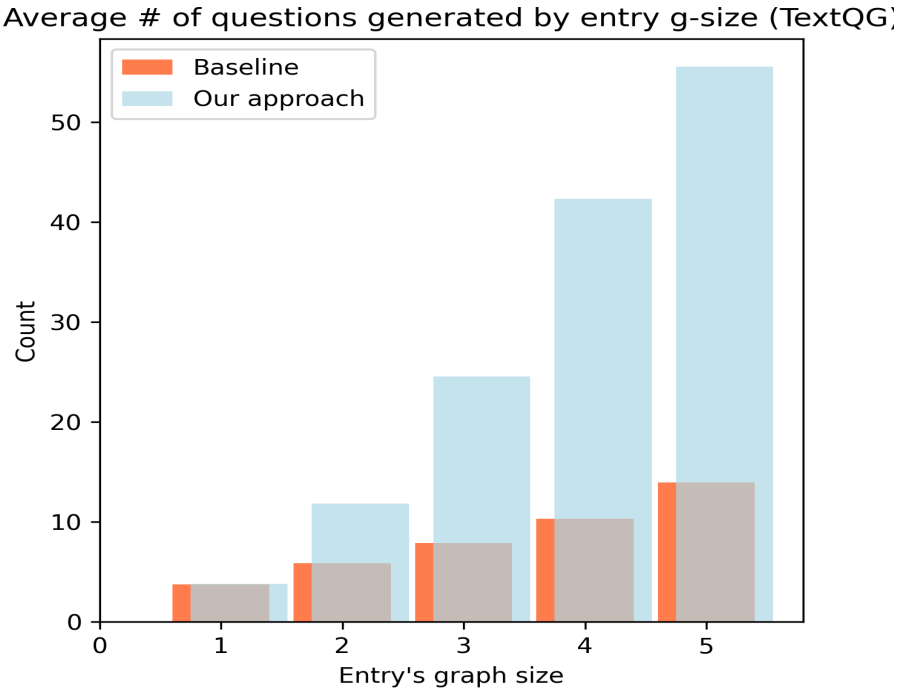


FIG. 5.2: **Portuguese: Comparative QG coverage (Text) — Baseline vs. Our Approach.** The number of generated questions is much higher for our approach across both modality and input size.

Modality	As reference	Portuguese	Russian
Text	Baseline	83.1	**81.3**
	Ours	**85.0**	**81.3**
Graph	Baseline	80.7	75.9
	Ours	**84.6**	**80.1**

TAB. 5.4: Avg BSc, where questions generated by System A for a given g or t are scored against the set of generated questions by System B for the same g or t.

the cross-lingual comparison between English and each of these two languages when answering questions across modalities. This section details how we do so.

In what follows, the answer a_X that is used to condition the generation of q_X is referred to as the ground truth (GT). $\hat{a_X}$ denotes a generated answer from context X. We use superscripts to distinguish outputs from different models — for e.g., $\hat{a}_X^A$ is the answer derived from input context X by model A.

We evaluate the multimodal QG-QA models by computing three consistency metrics that consider the various answers that a question q can be associated with: a_X, the ground truth; $\hat{a}_X$, the answer generated from source X (e.g., a text); and $\hat{a}_{X'}$, the answer generated from the other modality X' (e.g., a graph). *Internal Same-mod (GT)* compares $\hat{a}_X$ with the ground truth answer a_X, indicating the approach's self-consistency. *Internal X-mod (GT)* compares a_X with $\hat{a}_{X'}$ the answer derived from the other modality. *Internal X-mod (Gen Ans)* compares $\hat{a}_{X'}$ and

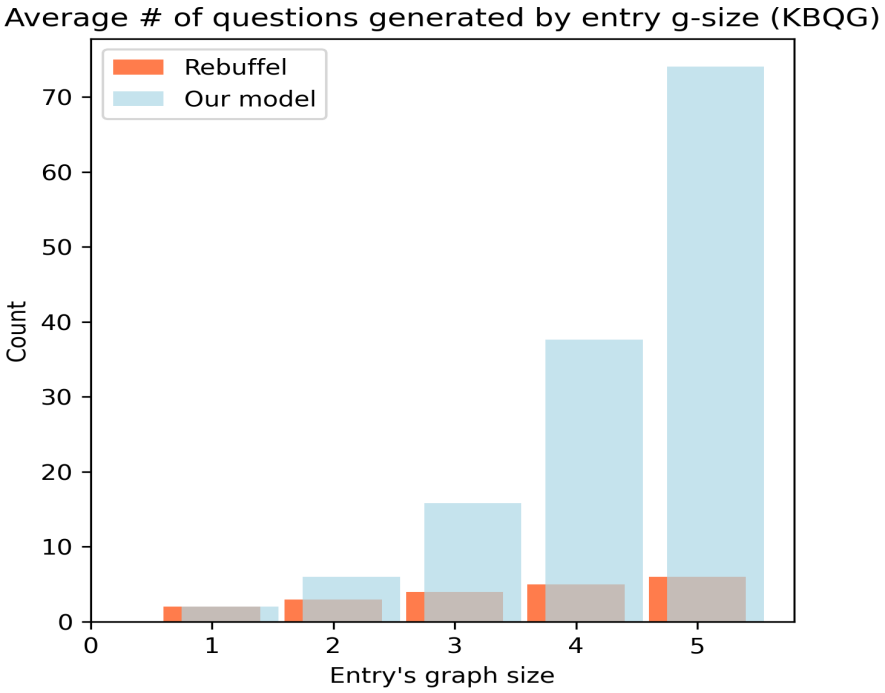


FIG. 5.3: **Portuguese: Comparative QG coverage (KGQG) — Baseline vs. Our Approach.** The number of generated questions is much higher for our approach across both modality and input size.

$\hat{a}_X$, the answers from each modality.

By also posing the questions generated by one approach to the other approach's QA in a cross-approach manner, we gain an external indication of their QG-QA capabilities. For this, we examine the two answers that approach B can generate ($\hat{a}^B_X$ and $\hat{a}^B_{X'}$) when given q^A_X, a question generated by A. We do this on three levels: (i) *X-Appr Same-mod (GT)* compares $\hat{a}^B_X$ against a^A_X on the same modality that q^A_X came from; (ii) *X-Appr X-mod (GT)* compares B's generated answer with the GT answer across modalities (i.e. $\hat{a}^B_{X'}$ vs a^A_X); and (iii) *X-Appr X-mod (Gen Ans)* where $\hat{a}^B_{X'}$ is compared against $\hat{a}^A_X$. A graphical overview of these comparisons is in Figure 5.6.

Both approaches usually generate a different number of questions for a given X; therefore, to ensure a fair evaluation, when an Approach A generates more questions for X, we randomly sample from its set as many questions that Approach B generates for X. We also added a self-consistency filter removing those questions that are unanswerable from their own context. Settings for these are in Appendix C.2.3.

As the token F1 metric commonly used in extractive textual QA cannot account for lexical variation present in cross-modal QA, we follow (Rebuffel et al., 2021)'s use of BERTScore (BSc) for evaluation; however, we also computed token F1 and exact match scores for verification, and these are provided in Tables C.1-C.2 in Appendix C.3.1.

Table 5.5 shows the results of the QA consistency tests for Portuguese and Russian. We observe similar trends for both languages, though the BScs for Russian are noticeably lower under the cross-modal (and cross-lingual too, since a_g is in English) settings. This is likely due to (i) the smaller size of the training data for Russian (Table 5.1), (ii) more frequent transliteration of

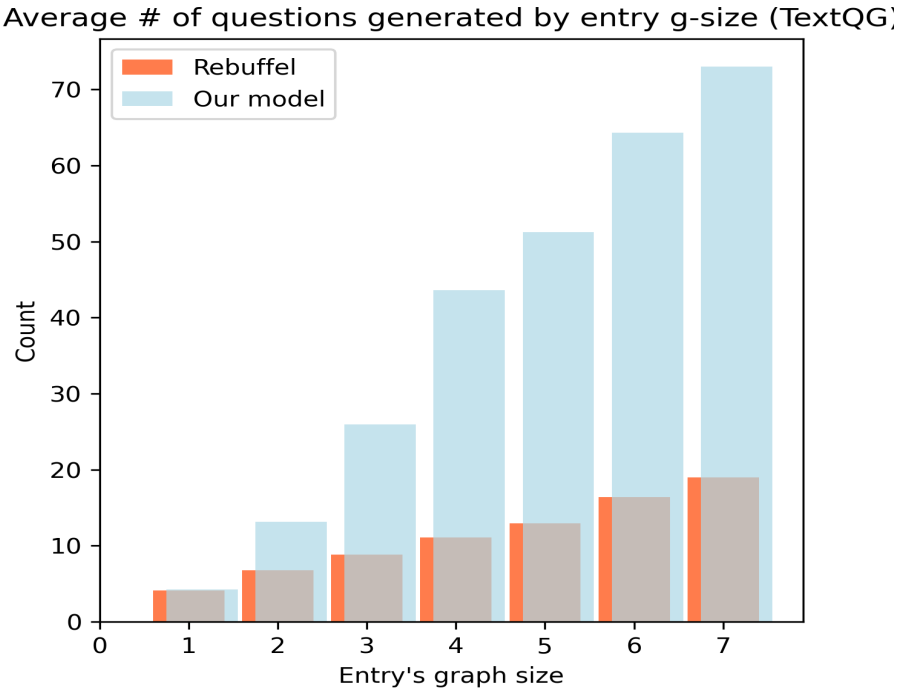


FIG. 5.4: **Russian: Comparative QG coverage (Text) — Baseline vs. Our Approach.** The number of generated questions is also much higher for our approach across both modality and input size.

foreign names into Cyrillic, and (iii) potentially, the freer word order in Russian which relies on case instead of S-V-O order to mark the subject and object (as it is for English and Portuguese).

▪ ***More self-consistent QA*** *Our approach consistently leads to higher scores compared to the baseline in the Internal comparisons for all languages, showing that multimodal, multi-task training using silver aligned data (even at much smaller scale — nearly four times less for Portuguese and more than three times less for Russian; cf. Table 5.1 and 5.3), improves QA self-consistency.*

▪ ***More consistent cross-modal QA*** *We also improve by between 8.9 (29.7 over 20.8, Russian) to 22.9 (65.6 over 43.7, Portuguese) BSc over the baseline in Internal X-mod (GT). This is particularly pertinent for the Data-QuestEval metric, as the metric is computed by comparing the generated answer $a_{X'}$ against the GT answer a_X. Our approach also always leads to gains in Internal X-mod (Gen Ans) over Internal X-mod (GT), which does not happen for the baseline. This means that our QG-QA generates answers that are more consistent between both modalities. This improvement could have come at the expense of the QG-QA's performance vis-a-vis the GT answer (i.e. Internal X-mod (GT)), but this is not the case for our approach, which further validates our multimodal multi-task training to give more consistent cross-modal QG-QA.*

▪ ***QG-QA externally validated*** *Whenever the baseline's QA is used to answer questions generated by our approach, QA accuracy drops significantly less (in some cases, there is a gain) than vice-versa (Table 5.5). This could be predominantly the result of either (i) better QG by our*

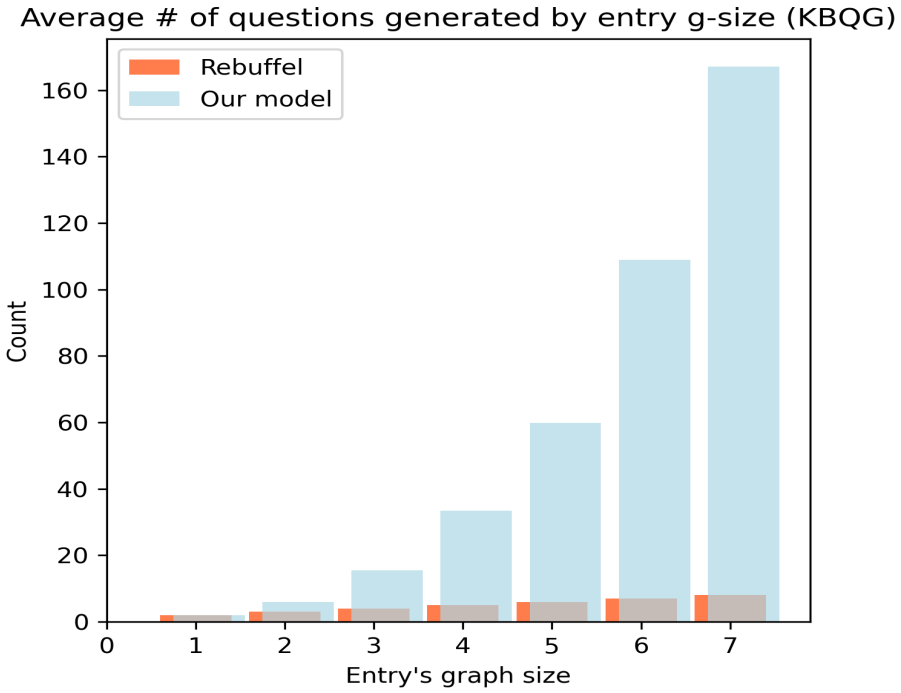


FIG. 5.5: **Russian: Comparative QG coverage (KGQG) — Baseline vs. Our Approach.** The number of generated questions is also much higher for our approach across both modality and input size.

approach; or (ii) better QA by the baseline — however, our approach's stronger performance in all the Internal, X-mod and X-Appr comparisons suggests that the former is the likely factor. This gives an indication of the quality of the questions generated by our approach over the baseline.

