UNIVERSITÉ DE LORRAINE | BIBLIOTHÈQUES UNIVERSITAIRES

AVERTISSEMENT
DROIT D'AUTEUR

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. Toute contrefaçon, plagiat ou reproduction illicite encourt une poursuite pénale.

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

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

UNIVERSITÉ DE LORRAINE
34 COURS LÉOPOLD — CS 25233
54052 NANCY CEDEX
TÉL : 03 54 50 54 00
CONTACT@UNIV-LORRAINE.FR
WWW.UNIV-LORRAINE.FR

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Multilingual KG-to-text Generation: From Data Creation To Human Evaluation

THÈSE

présentée et soutenue publiquement le 17 avril 2026

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Anna NIKIFOROVSKAJA

Composition du jury

Président :	Sina Zarriess	PR, University of Bielefeld, Allemagne
Rapporteurs :	Pierre Zweigenbaum	DR CNRS LISN, Paris Saclay
	Sina Zarriess	PR, University of Bielefeld, Allemagne
Examinateurs :	Saad Mahamood	AI Lead and NLG Expert at Shopware
	Dimitra Gkatzia	Associate Professor, Edinburgh Napier University
Directrice de thèse :	Claire Gardent	Directrice de recherche, CNRS, LORIA, France

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

Mis en page avec la classe thesul.

Remerciements

This thesis is the result of several years of work, persistence, and growth. There were ups and downs along the way, and I would not have made it through without the support of the people around me. I would like to sincerely thank everyone who accompanied me throughout this PhD journey.

First and foremost, I am deeply grateful to my supervisor, Claire Gardent. Her guidance, support, and insight were essential throughout these years, and I am especially thankful that we worked so well together. I learned a great deal from her, both scientifically and professionally, and I greatly appreciated her trust and encouragement throughout my PhD.

I would also like to thank the members of my jury, Pierre Zweigenbaum, Sina Zarrieß, Saad Mahamood, and Dimitra Gkatzia, for agreeing to be part of the jury and for the thoughtful and engaging discussions we had during the defense.

My sincere thanks also go to my teammates at LORIA. All the conversations, jokes, coffee breaks, and shared moments made these years much more enjoyable and will remain precious memories. In particular, I am grateful to William and Yifei, with whom I had the pleasure of collaborating on several projects, as well as Kelvin, who supported me greatly during the beginning of this journey.

My heartfelt thanks go to my friends — Anna M., who was also pursuing a PhD at the same time and with whom I could share both the challenges and joys of this experience, as well as Marika P., Anna S., Fabiola S. and Alexander V. — who were always there for me, even from afar.

I am equally grateful to my family for their unconditional support and encouragement throughout these years. Even though seeing each other was often difficult, I always knew I could count on their love, patience, and support.

Enfin, un immense merci à la Chorale U, que j'ai eu la chance de trouver à Nancy. L'ambiance chaleureuse de la chorale m'a beaucoup aidée dans les moments difficiles et après les longues journées de travail. Pierre et Madeline, directeurs de cette chorale, ont créé un environnement accueillant et bienveillant qui a été très précieux pour moi pendant ces années. Je voudrais remercier tout particulièrement Camille W., Aude M., Marie C. et Léa P. pour leur présence, leur gentillesse et tous les bons moments partagés.

"If we wait until we're ready, we'll be waiting for the rest of our lives."
– Lemony Snicket, "A Series of Unfortunate Events"

Résumé

Avec les vastes quantités de connaissances accessibles sous forme de graphes de connaissances (knowledge graphs, KG), la génération de texte à partir de graphes de connaissances (KG-to-text) est devenue un élément important de nombreuses applications utilisant ces connaissances pour améliorer les systèmes de dialogue, la traduction et la réponse aux questions. Cependant, alors que de nombreuses recherches ont été menées pour l'anglais, la génération KG-to-text pour d'autres langues reste peu explorée.

Dans cette thèse, nous abordons cet enjeu en nous concentrant d'abord sur le russe, en analysant les capacités de généralisation aux données hors distribution des méthodes existantes de génération KG-to-text en russe. En introduisant de nouveaux benchmarks pour l'anglais et le russe qui intègrent des ratios variables d'entités et de propriétés non présentes dans les données d'entrainement tirés de DBpedia et de Wikidata, la recherche identifie d'abord