▪ ***QA consistent over q size*** *Finer-grained analysis (Table C.3 and C.4 in Appendix C.3.2) shows that our approach's QA performance is consistent across nf, i.e. it is also generating and answering questions of consistent quality across questions of different "sizes" (i.e. simple and complex).*

5.7.4 Data-QuestEval metric

The QA consistency tests above reflect a "gold" setting where both modalities (texts and graphs) are semantically aligned. To evaluate whether our approach brings improvements to settings where this does not hold (i.e. Can it generate and answer questions across modalities where one modality has missing, or additional, information vis-a-vis the other?), we compared it against the baseline when used to compute the Data-QuestEval metric.

We do this by comparing the metric's resulting correlation with human judgments of semantic quality when using each approach. For this, we used the sampled set of system submissions to the Russian RDF-to-Text task in the 2020 WEBNLG Challenge, together with their human

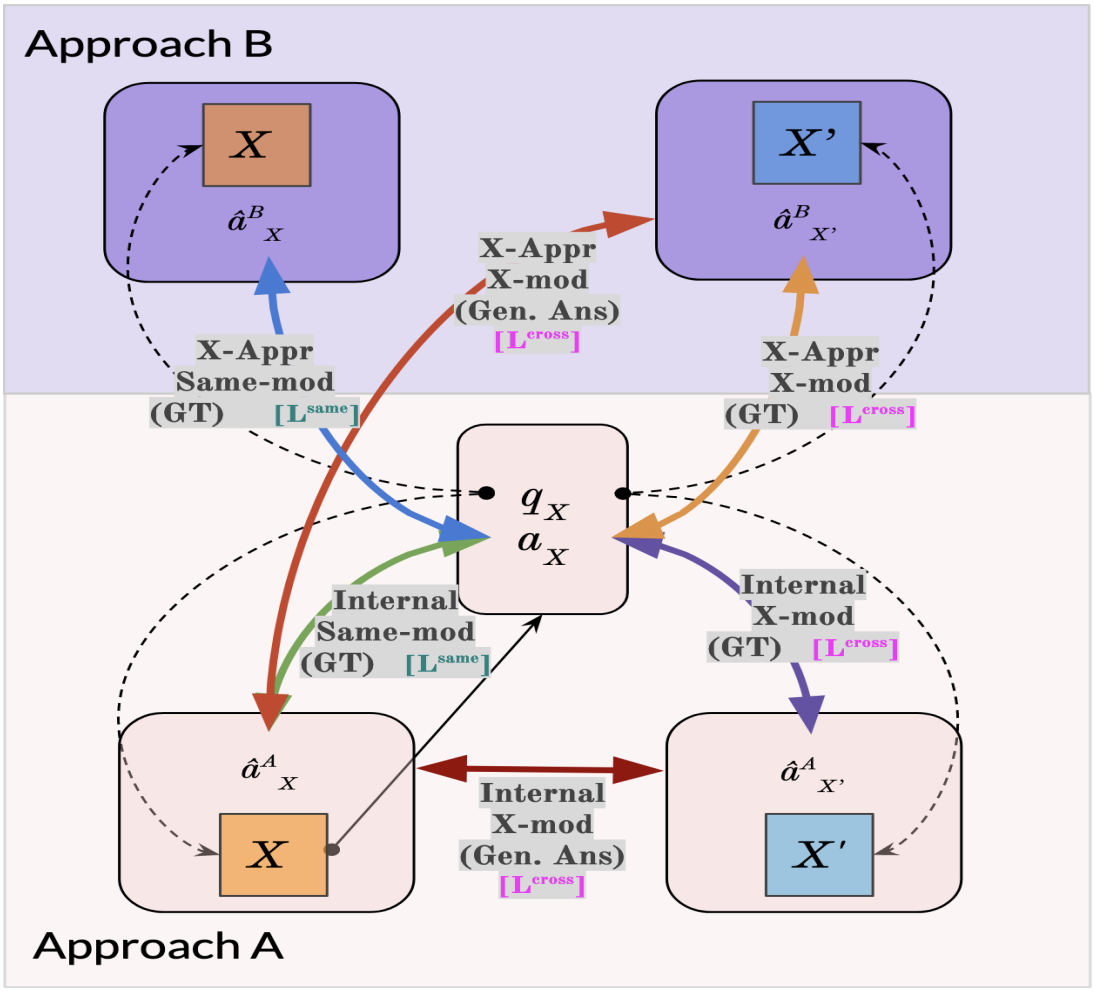


FIG. 5.6: **QA Accuracy**. Bold lines denote QA comparisons within/between modalities and/or approaches. Dotted arrows indicate the context X or X' that the question (q_X) is posed against to obtain the answers. $\mathbf{L}^{cross}$: cross-lingual answer comparison (e.g. PtBr/Ru against En); $\mathbf{L}^{same}$ denote comparison in the same language.

ratings[62], comprising 660 generated texts from six submissions.[63] Since the texts here were machine-generated given a WEBNLG graph input, these texts may contain errors or are ill-formed; therefore the approach giving a higher correlation with the human ratings indicates that it is better able to answer questions that cannot be answered by the other modality (i.e. detect a difference in information content).

▪ ***Better correlations with human judgments*** *Using our approach to compute the Data-QuestEval metric leads to a gain of more than 12 points in its correlation with human judgments (Table 5.6). Together with the more consistent internal and cross-modal QA above, this shows that our approach leads to more robust QG-QA. It also indicates that QA-based reference-free*

[62] https://github.com/WebNLG/challenge-2020

[63] We use the ratings of Data Coverage, Relevance and Correctness (summing their normalised scores). Since the outputs for the challenge's human evaluations were sampled in a random stratified manner, we computed correlation using Pearson's r.

Portuguese

	Internal		X-Appr	
QG QA	Baseline Baseline	Ours Ours	Baseline Ours	Ours Baseline
	Self (GT)			
T → T	$86.3_{(\pm 0.01)}$	**94.7**$_{(\pm 0.14)}$	$73.6^{(\mathbf{-12.7})}_{(\pm 0.10)}$	$60.1^{(-34.6)}_{(\pm 0.25)}$
G → G	$85.4_{(\pm 0.03)}$	**96.8**$_{(\pm 0.05)}$	$58.2^{(\mathbf{-27.2})}_{(\pm 0.12)}$	$61.0^{(-35.8)}_{(\pm 0.53)}$
	X-mod (GT)			
G → T	$43.7_{(\pm 0.15)}$	**65.6**$_{(\pm 0.37)}$	$47.0^{(\mathbf{+3.3})}_{(\pm 0.11)}$	$47.8^{(-17.8)}_{(\pm 0.30)}$
T → G	$40.0_{(\pm 0.08)}$	**62.9**$_{(\pm 0.46)}$	$45.7^{(\mathbf{+5.7})}_{(\pm 0.12)}$	$37.8^{(-25.1)}_{(\pm 0.35)}$
	X-mod (Gen Ans)			
G → T	$39.9_{(\pm 0.14)}$	**67.1**$_{(\pm 0.34)}$	$42.5^{(\mathbf{+2.6})}_{(\pm 0.09)}$	$47.7^{(-19.4)}_{(\pm 0.30)}$
T → G	$40.0_{(\pm 0.08)}$	**65.0**$_{(\pm 0.37)}$	$44.3^{(\mathbf{+4.3})}_{(\pm 0.11)}$	$39.0^{(-26.0)}_{(\pm 0.34)}$

Russian

	Internal		X-Appr	
QG QA	Baseline Baseline	Ours Ours	Baseline Ours	Ours Baseline
	Self (GT)			
T → T	$84.6_{(\pm 0.02)}$	**89.8**$_{(\pm 0.11)}$	$57.8^{(\mathbf{-26.8})}_{(\pm 0.10)}$	$57.2^{(-32.6)}_{(\pm 0.18)}$
G → G	$79.4_{(\pm 0.03)}$	**96.2**$_{(\pm 0.13)}$	$48.9^{(\mathbf{-30.5})}_{(\pm 0.10)}$	$57.1^{(-39.1)}_{(\pm 0.53)}$
	X-mod (GT)			
G → T	$20.8_{(\pm 0.04)}$	**29.7**$_{(\pm 0.26)}$	$24.5^{(\mathbf{+3.7})}_{(\pm 0.06)}$	$20.6^{(-9.1)}_{(\pm 0.21)}$
T → G	$20.2_{(\pm 0.06)}$	**30.0**$_{(\pm 0.08)}$	$17.3^{(\mathbf{-2.9})}_{(\pm 0.03)}$	$21.5^{(-8.5)}_{(\pm 0.06)}$
	X-mod (Gen Ans)			
G → T	$18.6_{(\pm 0.04)}$	**31.1**$_{(\pm 0.27)}$	$22.8^{(\mathbf{+4.2})}_{(\pm 0.06)}$	$20.3^{(-10.8)}_{(\pm 0.23)}$
T → G	$18.4_{(\pm 0.07)}$	**30.8**$_{(\pm 0.10)}$	$15.0^{(\mathbf{-3.4})}_{(\pm 0.03)}$	$22.2^{(-8.6)}_{(\pm 0.06)}$

TAB. 5.5: **Consistency Results**. Average of BScs between answers. In $_{\text{subscripts}}$ are std. dev. across five random runs; in $^{\text{superscripts}}$ are the difference between X-Appr and Internal, the differences provides a meaningful comparison between the baseline and our approach since each of their QA performances are different. Whenever our QA is used, the drop in performance is reduced.

evaluation methods like Data-QuestEval can be improved upon with wider QG coverage.

Baseline	Our Approach
16.4 (4.78E-06)	**28.7** (4.21E-16)

TAB. 5.6: **Correlations (Pearson's r) with human judgments.** The baseline vs our approach used to compute the Data-QuestEval metric. All (p-values) « 0.001.

5.7.5 Multilingual Retrieval-based QA

For further verification of the QG abilities of the baseline and our approach, we also posed the questions generated by each (the same ones as in Table 5.5) to mGEN (Asai et al., 2021). This is a multilingual retrieval-based QA model that was fine-tuned from a mT5-base (Xue et al., 2021) checkpoint to generate the answer to a question given a collection of retrieved input contexts.[64]

Portuguese		
	External Retrieval-QA	
QG	Baseline	Ours
QA	mGEN	mGEN
	Self (GT)	
T → T	$71.4_{(\pm 0.07)}$	**78.0**$_{(\pm 0.58)}$
G → G	$57.7_{(\pm 0.13)}$	**72.8**$_{(\pm 0.48)}$

Russian		
	External Retrieval-QA	
QG	Baseline	Ours
QA	mGEN	mGEN
	Self (GT)	
T → T	$65.9_{(\pm 0.13)}$	**68.5**$_{(\pm 0.18)}$
G → G	$66.9_{(\pm 0.15)}$	**76.0**$_{(\pm 0.77)}$

TAB. 5.7: **QA consistency (BSc)** comparing the QG of Baseline and Ours, using mGEN for QA.

▪ ***Better answerable questions*** *Compared to the baseline, our approach's questions were answered better by mGEN — we see (Table 5.7) an increase of between 2.6 BSc for TextQG (Russian) and 15.1 BSc (Portuguese) in answer accuracy, providing a further independent verification of our approach's QG capabilities.*

5.8 Conclusion

In the work for this chapter, we examined the task of multimodal QG and QA from text and from graph in multiple languages, which has application in QA-based evaluation of text generated from

[64] We use the version of the code and weights that was released as part of the MIA-2022 Shared Task (Asai et al., 2022). See https://github.com/mia-workshop/MIA-Shared-Task-2022/tree/main.

KG. By generating synthetic QA/QG data in English that has questions and answers aligned between text and graph, and using MT, heuristics and quality filters, we obtain "silver" data for Brazilian Portuguese and Russian that enables the training of multimodal multi-task models. Our models provide wider QG coverage, are cross-lingual for QG-QA from KG graphs, and achieve greater internal and cross-modal QA consistency over a baseline that is derived from work in English (Data-QuestEval) (Rebuffel et al., 2021) recently shown to be useful as a metric for evaluating the semantic consistency of RDF-to-Text generations. In fact, using our approach leads to $>$ 12 points gain in the metric's correlation with human judgments for Russian. We also see strong performance in QA accuracy of up to 15.5 BSc when our approach's questions are posed to a multilingual retrieval-based QA model. Holistically, our approach's consistently better performance in coverage and QA consistency demonstrates the improvements that it brings over the baseline.

Conclusion

In this thesis, we have explored the joint generation and answering of questions over texts and knowledge graphs (KG). Firstly, we investigated question generation (QG) from KG input, specifically on using pretrained models for the task and means to controllably generate questions of varied types for a single input KG graph. Next, we investigated controllably generating both simple and complex questions from text and from KG, which can also be answered across these two modalities. Finally, we investigated how to extend the work in the previous two chapters into lower-resourced languages other than English.

9 Summary of findings

The following is a summary of our key findings for the research questions we sought to address in this thesis:

1. What benefits can pretrained language models bring to the task of question generation from KG triples? How might it be leveraged for question-type and focus-controllable QG?

We found (in work described in Chapter 3) that the use of pretrained language models (LMs) helps alleviate the need for complicated solutions (heavy processing, supplementary information and reliance on KG embeddings that are complicated to train and update) used by previous methods to address the semantic gap between KG facts and their NL expressions. We also found that it is possible to generate questions that are of varied types through the finetuning of such pretrained LMs over data that is collected to reflect the variety of: (i) information that can be sought from a single KG fact (the subject or the object; and (ii) ways a question can be phrased. From our downstream evaluations, we also (i) quantified the performance drop in KGQA systems when the distribution of question types in the test data shifts from that of the data used in training the model; and (ii) showed that using our proposed approach for controllable QG for diversity of question focus and variety of question types for training such KGQA models can help mitigate this performance drops due to distribution shifts.

2. How can the comprehensiveness and coverage of QG systems be increased, and can this be done for complex question as well as across modalities for QG together with QA?

In work described in Chapter 4, through the use of silver-quality synthetic QG/QA data for finetuning a pretrained LM in a multi-task manner, we identified an approach to (i) overcome the lack of both data and models for cross-modal QG and QA over text and KG; and (ii) increase the coverage of questions generated from a given input so that it ranges over more of the input and do so in a systematic way. Using a unique consistency evaluation that extensively

investigates QG and QA cross-modally and across systems against a baseline, we show that our proposed approach gives QG and QA that leads to greater consistency. Key to this was (i) a procedure for generating silver-quality synthetic QG/QA instances over KG-text aligned data using round-trip filtering and heuristics to check for answerability over both text and KG graph, and (ii) controllability techniques for the complexity of the question. We also established that increasing the comprehensiveness and coverage of QG contributes to improved performance in QA-based reference-free metrics for evaluating data-to-text NLG outputs.

3. How can cross-modal QG and QA be extended into other lower-resourced languages, and can cross-lingual QG and QA between modalities be carried out?

In work described in Chapter 5, we show that it is possible to leverage a multilingual pretrained LM finetuned on synthetic data to improve the performance of multi-modal QG and QA over text and KG for languages outside of English. We show this for two lower-resourced languages, Russian and Portuguese Brazilian, and do it via an approach for generating silver-quality data using machine translation and cross-modal answer verification across English and the other languages. In doing so, we show that it is feasible to carry out QA cross-lingually (between English and Russian/Portuguese Brazilian), which is meaningful when seeking to verify information over text and KG, where most of the entity labels in the latter remain largely in English. We also show that this approach enables better cross-modal cross-lingual QG and QA over existing methods, which can be a means for extending QA-based evaluation for data-to-text generation outside of English.

10 Future directions

Wider variety of complex questions The work in this thesis have been focused on factual questions. There are, however, other types of questions that we as humans use and encounter, including other complex ones (see Table 2.1 in Chapter 2) such as those of the casual (antecedent and consequent) enablement, and instrumental/procedural variety. To be able to generate and answer such types of complex questions too will facilitate advancements for NLP tasks as well as do so for a wider set of domains. For instance, in technical or specialised domains where the information communicated and sought for often includes "why" and "how to" information (e.g. texts such as scientific papers, legislative documents, strategic plans, instruction manuals/guides), models that can generate and answer questions for such information can be very helpful for building agents that can assist us in knowledge-intensive work. One challenge is that such questions have answers that often require some abstraction over multiple concepts on the part of the answerer (as compared to identifying/extracting entities or values), with a wider variety of phrasing and structuring possible, thereby potentially needing innovative methods for evaluating the generating and answering of such questions.

Additional modalities While our focus have been on the text and KG modalities in this thesis, similar approaches and frameworks can be established for other structured and unstructured modalties such as tables, charts, conversations, images etc. Work already exists for verifying the information content over images (Reddy et al., 2021; Prasad et al., 2024), but the ability to extend to more modalities as well as across more than two modalities simultaneously can be beneficial. This is especially so in the face of the rapidly increasing capabilities of text, speech, image and video generation models and steadily decreasing barriers of access to them. To have more capable models like the ones in Chapters 4 & Chapter 5 can help provide us with technolo-

gies to counter potentially ill-intended usage of such generative models. Here, a challenge lies in the ability to execute mulit- and cross-modal question generation and answering reliably and efficiently at scale.

Questions-based semantic representation of information Together, the two preceding directions could facilitate efforts for using questions as a semantic representation for information; one that machines can reason effectively over but where humans can readily interpret. More generally, the ability to form an ordered answerable set of questions that comprehensively and systematically cover the content present in a piece of information (that could be in any modality; discourse, dialogue, images etc), can provide a form of structured semantic representation that is based on semi-structured units (NL QA pairs). Such an approach would permit leveraging both symbolic and neural methods for NLP tasks. The challenge here is to be able to extract QA pairs that draw connections (that could be explicit, or more often than not implicit) across a wider context – such as a document, as well as account for the variety of ways in which a communicator can choose to structure her/his message.

A

Appendices for Chapter 3

A.1 Mapping the WebNLG DBpedia triples to Wikidata

The approach for the migration is similar to that for SQ. We leveraged the "schema:about" property to map DBpedia entities into Wikidata. Specifically, we started by constructing SPARQL queries to return the set of all properties between each entity-pair (where both subject and object are entities as well as where the object is a literal in the forward and backward directions).

For the set of entity-pair/entity-literal pairs with at least one Wikidata property found between them, we group them by their original DBpedia property (we refer to one of these as a DBP property cluster). Given WQ's smaller size, we directly identified the most common Wikidata property found in each DBP property cluster. We used a similar manual inspection approach as SQ above (namely, manually inspect: (a) one entity-pair from the cluster, (b) its DBpedia triple; and (c) the Wikidata property label.) and only mapped the DBpedia property to this most common Wikidata property if (a), (b) and (c) above are semantically aligned. This process was also repeated in the backwards direction. Next, we used the mapping from these assignments (i.e. complete WQ triples found in Wikidata) for all other instances where the set of their entities and properties appear in WQ.

For the remaining unmapped WQ DBpedia properties, we wrote a SPARQL query with a contain function for the DBpedia property (with camel case removed and lowercased) on the English label and alternative labels for all the properties in Wikidata (approx 8,000) to identify candidate mappings. We manually inspected these candidates and assigned using the same criteria (steps a,b,c above) if it is semantically aligned with the DBpedia property. We took care to check the direction of the Wikidata property and reversed entity-pair/entity-literal-pairs that have this DBpedia property. A further set of properties in WQ (<25) remained unmapped after this, and a manual search of the Wikidata site was done to identify a suitable mapping.

For the remaining WQ triples not fully mapped (but with their DBpedia property mapped), we used "schema:about" to map their entities into Wikidata to complete the triple's mapping, with a fallback to search for matches of the entity (replacing the underscores with spaces and maintaining case) against the English labels and alternative labels for all entities in Wikidata. In the latter case, where there are multiple candidates, we select the first matched Wikidata entity (Q-code sorted by lexicographic order). Finally, for the remaining unmapped entities, we conducted manual searches (with the help of the `wikipedia` package to identify candidates) on the Wikipedia and Wikidata sites to attempt to map these entities. Unlike SQ above - where we only mapped the triple into Wikidata if both entities in the SQ triple are found in Wikidata, we accepted DBpedia entities that could not be found in Wikidata; for these, we removed the camel case of the DBpedia entity and used them as the entity's mapping. Although the entity may not currently be present in Wikidata, its presence in DBpedia suggests that it is attested in Wikipedia and could be added to Wikidata.

A.2 Combining the three datasets

Table A.1 contains the results of our experiments where we fine-tune on the combination of the three datasets (SQ, WQ and ZQ) and evaluate the model on each of their test set.

Model	SQ	WQ	ZQ	ALL
BLEU-4				
BART$_{rdf,qt}$	41.95	40.55	49.72	48.24
BART$_{rdf,qt,wkdqg}$	41.31	37.60	49.71	48.05
BERTScore				
BART$_{rdf,qt}$	73.51	68.92	75.72	75.17
BART$_{rdf,qt,wkdqg}$	72.63	68.06	75.57	74.89
ROUGE-L				
BART$_{rdf,qt}$	71.21	63.86	71.26	71.03
BART$_{rdf,qt,wkdqg}$	70.28	62.80	71.01	70.65
METEOR				
BART$_{rdf,qt}$	36.78	34.95	40.78	39.99
BART$_{rdf,qt,wkdqg}$	36.62	33.40	40.59	39.76

TAB. A.1: Results (automatic metrics) for RDF-only models. BART$_{rdf,qt,wkdqg}$: model fine-tuned on the WKDQG data and evaluated on each of the SQ, WQ and ZQ test sets. ALL shows the results proportionally averaged on the three test sets.

A.3 Examples: models' inputs and generation outputs

Table A.2 provides an overview of the format of the inputs to each of the models (Elsahar and ours) under various settings. The original SQ input and reference is provided, as well as examples of each model's generated output.

SimpleQ	**Input:** ⟨ M/03Y2SVR , ASTRONOMY/CELESTIAL_OBJECT/CAT... , M/0JVQ ⟩
Sample	**Reference** what category of celestial object is 7624 gluck

Model	**Training/inference input format**	**Training target format**	**Generated**
Elsahar	⟨ M/03Y2SVR , ASTRONOMY/CELESTIAL_OBJECT/CATEGORY , M/0JVQ ⟩, celestial object, category, PLACEHOLDEROBJ designated as PLACEHOLDERSUB	what OBJTYPE of SUBTYPE is PLACEHOLDER-SUB	what is 7624 gluck (relexicalised from system decoder output: what is PLACEHOLDER-SUB)
BART$_{rdf}$	⟨ 7624 GLUCK , INSTANCE OF , ASTEROID ⟩, ANSOBJ	what category of celestial object is 7624 gluck	what type of celestial object is 7624 gluck
BART$_{rdf,mtl}$	⟨ 7624 GLUCK , INSTANCE OF , ASTEROID ⟩, ANSOBJ, category	ditto	what type of celestial object is (7624) 1979 gd1
BART$_{rdf,qt}$	⟨ 7624 GLUCK , INSTANCE OF , ASTEROID ⟩, WHAT, ANSOBJ, category	ditto	what type of celestial object is 7624 gluck
BART$_{rdf+nl}$	⟨ 7624 GLUCK , INSTANCE OF , ASTEROID ⟩, ANSOBJ, asteroid designated as 7624 Gluck	ditto	what kind of celestial body is 7624 gluck
BART$_{rdf+nl,mtl}$	⟨ 7624 GLUCK , INSTANCE OF , ASTEROID ⟩, ANSOBJ, category, asteroid designated as 7624 Gluck	ditto	what type of celestial body is 7624 gluck
BART$_{rdf+nl,qt}$	⟨ 7624 GLUCK , INSTANCE OF , ASTEROID ⟩, WHAT, ANSOBJ, category, asteroid designated as 7624 Gluck	ditto	what kind of celestial object is 7624 gluck
BART$_{rdf,qt}$, zero-shot property	⟨ 7624 GLUCK , INSTANCE OF , ASTEROID ⟩, WHAT, ANSOBJ, category	ditto	what is the category of 7624 gluck
BART$_{rdf+nl,qt}$, zero-shot property	⟨ 7624 GLUCK , INSTANCE OF , ASTEROID ⟩, WHAT, ANSOBJ, category, asteroid designated as 7624 Gluck	ditto	what is the category of 7624 gluck

TAB. A.2: Examples of system outputs (**Generated**), used in the automatic evaluation. Other columns compare the formats for the input and target between the models (Elsahar and ours).

B

Appendices for Chapter 4

B.1 Detailed results

▪ **Finer-grained analysis of QTT QA performance** Table B.1 provides a finer-grained view of QTT's same-mod and cross-mod QA consistency performance; it is evaluated for QA performance on the set of questions relating to varying number of facts ($1 \leqslant nf \leqslant 4$).

	Num Facts			
	1	2	3	4
	Same-mod (GT)			
T → T	96.3	98.5	98.3	98.1
G → G	97.8	99.1	99.2	99.3
	X-mod (GT)			
T → G	77.7	77.9	77.7	77.2
G → T	73.2	77.9	75.1	74.6
	X-mod (Gen Ans)			
T → G	79.2	77.9	78.1	77.1
G → T	74.3	78.2	75.5	75.0

Tab. B.1: **Fine-grained analysis of QTT's QA performance (BSc). Num Facts** denote the number of facts (nf) the set of QA-pairs relate to (i.e. 1 denotes an SQ of 1 fact, 2 denotes a CQ of 2 facts etc...). The nf sets are mutually exclusive.

B.2 Implementation details: CQ-Gen and SQ-Gen

B.2.1 Training data

Q-KELM is used as the training data for the CQ-Gen and SQ-Gen models (respectively using CQ-KELM and SQ-KELM from Q-KELM). Each training instance in CQ-Gen and SQ-Gen is assembled in the manner described in the following paragraphs; examples for them can be found in Table 4.3 and B.2.

▪ **Context** X When generating from text t, the entire t is used as input for both CQ-Gen and SQ-Gen. For generation from KG graph, the input to both models is g', a graph of g of size nf where $1 \leqslant nf \leqslant 4$. For SQ-Gen, the size of g' is always 1; for CQ-Gen, it is $2 \leqslant nf \leqslant 4$.

▪ **Answer** a_X For generation from text, a_t the text answer that was extracted by the RoBERTa QA model (see Section 4.2.1) is used. When generating from KG graph, the graph answer a_g (the entity/value in g that is aligned with a_t (see below) is used.

▪ ▪ ***Aligning*** a_g The graph answer a_g is the entity in g which either exactly matches, contains or else has the smallest edit distance to a_t. If a_t cannot be matched to a graph entity, the (q, a_t) is rejected, together with those unanswerable by the QA model.

▪ **Answer semantic type** a_{type} The answer entity semantic type a_{type} is used for QG when generating from text as well as from graph. It is obtained by using the graph entity (a_g) aligned

with the answer by querying Wikidata for the set of entities/values that a_g has the 'instance of' and/or 'subclass of' property with.

▪ **Question complexity control** nf In CQ-KELM and SQ-KELM each (q_X, a_X) pair is associated with a graph g' obtained with heuristic matching (see below). Since g and t are parallel, nf (the size of g') is used as the control for the complexity of the question for generation from both text and graph.

▪ ▪ ***Determining*** nf The size (nf) of the question is determined by matching a question and its answer to the corresponding graph $g' \subseteq g$ where g' is the set of triples $\langle s, p, o \rangle$ in g such that: either s and o have an overlap of $\geqslant 1$ token with $q + a_t$, and/or they can be detected in $q + a_t$.

▪ **Controls** A textual prompt is added to both the input and the target to control the question generation; the prompt differs for CQ-Gen and SQ-Gen. SQs are those where $q + a_t$ contain only two entities, whereas we define CQs as those with at least seven tokens and such that $q + a_t$ contain > 2 entities, and the size of the matching graph is at least 2 (i.e. $nf \geqslant 2$). A set of special tokens (added to the T5 tokeniser vocabulary) is used to demarcate the components in the input and target. These prompts and tokens can be seen in the examples found in Tables 4.3 and B.2.

▪ **Question type** q_{type} The question type, which is only used in SQ-Gen, is detected using the question type filter in the NL-Augmenter framework (Dhole et al., 2021) [65].

▪ **Target** The target is a single question for a given input for both CQ-Gen and SQ-Gen.

Modality		
Text	Input	generate 1 simple question of 1 fact from text [ANS] 1977 [QST] when [Sp2] time [INP] The Lasse Viren Finnish Invitational, which was created in 1977, is part of the sport of athletics.
	Target	sq 1 text 1 [ANS] 1977 [QST] When was the Lasse Viren Finnish Invitational created?
Graph	Input	generate 1 simple question of 1 fact from rdf [ANS] Pennsylvania Avenue [QST] where [Sp2] street [INP] [SUB] National Archives Building [PRP] located on street [OBJ] Pennsylvania Avenue
	Target	sq 1 rdf 1 [ANS] Pennsylvania Avenue [QST] Where is the National Archives Building located?

Tab. B.2: **SQ-GEN** Examples of inputs and targets for simple questions training instances for text and for graph. [ANS], [INP], [QST], [Sp1], and [Sp2] are special tokens we use to demarcate parts of the input/target. [SUB], [PRP] and [OBJ] are used in graph inputs to demarcate subject, property and object elements of a triple.

[65] https://github.com/GEM-benchmark/NL-Augmenter

B.2.2 Technical details

CQ-Gen and SQ-Gen are each T5-base pretrained models that were fine-tuned from their public checkpoints. Each of them was tuned for up to 10 epochs with early stopping (on the loss for the validation set for when it stops decreasing, with a patience of 3 epochs). A learning rate of 2e-4 was used, together with a linear warmup ratio on 10% of the total training steps, and an effective batch size of 144. For both models, we used the HuggingFace `transformers` library. We used the Lightning integration of the DeepSpeed framework for efficient training and used bf16 precision together with the DeepSpeedCPUAdam optimizer.

B.3 Implementation details: QTT

B.3.1 Training data

QTT-Data was used as the training data for QTT. Examples of the training instances for each task/subtask can be found in the following tables: TextQG and KGQG can be found in Tables B.3 and B.4, complex and simple TextQA as well as KGQA can be found in Table B.5, the KG-to-Text as well as Text-to-KG tasks can be found in Table B.6; and the EntType tasks can be found in Table B.7.

Obtaining sequence of questions for QTT QG

For questions grounded in text, since each WebNLG text t is associated with a graph g, we use it to gather the set of questions in QTT-Data generated from g itself and its sub-graphs (and the corresponding sub-texts of t that is present in WebNLG). From this set, we gather all nf-sized questions which share an a_t to form the set of questions associated with (t, a_t, nf).

For questions grounded in graphs, two treatments are used to gather questions since the inputs to QTT differ for the generation of CQs and SQs (see Appendix B.3.2). For CQs, given g', which is a graph of g (a graph occurring in WebNLG), we gather all questions with size nf and answer a_g that are associated with g'. For SQs, given a WebNLG graph g, we gather all questions of size one and answer a_g, which can be computed from a triple contained in g.

Finally, the target in the TextQG and KGQG tasks is typically a set of 3 questions drawn from $\overrightarrow{q}$ without replacement. If $\mid \overrightarrow{q} \mid > 3$, $\overrightarrow{q}$ is padded to ensure $\mid \overrightarrow{q} \mid \% \, 3 = 0$. If, however, $\mid \overrightarrow{q} \mid \leqslant 3$ — as it happens in the complex KGQG setting — the sequence of questions in the target is simply $\overrightarrow{q}$. Each set of $\leqslant 3$ questions is then instantiated as a new training instance. This is done in order to train the QG models to generate multiple questions for a given context while still staying within the T5 model's maximum input length.

Deriving and maximising QA data

We derive QA data by creating for each $(X, a_X, \overrightarrow{q})$ tuple in QTT-Data as many QA training instances of the form (X, a_X, q) as there are questions in $\overrightarrow{q}$.

In addition, if a question q with answer a_t is answered by a text t (resp. graph g by a_g) which is strictly contained within another larger text t^+ (resp. g^+) in WebNLG, we also associate (q, a_t) with t^+ (resp. (q, a_g) with g^+).

Creating EntType instances

The target for the **EntType** auxiliary task is the set of entities and values (*eset*) detected in the input context, and each of the $e \in eset$ is paired with their semantic types.

For text, $eset_t$ is detected by applying the `BLINK` entity linker for text (Wu et al., 2020), and `Duckling` on the text t. For entities, the semantic type information is retrieved from Wikidata (from the RDF dump dated 29 December 2021). Any type that contains the strings "MediaWiki", "Wikimedia" or "disambiguation" are excluded. For values, we use the type information predicted by `Duckling`.

For graph, the set of entities/values ($eset_g$) that are present in g is used. For values in g, such as dates, times, monetary sums etc we leverage the predictions from `Duckling` (that was applied on the t that is associated with g). We identify from the `Duckling` prediction set the one: with the lowest edit distance to the value, which is an alphanumeric string, and has a normalised edit distance < 0.5 (if there is one such).

Finally, we exclude the following from the training instances: (i) when an answer semantic type for an entity cannot be found; and (ii) when the difference in the number of entity/values sets between the g and its t is more than 2. The latter is so as to avoid semantically similar g and t instances having significantly different targets, which is particularly important since we train in a multi-task setting where all tasks are seen simultaneously.

Negative sampling for QA

For the TextQA and KGQA tasks, negative examples were created so as to allow the model to recognise questions that are unanswerable given the context. These samples are created using a random (i.e. 50-50) assignment to one of these two strategies:

- **Strategy 1: simple negatives** for a given (X, q, a_X) instance are created by picking another (X_{other}, q', a'_X) instance in the QA training set where X and X_{other} do not share any common entities/values between them. If X is a text, we use the graph that it is paired with in WebNLG to check for common entities. A new instance $(X, q',$ "unanswerable") is then created in the training data.

- **Strategy 2: hard negatives** are created in the following manner: a mapping M is first created ahead of time where every entity e that can be found in the training data is associated with the set of all the (X, q, a_X) instances where e is mentioned in X ($e \in X$). If X is a text, we use the graph it is paired with in WebNLG when creating M.

  For a given (X, q, a_X) instance, another instance, i.e. $(X_{other}, q', a_{X_{other}})$ is randomly chosen using M and a random $e \in X$. If a single token from the answer $a_{X_{other}}$ overlaps with (i) any of the tokens for any e (an entity or value) found in X or (ii) any of the tokens of X, then this $(X_{other}, q', a_{X_{other}})$ candidate is rejected. Otherwise, a new instance $(X, q',$ "unanswerable") is created in the training data using the instance and the process for (X, q, a_X) terminates. When a candidate instance is rejected, a new $(X_{other'}, q', a_{X_{other'}})$ is drawn from M using another $e \in X$. After 10 tries or when all $e \in X$ has been exhausted, a simple negative is created instead.

Upsampling

We carry out upsampling on two levels when preparing QTT-Data (the training data for QTT).

▪ **Between modalities for QG subtasks and between modalities for the same task** For QG, this is done between complex TextQG and complex KGQG, simple TextQG and simple KGQG to ensure that the QG subtasks are balanced. It is also done between modalities for the QA tasks (e.g. TextQA and KGQA) to ensure that the tasks are balanced between modalities. The negative samples for the QA tasks are also upsampled in the same way.

For instance, m_1 and m_2 are the sets of samples for modality 1 and modality 2 respectively, and suppose $\mid m_1 \mid > \mid m_2 \mid$.

If $\mid m_2 \mid / \mid m_1 \mid \geqslant 0.5$, we randomly sample $\mid m_1 \mid - \mid m_2 \mid$ from m_2 to balance them.

If however $\mid m_2 \mid / \mid m_1 \mid < 0.5$, we upsample m_2 up to at most $1/3 \cdot \mid m_1 \mid$. This is to ensure that we do not overrepresent m_2 in the data and overfit on it during training.

▪ **Globally for certain tasks** This was done for simple QG as a whole (i.e. after simple TextQG and simple KGQG have been consolidated as one), KG-to-Text, Text-to-KG, and Ent-Type tasks. This is because there are significantly fewer instantiated samples of these tasks in the data than in the rest. The number of samples for each of these tasks was tripled.

B.3.2 Technical details

▪ **QA with FiD** We fine-tune with the base version of the publicly released FiD checkpoint[66] that is trained on the TriviaQA dataset (Joshi et al., 2017). Using the training data for either QTT (i.e. QTT-Data) or DQE (Section 4.4), we further fine-tune this checkpoint for 15,000 steps to give four different models (FiD_t^D trained on DQE's training data for text; FiD_g^D trained on DQE's synthetic training data for graph; FiD_t^Q and FiD_g^Q trained with QTT-Data with the appropriate context X used, i.e. text and graph respectively). For evaluation, we use the same set of generated questions (q_X) and reference answers, i.e. the answer used to condition QG (a_X) produced by each of DQE and QTT; this is the same set of questions and reference answers used to compute the results in Table 4.6

▪ **Correlations** Following (Rebuffel et al., 2021), both the Spearman's ρ and Pearson's r correlations in this paper were computed with the SciPy python library (Virtanen et al., 2020)

▪ **Text answer selector model** This is a T5-base model that is fine-tuned on the task of answer selection on text. The input to the model is a text t, and the target is a set of answer spans in QTT-Data that are in t. The answer spans comprise: (i) the set of all a_t in QTT-Data for a given t, together with the set of all mention spans obtained from `BLINK`/`Duckling` (which were also used in the **EntType** auxiliary tasks). The answer spans are sorted by the sequence of appearance in t. The model was trained for 10 epochs with early stopping (patience of 3 epochs) on the development set loss. A batch size of 32 and a learning rate of 2e-4 (with a linear warmup of 10% of the training steps) was used.

[66] https://dl.fbaipublicfiles.com/FiD/pretrained_models/tqa_reader_base.tar.gz

Modality		
Complex TextQG	Input	cqg task [Sp1] text 2 [ANS] Aarhus University [INP] The School of Business and Social Sciences at Aarhus University (in Denmark) is affiliated to the European University Association (HQ in Brussels). Denmark's leader is Lars Lokke Rasmussen.
	Target	cqg task [Sp1] text 2 [ANS] Aarhus University [QST] Lars Lokke Rasmussen is the leader of Denmark, which is the home of the School of Business Studies and Social Sciences at what university? [QST] The School of Business and Social Sciences at what university is affiliated to the European University Association? [QST] What is the name of the university in Denmark that is home to the School of Business and Social Sciences?
Complex KGQG	Input	cqg task [Sp1] rdf 3 [ANS] 28.0 (metres) [INP] entity [3Arena], height [28.0 (metres)], located in the administrative territorial entity [North Wall Quay], building type [Concert and events venue] [TEND]
	Target	cqg task [Sp1] rdf 3 [ANS] 28.0 (metres) [QST] What is the height of the concert and events venue in North Wall Quay?

Tab. B.3: **QTT** Examples of inputs and targets for **complex** TextQG and KGQG training instances. [ANS], [INP], [QST], [Sp1], and [TEND] are special tokens we use to demarcate parts of the input/target.

Modality		
Simple TextQG	Input	sqg task [Sp1] text 1 [ANS] 1985 [INP] 200 Public Square, Cleveland, with 45 floors covering 111484 square metres, was completed in 1985
	Target	sqg task [Sp1] text 1 [ANS] 1985 [QST] In what year was 200 Public Square completed? [QST] How many years was 200 Public Square completed? [QST] Which year was 200 Public Square completed?
Simple KGQG	Input	sqg task [Sp1] rdf 1 [ANS] Abraham A. Ribicoff [INP] entity [Abraham A. Ribicoff], spouse [Ruth Ribicoff][TEND]
	Target	sqg task [Sp1] rdf 1[ANS] Abraham A. Ribicoff [QST] What was Ruth Ribicoff's husband's name? [QST] Who was Ruth Ribicoff's husband? [QST] What is the name of the man Ruth Ribicoff married to?

Tab. B.4: **QTT** Examples of inputs and targets for **simple** TextQG and KGQG training instances. [ANS], [INP], [QST], [Sp1], and [TEND] are special tokens we use to demarcate parts of the input/target.

Modality		
Text	Input	textqa task [Sp1] Alaa Abdul Zahra has played for AL Kharaitiyat SC and for what sports club of the Qatar Stars League? [INP] Alaa Abdul Zahra has played for AL Kharaitiyat SC and for Al-Khor Sports Club of the Qatar Stars League. Al Kharaitiyat SC play at Al Khor and are managed by Amar Osim.
	Target	textqa task [Sp1] Al-Khor Sports Club
	Input	textqa task [Sp1] What is the name of the monument that was constructed in 2000 in Adams County, Pennsylvania? [INP] In Soldevanahalli, Acharya Dr. Sarvapalli Radhakrishnan Road, Hessarghatta Main Road, Bangalore – 560090 is the location of the Acharya Institute of Technology in India which is affiliated with the Visvesvaraya Technological University. The motto of the Institute which was established in the year 2000 is "Nurturing Excellence" and there are 700 postgraduate students.
	Target	textqa task [Sp1] unanswerable
Graph	Input	kbqa task [Sp1] In what city in Switzerland is the accademie di architettura di mendresio located? [INP] entity [Accademia di Architettura di Mendrisio], country [Switzerland], number of students [600], established [1996], city [Mendrisio], location [Ticino] [TEND] , entity [Switzerland], leader [Johann Schneider - Ammann] [TEND]
	Target	kbqa task [Sp1] Mendrisio
	Input	kbqa task [Sp1] In what year was the Ataturk Monument opened? [INP] entity [Turkey], leader title [President of Turkey], leader [Ahmet Davutoglu], largest city [Istanbul], currency [Turkish lira] [TEND] , entity [Ataturk Monument (Izmir)], designer [Pietro Canonica], material [Bronze], location [Turkey] [TEND]
	Target	kbqa task [Sp1] unanswerable

TAB. B.5: **QTT** Examples of inputs and targets for TextQA and KGQA training instances. [INP], [Sp1], and [TEND] are special tokens we use to demarcate parts of the input/target.

Modality		
KG-to-Text	Input	data2text task [Sp1] entity [Allama Iqbal International Airport], operating organisation [Pakistan Civil Aviation Authority], location [Punjab, Pakistan], city served [Lahore][TEND] , entity [Pakistan], leader [Mamnoon Hussain] [TEND] , entity [Punjab, Pakistan], country [Pakistan] [TEND]
	Target	data2text task [Sp1] Allama Iqbal International airport is located in Punjab, Pakistan and is operated by The Pakistan Civil Aviation Authority. Lahore city is served by the airport. Mamnoon Hussain is the leader of Pakistan.
Text-to-KG	Input	text2data task [Sp1] A Long Long Way was written in Ireland and published by Penguin Random House (parent company Viking Press).
	Target	text2data task [Sp1] entity [A Long Long Way], country [Ireland], publisher [Viking Press] [TEND] , entity [Viking Press], parent company [Penguin Random House] [TEND]

Tab. B.6: **QTT** Examples of inputs and targets for KG-to-Text and Text-to-KG training instances. [INP], [Sp1], and [TEND] are special tokens we use to demarcate parts of the input/target.

Modality		
EntType_t	Input	entlink task [Sp1] Abner currently plays football for Real Madrid Castilla.
	Target	entlink task [Sp1] [INP] Abner [Sp3] Homo sapiens; person; natural person; omnivore [INP] Real Madrid Castilla [Sp3] football team; sports team
EntType_g	Input	entlink task [Sp1] entity [Buzz Aldrin], place of birth [Glen Ridge], country of citizenship [United States of America], status [Retired] [TEND]
	Target	entlink task [Sp1] [INP] Buzz Aldrin [Sp3] Homo sapiens; person; natural person; omnivore [INP] Glen Ridge [Sp3] borough in the United States; municipality of New Jersey [INP] United States of America [Sp3] state; country; political territorial entity; republic; federation; historical country; state with limited recognition; democracy; nation

Tab. B.7: **QTT** Examples of inputs and targets for EntType (on text and on graph) training instances. [INP], [Sp1], [Sp3], and [TEND] are special tokens we use to demarcate parts of the input/target.

C

Appendices for Chapter 5

C.1 Data details

C.1.1 Synthetic QA data in English

We used a subset of KELM filtered for (g, t) pairs where g has between 2 and 5 triples; this is because larger sizes typically lead to unnatural questions. KELM (Agarwal et al., 2021) has 15M (g, t) pairs of which we used a subset of about 2M pairs to seed the generation of synthetic QA data on; about 1.1M texts remain in Q-KELM^{En} after our QA pair filtering steps. The distribution of Q-KELM^{En}'s questions by nf can be found in Table 5.1.

C.1.2 Synthetic QA data in Portuguese and Russian

- **Quality filters for machine translation**

- **Verifying MT on Portuguese texts** To ensure the quality of the machine translations of the texts from English to Portuguese, we backtranslated them and used two automatic scores: (i) BERTScore (BSc) (Zhang et al., 2020) and (ii) Google BLEU (GLEU) (Wu et al., 2016) to assess the semantic and lexical similarity of the translations and the original text — we used cut-offs of $\geqslant$0.7 for BERTScore and $\geqslant$0.3 for GLEU. We also used a regular expression to check for the presence of 10 or more consecutively repeated words which is a typical error in MT, indicating issues with the quality of the translation.

- **Verifying MT on questions** For questions, we also use the same BSc and GLEU cut-offs above ($\geqslant$0.7 and $\geqslant$0.3) on the backtranslations of the questions to filter out poor quality translations.

Aligning text answer to graph entity

The process here involves mapping g to Portuguese/Russian using Wikidata multilingual labels for entity names. If a multilingual label for the language cannot be found, the name is translated from English using Google Translate API; here, we found that a commercial-grade MT system can be better suited to translate entity names. These multilingual versions of g are only used to aid the mapping of a_t^{PtBr}/a_t^{Ru} to a graph entity and are set aside thereafter (i.e. we do not use them in training)

C.2 Training and evaluation details

C.2.1 Negative sampling for QA

For the textual QA and KG QA tasks, negative examples were created so as to allow the model to recognise questions that are unanswerable given the context. These samples are created using a random (i.e. 50-50) assignment to one of these two strategies:

- **Strategy 1: simple negatives** for a given (X, q, a_X) instance are created by picking another $(X_{other}, q', a_{X'})$ instance in the QA training set where X and X_{other} do not share any common entities/values between them. If X is a text, we use the graph that it is paired with in WEBNLG to check for common entities. A new instance $(X, q',$ "unanswerable") is then created in the training data.

▪ **Strategy 2: hard negatives** are created in the following manner: a mapping M is first created ahead of time where every entity e that can be found in the training data is associated with the set of all the (X, q, a_X) instances where e is mentioned in X ($e \in X$). If X is a text, we use the graph it is paired with in WebNLG when creating M.

For a given (X, q, a_X) instance, another instance, i.e. $(X_{other}, q', a_{X_{other}})$ is randomly chosen using M and a random $e \in X$. If a single token from the answer $a_{X_{other}}$ overlaps with (i) any of the tokens for any e (an entity or value) found in X or (ii) any of the tokens of X, then this $(X_{other}, q', a_{X_{other}})$ candidate is rejected. Otherwise, a new instance $(X, q',$ "unanswerable") is created in the training data using the instance and the process for (X, q, a_X) terminates. When a candidate instance is rejected, a new $(X_{other'}, q', a_{X_{other'}})$ is drawn from M using another $e \in X$. After 10 tries or when all $e \in X$ has been exhausted, a simple negative is created instead.

▪ **Baseline: negative QA instances** For training the baseline's textual QA and KG QA models, we create unanswerable instances in a 2:1 ratio by randomly replacing the context for a question with another in the data.

C.2.2 Upsampling

We carry out upsampling on two levels when preparing Q-WebNLG^{PtBr} and Q-WebNLG^{Ru} (the training data for our language-specific multimodal multi-task QG-QA models).

▪ **Between modalities for QG subtasks and between modalities for the same task** For QG, this is done between complex textual and complex KG QG, simple textual and simple KG QG to ensure that the QG subtasks are balanced. It is also done between modalities for the QA tasks (e.g. textual QA and KG QA) to ensure that the tasks are balanced between modalities. The negative samples for the QA tasks are also upsampled in the same way.

For instance, m_1 and m_2 are the sets of samples for modality 1 and modality 2 respectively, and suppose $\mid m_1 \mid > \mid m_2 \mid$.

If $\mid m_2 \mid / \mid m_1 \mid \geqslant 0.5$, we randomly sample $\mid m_1 \mid - \mid m_2 \mid$ from m_2 to balance them.

If however $\mid m_2 \mid / \mid m_1 \mid < 0.5$, we upsample m_2 up to at most $1/3 \cdot \mid m_1 \mid$. This is to ensure that we do not overrepresent m_2 in the data and overfit on it during training.

▪ **Globally for certain tasks** This was done for simple QG as a whole (i.e. after simple textual and simple KG QG have been consolidated as one). This is because there are significantly fewer instantiated samples of this task in the data than in the rest. The number of samples for these tasks was tripled.

C.2.3 Evaluation settings

▪ **BERTScore** We use the same settings, except the following for a clearer analysis: (i) lowercasing, (ii) "unanswerable" strings were set to an empty string to avoid non-zero BSc for these and "over-counting" them, and (iii) rescaling BScs against a baseline (computed following the official method[67]) for a wider spread.

[67] https://github.com/Tiiiger/bert_score/blob/master/journal/rescale_baseline.md

▪ **Self-consistency filter** Since the models are trained on machine-translated synthetic data, some generated questions may be ill-formed and pose an impact on QA. We therefore filter from both our model and the baseline, the questions: (i) which cannot be answered from their source context; or (ii) whose generated answer $\hat{a}_X$ has a BSc < 0.7 when compared against the reference a_X. We focus our analysis on QA Consistency (Section 5.6), the Data-QuestEval (Section 5.7.4) and mGEN (Section 5.7.5) evaluations on the results after filtering as this is the upper bound of the approaches' performance. For congruence with our QG Coverage analysis, if the filtering will leave a given approach A — and therefore B as well — with no QA pairs (i.e. no coverage), we keep one QA pair for A.

▪ **Settings for mGEN experiments** Since mGEN was trained with language tokens to produce answers in different languages for a question in language L, we leverage these tokens to replicate the same language setting for the inputs and outputs as our experimental set-up: (i) when the answering modality is a text, the input (both question and context) are in Portuguese/Russianand the output answer is in the same language. (ii) When the answering modality is a graph, we use the linearisation scheme from (Oguz et al., 2022) to utilise mGEN for KG QA. The question in the input is in Portuguese/Russian and the context is in English; the output answer is in English.

C.3 Detailed results

C.3.1 QA consistency: Token F1 and Exact Match

Portuguese

	Internal		**X-Appr**	
QG	Baseline	QTT	Baseline	QTT
QA	Baseline	QTT	QTT	Baseline
	Same-mod (GT)			
T → T	$89.9_{(\pm 0.02)}$	$94.9_{(\pm 0.25)}$	$77.6^{(-12.3)}_{(\pm 0.09)}$	$67.9^{(-27.0)}_{(\pm 0.26)}$
G → G	$80.5_{(\pm (0.04)}$	$90.4_{(\pm 0.31)}$	$56.6^{(-23.9)}_{(\pm 0.11)}$	$52.2^{(-38.2)}_{(\pm 0.26)}$

Russian

	Internal		**X-Appr**	
QG	Baseline	QTT	Baseline	QTT
QA	Baseline	QTT	QTT	Baseline
	Same-mod (GT)			
T → T	$86.7_{(\pm 0.10)}$	$84.7_{(\pm 0.14)}$	$53.1^{(-33.6)}_{(\pm 0.06)}$	$42.5^{(-42.2)}_{(\pm 0.11)}$
G → G	$72.6_{(\pm 0.05)}$	$95.2_{(\pm 0.27)}$	$61.2^{(-11.4)}_{(\pm 0.15)}$	$48.9^{(-46.3)}_{(\pm 0.58)}$

TAB. C.1: **Consistency Results**. Average of **Token F1** between answers. In the first brackets () are the standard deviations across five random runs; for the right column, in the second brackets() are the differences between X-Appr and Internal.

Portuguese

	Internal		X-Appr	
QG	Baseline	QTT	Baseline	QTT
QA	Baseline	QTT	QTT	Baseline
	Same-mod (GT)			
T → T	$72.5_{(\pm 0.08)}$	$87.5_{(\pm 0.61)}$	$58.1^{(-14.4)}_{(\pm 0.16)}$	$43.2^{(-44.3)}_{(\pm 0.21)}$
G → G	$73.2_{(\pm 0.10)}$	$87.2_{(\pm 0.36)}$	$48.9^{(-24.3)}_{(\pm 0.21)}$	$41.9^{(-45.3)}_{(\pm 0.26)}$

Russian

	Internal		X-Appr	
QG	Baseline	QTT	Baseline	QTT
QA	Baseline	QTT	QTT	Baseline
	Same-mod (GT)			
T → T	$58.3_{(\pm 0.23)}$	$66.0_{(\pm 0.33)}$	$33.8^{(-24.5)}_{(\pm 0.12)}$	$36.6^{(-29.4)}_{(\pm 0.12)}$
G → G	$60.9_{(\pm 0.16)}$	$92.9_{(\pm 0.52)}$	$39.3^{(-21.6)}_{(\pm 0.14)}$	$33.9^{(-59.0)}_{(\pm 0.74)}$

TAB. C.2: **Consistency Results.** Average of **Exact Match** between answers. In the first brackets () are the standard deviations across five random runs; for the right column, in the second brackets() are the differences between X-Appr and Internal

C.3.2 Finer-grained QA consistency tests

Portuguese

	Num Facts			
	1	2	3	4
	Same-mod (GT)			
T → T	93.6	96.0	95.5	95.4
G → G	95.7	98.1	97.8	98.5
	Cross-mod (GT)			
T → G	65.6	62.5	58.2	54.6
G → T	67.1	66.0	61.0	57.0
	Cross-mod (Gen Ans)			
T → G	68.6	63.7	59.3	55.7
G → T	69.2	67.0	61.1	57.4

TAB. C.3: **Fine-grained analysis of QTT's QA performance (BSc) for Portuguese. Num Facts** denote the number of facts (nf) the set of QA-pairs relate to (i.e. 1 denotes an SQ of 1 fact, 2 denotes a CQ of 2 facts etc...). The nf sets are mutually exclusive.

Russian				
	Num Facts			
	1	2	3	4
	Same-mod (GT)			
T → T	90.2	90.6	90.6	86.0
G → G	93.8	97.6	98.0	96.8
	Cross-mod (GT)			
T → G	29.8	30.1	30.8	29.5
G → T	29.6	29.3	30.0	31.0
	Cross-mod (Gen Ans)			
T → G	30.8	30.3	30.9	31.6
G → T	32.8	29.5	30.7	30.9

TAB. C.4: **Fine-grained analysis of QTT's QA performance (BSc) for Russian. Num Facts** denote the number of facts (nf) the set of QA-pairs relate to (i.e. 1 denotes an SQ of 1 fact, 2 denotes a CQ of 2 facts etc...). The nf sets are mutually exclusive.

C.3.3 Multilingual Retrieval-based QA

Portuguese		
	External Retrieval-QA	
QG	Baseline	Ours
QA	mGEN	mGEN
	Self (GT)	
T → T	$73.1_{(\pm 0.04)}$	$75.8_{(\pm 0.60)}$
G → G	$56.3_{(\pm 0.10)}$	$70.0_{(\pm 0.39)}$

Russian		
	External Retrieval-QA	
QG	Baseline	Ours
QA	mGEN	mGEN
	Self (GT)	
T → T	$61.7_{(\pm 0.19)}$	$60.6_{(\pm 0.17)}$
G → G	$59.4_{(\pm 0.17)}$	$67.2_{(\pm 0.82)}$

TAB. C.5: **QA consistency (Token F1)** comparing the QG of Baseline and Ours, using mGEN for QA.

Portuguese

	External Retrieval-QA	
QG	Baseline	Ours
QA	mGEN	mGEN
	Self (GT)	
T → T	$52.4_{(\pm 0.13)}$	$63.5_{(\pm 0.79)}$
G → G	$45.2_{(\pm 0.15)}$	$59.3_{(\pm 0.74)}$

Russian

	External Retrieval-QA	
QG	Baseline	Ours
QA	mGEN	mGEN
	Self (GT)	
T → T	$36.4_{(\pm 0.13)}$	$42.2_{(\pm 0.14)}$
G → G	$45.8_{(\pm 0.20)}$	$58.8_{(\pm 1.08)}$

Tab. C.6: **QA consistency (Exact Match)** comparing the QG of Baseline and Ours, using mGEN for QA.

Bibliography

A. Agarwal, N. Sachdeva, R. K. Yadav, V. Udandarao, V. Mittal, A. Gupta, and A. Mathur. 2019. Eduqa: Educational domain question answering system using conceptual network mapping. In ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 8137–8141.

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3554–3565, Online. Association for Computational Linguistics.

Priyanka Agrawal, Chris Alberti, Fantine Huot, Joshua Maynez, Ji Ma, Sebastian Ruder, Kuzman Ganchev, Dipanjan Das, and Mirella Lapata. 2023. Qameleon: Multilingual qa with only 5 examples.

Kisuh Ahn, Beatrice Alex, Johan Bos, Tiphaine Dalmas, Jochen L. Leidner, and Matthew B. Smillie. 2004. Cross-lingual question answering using off-the-shelf machine translation. In Proceedings of the 5th Conference on Cross-Language Evaluation Forum: Multilingual Information Access for Text, Speech and Images, CLEF'04, page 446–457, Berlin, Heidelberg. Springer-Verlag.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Felipe Almeida Costa, Thiago Castro Ferreira, Adriana Pagano, and Wagner Meira. 2020. Building the first English-Brazilian Portuguese corpus for automatic post-editing. In Proceedings of the 28th International Conference on Computational Linguistics, pages 6063–6069, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 2425–2433.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4623–4637, Online. Association for Computational Linguistics.

Akari Asai, Shayne Longpre, Jungo Kasai, Chia-Hsuan Lee, Rui Zhang, Junjie Hu, Ikuya Yamada, Jonathan H. Clark, and Eunsol Choi. 2022. MIA 2022 shared task: Evaluating cross-lingual open-retrieval question answering for 16 diverse languages. In Proceedings of the Workshop on Multilingual Information Access (MIA), pages 108–120, Seattle, USA. Association for Computational Linguistics.

Akari Asai, Xinyan Yu, Jungo Kasai, and Hanna Hajishirzi. 2021. One question answering model for many languages with cross-lingual dense passage retrieval. In Advances in Neural Information Processing Systems, volume 34, pages 7547–7560. Curran Associates, Inc.

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.

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

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, Sofia, Bulgaria. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In Proceedings of the 20th International Joint Conference on Artifical Intelligence, IJCAI'07, page 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Sheng Bi, Xiya Cheng, Yuan-Fang Li, Yongzhen Wang, and Guilin Qi. 2020. Knowledge-enriched, type-constrained and grammar-guided question generation over knowledge bases. In Proceedings of the 28th International Conference on Computational Linguistics, pages 2776–2786, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013a. Translating embeddings for modeling multi-relational data. In Advances in Neural Information Processing Systems, volume 26. Curran Associates, Inc.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013b. Translating embeddings for modeling multi-relational data. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, page 2787–2795, Red Hook, NY, USA. Curran Associates Inc.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 2206–2240. PMLR.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92, page 144–152, New York, NY, USA. Association for Computing Machinery.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+), pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Thiago Castro Ferreira, Diego Moussallem, Emiel Krahmer, and Sander Wubben. 2018. Enriching the WebNLG corpus. In Proceedings of the 11th International Conference on Natural Language Generation, pages 171–176, Tilburg University, The Netherlands. Association for Computational Linguistics.

Ping Chen, Fei Wu, Tong Wang, and Wei Ding. 2018. A semantic qa-based approach for text summarization evaluation. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1).

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2020. Reinforcement learning based graph-to-sequence model for natural question generation. In International Conference on Learning Representations.

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2023. Toward subgraph-guided knowledge graph question generation with graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, pages 1–12.

Yi Cheng, Siyao Li, Bang Liu, Ruihui Zhao, Sujian Li, Chenghua Lin, and Yefeng Zheng. 2021. Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 5968–5978, Online. Association for Computational Linguistics.

Alexandr Chernov, Volha Petukhova, and Dietrich Klakow. 2015. Linguistically motivated question classification. In Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015), pages 51–59, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. 2019. Look before you Hop: Conversational Question Answering over Knowledge Graphs Using Judicious Context Expansion. In Proceedings of the ACM International Conference on Information and Knowledge Management, pages 729–738. Association for Computing Machinery.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 7282–7296, Online. Association for Computational Linguistics.

Liam Cripwell, Anya Belz, Claire Gardent, Albert Gatt, Claudia Borg, Marthese Borg, John Judge, Michela Lorandi, Anna Nikiforovskaya, and William Soto Martinez. 2023. The 2023 WebNLG shared task on low resource languages. overview and evaluation results (WebNLG 2023). In Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023), pages 55–66, Prague, Czech Republic. Association for Computational Linguistics.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. Question answering on knowledge bases and text using universal schema and memory networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 358–365, Vancouver, Canada. Association for Computational Linguistics.

Zhenyun Deng, Yonghua Zhu, Yang Chen, Michael Witbrock, and Patricia Riddle. 2022. Interpretable amr-based question decomposition for multi-hop question answering. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, pages 4093–4099. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In Proceedings of the Ninth Workshop on Statistical Machine

Translation, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kaustubh D. Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Srivastava, Samson Tan, Tongshuang Wu, Jascha Sohl-Dickstein, Jinho D. Choi, Eduard Hovy, Ondrej Dusek, Sebastian Ruder, Sajant Anand, Nagender Aneja, Rabin Banjade, Lisa Barthe, Hanna Behnke, Ian Berlot-Attwell, Connor Boyle, Caroline Brun, Marco Antonio Sobrevilla Cabezudo, Samuel Cahyawijaya, Emile Chapuis, Wanxiang Che, Mukund Choudhary, Christian Clauss, Pierre Colombo, Filip Cornell, Gautier Dagan, Mayukh Das, Tanay Dixit, Thomas Dopierre, Paul-Alexis Dray, Suchitra Dubey, Tatiana Ekeinhor, Marco Di Giovanni, Rishabh Gupta, Rishabh Gupta, Louanes Hamla, Sang Han, Fabrice Harel-Canada, Antoine Honore, Ishan Jindal, Przemyslaw K. Joniak, Denis Kleyko, Venelin Kovatchev, Kalpesh Krishna, Ashutosh Kumar, Stefan Langer, Seungjae Ryan Lee, Corey James Levinson, Hualou Liang, Kaizhao Liang, Zhexiong Liu, Andrey Lukyanenko, Vukosi Marivate, Gerard de Melo, Simon Meoni, Maxime Meyer, Afnan Mir, Nafise Sadat Moosavi, Niklas Muennighoff, Timothy Sum Hon Mun, Kenton Murray, Marcin Namysl, Maria Obedkova, Priti Oli, Nivranshu Pasricha, Jan Pfister, Richard Plant, Vinay Prabhu, Vasile Pais, Libo Qin, Shahab Raji, Pawan Kumar Rajpoot, Vikas Raunak, Roy Rinberg, Nicolas Roberts, Juan Diego Rodriguez, Claude Roux, Vasconcellos P. H. S., Ananya B. Sai, Robin M. Schmidt, Thomas Scialom, Tshephisho Sefara, Saqib N. Shamsi, Xudong Shen, Haoyue Shi, Yiwen Shi, Anna Shvets, Nick Siegel, Damien Sileo, Jamie Simon, Chandan Singh, Roman Sitelew, Priyank Soni, Taylor Sorensen, William Soto, Aman Srivastava, KV Aditya Srivatsa, Tony Sun, Mukund Varma T, A Tabassum, Fiona Anting Tan, Ryan Teehan, Mo Tiwari, Marie Tolkiehn, Athena Wang, Zijian Wang, Gloria Wang, Zijie J. Wang, Fuxuan Wei, Bryan Wilie, Genta Indra Winata, Xinyi Wu, Witold Wydmański, Tianbao Xie, Usama Yaseen, M. Yee, Jing Zhang, and Yue Zhang. 2021. Nl-augmenter: A framework for task-sensitive natural language augmentation.

Chenhe Dong, Ying Shen, Shiyang Lin, Zhenzhou Lin, and Yang Deng. 2024. A unified framework for contextual and factoid question generation. IEEE Transactions on Knowledge and Data Engineering, 36(1):21–34.

Rui Dong, David Smith, Shiran Dudy, and Steven Bedrick. 2019. Noisy neural language modeling for typing prediction in BCI communication. In Proceedings of the Eighth Workshop on Speech and Language Processing for Assistive Technologies, pages 44–51, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Vancouver, Canada. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, An-

thony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David

Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models.

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In The Semantic Web – ISWC 2019, pages 69–78. Springer International Publishing.

Pavel Efimov, Andrey Chertok, Leonid Boytsov, and Pavel Braslavski. 2020. Sberquad – rus-

sian reading comprehension dataset: Description and analysis. In Experimental IR Meets Multilinguality, Multimodality, and Interaction, pages 3–15. Springer International Publishing.

Hady Elsahar, Christophe Gravier, and Frederique Laforest. 2018. Zero-shot question generation from knowledge graphs for unseen predicates and entity types. 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 218–228, New Orleans, Louisiana. Association for Computational Linguistics.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. 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 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, page 1156–1165, New York, NY, USA. Association for Computing Machinery.

Angela Fan and Claire Gardent. 2020. Multilingual AMR-to-text generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2889–2901, Online. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Zichu Fei, Qi Zhang, Tao Gui, Di Liang, Sirui Wang, Wei Wu, and Xuanjing Huang. 2022. CQG: A simple and effective controlled generation framework for multi-hop question generation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6896–6906, Dublin, Ireland. Association for Computational Linguistics.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building watson: An overview of the deepqa project. AI Magazine, 31(3):59–79.

Yifan Gao, Lidong Bing, Wang Chen, Michael Lyu, and Irwin King. 2019. Difficulty controllable generation of reading comprehension questions. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pages 4968–4974. International Joint Conferences on Artificial Intelligence Organization.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017a. Creating training corpora for nlg micro-planners. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 179–188. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017b. 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.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021), pages 96–120, Online. Association for Computational Linguistics.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, page 1263–1272. JMLR.org.

Sujatha Das Gollapalli and See-Kiong Ng. 2022. QSTS: A question-sensitive text similarity measure for question generation. In Proceedings of the 29th International Conference on Computational Linguistics, pages 3835–3846, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Huanli Gong, Liangming Pan, and Hengchang Hu. 2022. KHANQ: A dataset for generating deep questions in education. In Proceedings of the 29th International Conference on Computational Linguistics, pages 5925–5938, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Arthur C. Graesser and Natalie K. Person. 1994. Question asking during tutoring. American Educational Research Journal, 31(1):104–137.

Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: an automatic question-answerer. In Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference, IRE-AIEE-ACM '61 (Western), page 219–224, New York, NY, USA. Association for Computing Machinery.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In Proceedings of the 54th Annual Meeting of the Association for

Computational Linguistics (Volume 1: Long Papers), pages 140–149, Berlin, Germany. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 3929–3938. PMLR.

Michael A.K. Halliday and Ruqaiya Hasan. 1976. Cohesion in English. Longman, London.

Kelvin Han, Thiago Castro Ferreira, and Claire Gardent. 2022. Generating questions from Wikidata triples. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 277–290, Marseille, France. European Language Resources Association.

Namgi Han, Goran Topic, Hiroshi Noji, Hiroya Takamura, and Yusuke Miyao. 2020. An empirical analysis of existing systems and datasets toward general simple question answering. In Proceedings of the 28th International Conference on Computational Linguistics, pages 5321–5334, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An open toolkit for knowledge embedding. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 139–144, Brussels, Belgium. Association for Computational Linguistics.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of BERTScore. In Proceedings of the Sixth Conference on Machine Translation, pages 507–517, Online. Association for Computational Linguistics.

Sanda M Harabagiu, Dan I Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan C Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. 2000. Falcon: Boosting knowledge for answer engines. In TREC, volume 9, pages 479–488.

Michael Heilman and Noah A. Smith. 2010. Good question! statistical ranking for question generation. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 609–617, Los Angeles, California. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural Comput., 9(8):1735–1780.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge graphs. ACM Comput. Surv., 54(4).

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In International Conference on Learning Representations.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In Proceedings of the 2021 Conference on Empirical Methods in

Natural Language Processing, pages 7856–7870, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In Proceedings of the 13th International Conference on Natural Language Generation, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Sen Hu, Lei Zou, and Zhanxing Zhu. 2019. How question generation can help question answering over knowledge base. In Natural Language Processing and Chinese Computing, pages 80–92. Springer International Publishing.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM '19, page 105–113, New York, NY, USA. Association for Computing Machinery.

Fantine Huot, Joshua Maynez, Shashi Narayan, Reinald Kim Amplayo, Kuzman Ganchev, Annie Priyadarshini Louis, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Text-blueprint: An interactive platform for plan-based conditional generation. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 105–116, Dubrovnik, Croatia. Association for Computational Linguistics.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. MathPrompter: Mathematical reasoning using large language models. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track), pages 37–42, Toronto, Canada. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 874–880, Online. Association for Computational Linguistics.

Khushnur Jahangir, Philippe Muller, and Chloé Braud. 2024. Complex question generation using discourse-based data augmentation. In Proceedings of the 5th Workshop on Computational Approaches to Discourse (CODI 2024), pages 105–119, St. Julians, Malta. Association for Computational Linguistics.

Hongyan Jing, Regina Barzilay, Kathleen McKeown, and Michael Elhadad. 1998. Summarization evaluation methods: Experiments and analysis. Papers from the 1998 AAAI Spring Symposium.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2024. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition. Online manuscript released August 20, 2024.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In Proceedings of the 13th International Conference on Natural Language Generation, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Hans Kamp. 2004. A Theory of Truth and Semantic Representation. In Semantics: A Reader. Oxford University Press.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6769–6781, Online. Association for Computational Linguistics.

Rohit J. Kate, Yuk Wah Wong, and Raymond J. Mooney. 2005. Learning to transform natural to formal languages. In Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3, AAAI'05, page 1062–1068. AAAI Press.

Daniel Khashabi, Yeganeh Kordi, and Hannaneh Hajishirzi. 2022. Unifiedqa-v2: Stronger generalization via broader cross-format training. arXiv preprint arXiv:2202.12359.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations.

Oleksandr Kolomiyets and Marie-Francine Moens. 2011. A survey on question answering technology from an information retrieval perspective. Information Sciences, 181(24):5412–5434.

Kalpesh Krishna and Mohit Iyyer. 2019. Generating question-answer hierarchies. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2321–2334, Florence, Italy. Association for Computational Linguistics.

Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. 2019. Information maximizing visual question generation. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 2008–2018.

Vishwajeet Kumar, Yuncheng Hua, Ganesh Ramakrishnan, Guilin Qi, Lianli Gao, and Yuan-Fang Li. 2019. Difficulty-controllable multi-hop question generation from knowledge graphs. In International Semantic Web Conference, pages 382–398. Springer.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7:452–466.

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 969–974, Online. Association for Computational Linguistics.

Ora Lassila and Ralph R. Swick. 1999. Resource Description Framework (RDF) Model and Syntax Specification.

Carolin Lawrence, Bhushan Kotnis, and Mathias Niepert. 2019. Attending to future tokens for bidirectional sequence 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 1–10, Hong Kong, China. Association for Computational Linguistics.

Gwénolé Lecorvé, Morgan Veyret, Quentin Brabant, and Lina M. Rojas Barahona. 2022. SPARQL-to-text question generation for knowledge-based conversational applications. In Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 131–147, Online only. Association for Computational Linguistics.

Hwanhee Lee, Thomas Scialom, Seunghyun Yoon, Franck Dernoncourt, and Kyomin Jung. 2021. QACE: Asking questions to evaluate an image caption. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4631–4638, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jens Lehmann, Dhananjay Bhandiwad, Preetam Gattogi, and Sahar Vahdati. 2024. Beyond Boundaries: A Human-like Approach for Question Answering over Structured and Unstructured Information Sources. Transactions of the Association for Computational Linguistics, 12:786–802.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web, 6(2):167–195.

Wendy G. Lehnert. 1978. The Process of Question Answering: A Computer Simulation of Cognition. Routledge.

DB Lenat and RV Guha. 1993. Building large knowledge-based systems: Representation and inference in the cyc project. Artificial Intelligence, 61(1):4152.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020b. MLQA: Evaluating cross-lingual extractive question answering. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7315–7330, Online. Association for Computational Linguistics.

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. 2020c. Retrieval-augmented generation for knowledge-intensive nlp tasks. In Advances in Neural Information Processing Systems, volume 33, pages 9459–9474. Curran Associates, Inc.

Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020. Efficient one-pass end-to-end entity linking for questions. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6433–6441, Online. Association for Computational Linguistics.

Fangjun Li, David C. Hogg, and Anthony G. Cohn. 2024. Advancing spatial reasoning in large language models: An in-depth evaluation and enhancement using the stepgame benchmark. Proceedings of the AAAI Conference on Artificial Intelligence, 38(17):18500–18507.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021. Few-shot knowledge graph-to-text generation with pretrained language models. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1558–1568, Online. Association for Computational Linguistics.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2022. A systematic investigation of commonsense knowledge in large language models. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11838–11855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Cao Liu, Kang Liu, Shizhu He, Zaiqing Nie, and Jun Zhao. 2019a. Generating questions for knowledge bases via incorporating diversified contexts and answer-aware loss. 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 2431–2441, Hong Kong, China. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2021. Text generation from discourse representation structures. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 397–415, Online. Association for Computational Linguistics.

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.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: designing data and methods for effective instruction tuning. In Proceedings of the 40th International Conference on Machine Learning, ICML'23. JMLR.org.

Luis Enrico Lopez, Diane Kathryn Cruz, Jan Christian Blaise Cruz, and Charibeth Cheng. 2021. Simplifying paragraph-level question generation via transformer language models.

Bruce Lowerre. 1990. The Harpy speech understanding system, page 576–586. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 783–792, Honolulu, Hawaii. Association for Computational Linguistics.

Chenyang Lyu, Lifeng Shang, Yvette Graham, Jennifer Foster, Xin Jiang, and Qun Liu. 2021. Improving unsupervised question answering via summarization-informed question generation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4134–4148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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, Tilburg University, The Netherlands. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In Proc. Interspeech 2010, pages 1045–1048.

George A. Miller. 1994. WordNet: A lexical database for English. In Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994.

David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, page 509–518, New York, NY, USA. Association for Computing Machinery.

Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing, pages 17–22.

Alireza Mohammadshahi, Thomas Scialom, Majid Yazdani, Pouya Yanki, Angela Fan, James Henderson, and Marzieh Saeidi. 2023. RQUGE: Reference-free metric for evaluating question generation by answering the question. In Findings of the Association for Computational Linguistics: ACL 2023, pages 6845–6867, Toronto, Canada. Association for Computational Linguistics.

Salman Mohammed, Peng Shi, and Jimmy Lin. 2018. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 291–296, New Orleans, Louisiana. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-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 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.

Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question answer classification. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 776–783, Prague, Czech Republic. Association for Computational Linguistics.

Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1802–1813, Berlin, Germany. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Kevin P. Murphy. 2023. Probabilistic Machine Learning: Advanced Topics. MIT Press.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. 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 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Conditional Generation with a Question-Answering Blueprint. Transactions of the Association for Computational Linguistics, 11:974–996.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence, 193:217–250.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a better metric for evaluating question generation systems. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022. UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 1535–1546, Seattle, United States. Association for Computational Linguistics.

Andrew M Olney, Arthur C Graesser, and Natalie K Person. 2012. Question generation from concept maps. Dialogue & Discourse, 3(2):75–99.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski

Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2024. Training language models to follow instructions with human feedback. In Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1463–1475, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Thomas Pellissier Tanon, Marcos Dias de Assunção, Eddy Caron, and Fabian M. Suchanek. 2018. Demoing platypus – a multilingual question answering platform for wikidata. In The Semantic Web: ESWC 2018 Satellite Events: ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers, page 111–116, Berlin, Heidelberg. Springer-Verlag.

Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. 2016. From freebase to wikidata: The great migration. In Proceedings of the 25th International Conference on World Wide Web, WWW '16, page 1419–1428, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014a. GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014b. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543.

Aleksandr Perevalov, Andreas Both, Dennis Diefenbach, and Axel-Cyrille Ngonga Ngomo. 2022. Can machine translation be a reasonable alternative for multilingual question answering systems over knowledge graphs? In Proceedings of the ACM Web Conference 2022, WWW '22, page 977–986, New York, NY, USA. Association for Computing Machinery.

Aleksandr Perevalov, Andreas Both, and Axel-Cyrille Ngonga Ngomo. 2023. Multilingual question answering systems for knowledge graphs-a survey. Semantic Web Journal.

Laura Perez-Beltrachini, Parag Jain, Emilio Monti, and Mirella Lapata. 2023. Semantic parsing for conversational question answering over knowledge graphs. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 2507–2522, Dubrovnik, Croatia. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2523–2544, Online. Association for Computational Linguistics.

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, Hong Kong, China. Association for Computational Linguistics.

Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024. Rephrase, augment, reason: Visual grounding of questions for vision-language models. In The Twelfth International Conference on Learning Representations.

Rashmi Prasad, Bonnie Webber, and Aravind Joshi. 2014. Reflections on the Penn Discourse TreeBank, comparable corpora, and complementary annotation. Computational Linguistics, 40(4):921–950.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5811–5826, Online. Association for Computational Linguistics.

Kechen Qin, Cheng Li, Virgil Pavlu, and Javed Aslam. 2021. Improving query graph generation for complex question answering over knowledge base. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4201–4207, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiazuo Qiu and Deyi Xiong. 2019. Generating highly relevant questions. 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 5983–5987, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67.

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, Austin, Texas. Association for Computational Linguistics.

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.

Clement Rebuffel, Thomas Scialom, Laure Soulier, Benjamin Piwowarski, Sylvain Lamprier, Jacopo Staiano, Geoffrey Scoutheeten, and Patrick Gallinari. 2021. Data-QuestEval: A referenceless metric for data-to-text semantic evaluation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8029–8036, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Raj Reddy. 1977. Speech understanding systems. Summary of results of the five-year research effort at Carnegie-Mellon University. Interim Report Carnegie-Mellon Univ., Pittsburgh, PA. Dept. of Computer Science.

Revanth Gangi Reddy, Xilin Rui, Manling Li, Xudong Lin, Haoyang Wen, Jaemin Cho, Lifu Huang, Mohit Bansal, Avirup Sil, Shih-Fu Chang, et al. 2021. Mumuqa: Multimedia multi-hop news question answering via cross-media knowledge extraction and grounding. arXiv preprint arXiv:2112.10728.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. Transactions of the Association for Computational Linguistics, 7:249–266.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-scale semantic parsing without question-answer pairs. Transactions of the Association for Computational Linguistics, 2:377–392.

Arij Riabi, Thomas Scialom, Rachel Keraron, Benoît Sagot, Djamé Seddah, and Jacopo Staiano. 2021. Synthetic data augmentation for zero-shot cross-lingual question answering. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7016–7030, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI, pages 211–227, Online. Association for Computational Linguistics.

Petar Ristoski and Heiko Paulheim. 2016. Rdf2vec: Rdf graph embeddings for data mining. In The Semantic Web – ISWC 2016, pages 498–514, Cham. Springer International Publishing.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5418–5426, Online. Association for Computational Linguistics.

Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at trec-3. In Overview of the Third Text REtrieval Conference (TREC-3), pages 109–126. Gaithersburg, MD: NIST.

Rishiraj Saha Roy and Avishek Anand. 2022. Question answering for the curated web: Tasks and methods in qa over knowledge bases and text collections. Springer Nature.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. nature, 323(6088):533–536.

Vasile Rus and Graesser C. Arthur. 2009. Workshop report: The question generation shared task and evaluation challenge. Technical report, The University of Memphis, United States.

Vasile Rus, Zhiqiang Cai, and Art Graesser. 2008. Question generation: Example of a multi-year evaluation campaign.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Christian Moldovan. 2010. The first question generation shared task evaluation challenge. In Proceedings of the 6th International Natural Language Generation Conference. Association for Computational Linguistics.

Amrita Saha, Vardaan Pahuja, Mitesh Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex Sequential Question Answering: Towards Learning to Converse Over Linked Question Answer Pairs with a Knowledge Graph. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1).

Stephanie Schoch, Diyi Yang, and Yangfeng Ji. 2020. "this is a problem, don't you agree?" framing and bias in human evaluation for natural language generation. In Proceedings of the 1st Workshop on Evaluating NLG Evaluation, pages 10–16, Online (Dublin, Ireland). Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Priyanka Sen, Sandeep Mavadia, and Amir Saffari. 2023. Knowledge graph-augmented language models for complex question answering. In Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE), pages 1–8, Toronto, Canada. Association for Computational Linguistics.

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.

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 588–598, Berlin, Germany. Association for Computational Linguistics.

Dominic Seyler, Mohamed Yahya, and Klaus Berberich. 2015. Generating quiz questions from knowledge graphs. In Proceedings of the 24th International Conference on World Wide Web, pages 113–114.

Dominic Seyler, Mohamed Yahya, and Klaus Berberich. 2017. Knowledge questions from knowledge graphs. In Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR '17, page 11–18, New York, NY, USA. Association for Computing Machinery.

Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. 2019. Cycle-consistency for robust visual question answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6649–6658.

Siamak Shakeri, Noah Constant, Mihir Kale, and Linting Xue. 2021. Towards zero-shot multilingual synthetic question and answer generation for cross-lingual reading comprehension. In Proceedings of the 14th International Conference on Natural Language Generation, pages 35–45, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Anastasia Shimorina, Elena Khasanova, and Claire Gardent. 2019. Creating a corpus for Russian data-to-text generation using neural machine translation and post-editing. In Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing, pages 44–49, Florence, Italy. Association for Computational Linguistics.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. Transactions of the Association for Computational Linguistics, 11:1–17.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018. Leveraging context information for natural question generation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 569–574, New Orleans, Louisiana. Association for Computational Linguistics.

Linfeng Song and Lin Zhao. 2017. Question generation from a knowledge base with web exploration.

Yuanfeng Song, Di Jiang, Weiwei Zhao, Qian Xu, Raymond Chi-Wing Wong, and Qiang Yang. 2019. Chameleon: A language model adaptation toolkit for automatic speech recognition of conversational speech. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, pages 37–42, Hong Kong, China. Association for Computational Linguistics.

Karen Sparck Jones. 1988. A statistical interpretation of term specificity and its application in retrieval, page 132–142. Taylor Graham Publishing, GBR.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge.

Mark Steedman. 1987. Combinatory grammars and parasitic gaps. Natural Language Linguistic Theory, 5(3):403–439.

Yao Sun, Anastasiia Tatlubaeva, Zhihan Li, and Chester Palen-Michel. 2024. What are the implications of your question? non-information seeking question-type identification in CNN transcripts. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 17444–17448, Torino, Italia. ELRA and ICCL.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. 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 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. Multimodal{qa}: complex question answering over text, tables and images. In International Conference on Learning Representations.

Jan Trienes, Sebastian Joseph, Jörg Schlötterer, Christin Seifert, Kyle Lo, Wei Xu, Byron Wallace, and Junyi Jessy Li. 2024. InfoLossQA: Characterizing and recovering information loss in text simplification. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4263–4294, Bangkok, Thailand. Association for Computational Linguistics.

Ricardo Usbeck, Ria Hari Gusmita, Axel-Cyrille Ngonga Ngomo, and Muhammad Saleem. 2018. 9th challenge on question answering over linked data (qald-9) (invited paper). In Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, California, United States of America, October 8th - 9th, 2018, volume 2241 of CEUR Workshop Proceedings, page 58–64. CEUR-WS.org.

Ricardo Usbeck, Xi Yan, Aleksandr Perevalov, Longquan Jiang, Julius Schulz, Angelie Kraft, Cedric Möller, Junbo Huang, Jan Reineke, Axel-Cyrille Ngonga Ngomo, et al. 2023. Qald-10—the 10th challenge on question answering over linked data.

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2022. Generative language models for paragraph-level question generation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 670–688, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2023a. An empirical comparison of LM-based question and answer generation methods. In Findings of the Association for Computational Linguistics: ACL 2023, pages 14262–14272, Toronto, Canada. Association for Computational Linguistics.

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2023b. A practical toolkit for multilingual question and answer generation. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 86–94, Toronto, Canada. Association for Computational Linguistics.

Chris van der Lee, Thiago Castro Ferreira, Chris Emmery, Travis J. Wiltshire, and Emiel Krahmer. 2023. Neural data-to-text generation based on small datasets: Comparing the added value of two semi-supervised learning approaches on top of a large language model. Computational Linguistics, pages 555–611.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In Proceedings of the 12th International Conference on Natural Language Generation, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.

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, volume 30. Curran Associates, Inc.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. Scipy 1.0: fundamental algorithms for scientific computing in python. Nature methods, 17(3):261–272.

Ellen Voorhees. 2000. The trec-8 question answering track report.

Ellen Voorhees and D Tice. 2000. The trec-8 question answering track evaluation.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. Commun. ACM, 57(10):78–85.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5008–5020, Online. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In International Conference on Learning Representations.

Shuohang Wang and Jing Jiang. 2017. Machine comprehension using match-LSTM and answer pointer. In International Conference on Learning Representations.

Xiaoqiang Wang, Bang Liu, Siliang Tang, and Lingfei Wu. 2022. QRelScore: Better evaluating generated questions with deeper understanding of context-aware relevance. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 562–581, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2193–2203, Melbourne, Australia. Association for Computational Linguistics.

Zichao Wang and Richard Baraniuk. 2023. MultiQG-TI: Towards question generation from multi-modal sources. In Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), pages 682–691, Toronto, Canada. Association for Computational Linguistics.

William A. Woods. 1977. Lunar rocks in natural English: Explorations in natural language question answering. In Antonio Zampolli, editor, Linguistic Structures Processing, pages 521–569. North Holland, Amsterdam.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6397–6407, Online. Association for Computational Linguistics.

Yating Wu, Ritika Mangla, Greg Durrett, and Junyi Jessy Li. 2023. QUDeval: The evaluation of questions under discussion discourse parsing. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 5344–5363, Singapore. 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, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Ziang Xiao, Susu Zhang, Vivian Lai, and Q. Vera Liao. 2023. Evaluating evaluation metrics: A framework for analyzing NLG evaluation metrics using measurement theory. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 10967–10982, Singapore. Association for Computational Linguistics.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. In International Conference on Learning Representations.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 483–498, Online. Association for Computational Linguistics.

Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In Proceedings of the International Conference on Learning Representations (ICLR) 2015.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Probabilistic databases of universal schema. In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX), pages 116–121, Montréal, Canada. Association for Computational Linguistics.

Xuchen Yao, Jonathan Berant, and Benjamin Van Durme. 2014. Freebase QA: Information extraction or semantic parsing? In Proceedings of the ACL 2014 Workshop on Semantic Parsing, pages 82–86, Baltimore, MD. Association for Computational Linguistics.

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with Freebase. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 956–966, Baltimore, Maryland. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 535–546, Online. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. 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 1321–1331, Beijing, China. Association for Computational Linguistics.

Jing Yu, Zihao Zhu, Yujing Wang, Weifeng Zhang, Yue Hu, and Jianlong Tan. 2020. Cross-modal knowledge reasoning for knowledge-based visual question answering. Pattern Recognition, 108:107563.

Chen Zhang, Yuxuan Lai, Yansong Feng, Xingyu Shen, Haowei Du, and Dongyan Zhao. 2023a. Cross-lingual question answering over knowledge base as reading comprehension. In Findings of the Association for Computational Linguistics: EACL 2023, pages 2439–2452, Dubrovnik, Croatia. Association for Computational Linguistics.

Kun Zhang, Oana Balalau, and Ioana Manolescu. 2023b. FactSpotter: Evaluating the factual faithfulness of graph-to-text generation. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 10025–10042, Singapore. Association for Computational Linguistics.

Kun Zhang, Yunqi Qiu, Yuanzhuo Wang, Long Bai, Wei Li, Xuhui Jiang, Huawei Shen, and Xueqi Cheng. 2022. Meta-CQG: A meta-learning framework for complex question generation over knowledge bases. In Proceedings of the 29th International Conference on Computational Linguistics, pages 6105–6114, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and yang: Balancing and answering binary visual questions. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5014–5022.

Shijie Zhang, Lizhen Qu, Shaodi You, Zhenglu Yang, and Jiawan Zhang. 2017. Automatic generation of grounded visual questions. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pages 4235–4243.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In International Conference on Learning Representations.

Jie Zhao, Xiang Deng, and Huan Sun. 2019. Easy-to-hard: Leveraging simple questions for complex question generation.

Liang Zhou, Chin-Yew Lin, and Eduard Hovy. 2006. Re-evaluating machine translation results with paraphrase support. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pages 77–84, Sydney, Australia. Association for Computational Linguistics.

Elizaveta Zimina, Jyrki Nummenmaa, Kalervo Jarvelin, Jaakko Peltonen, and Kostas Stefanidis. 2018. Mug-qa: Multilingual grammatical question answering for rdf data. In 2018 IEEE International Conference on Progress in Informatics and Computing (PIC), pages 57–61.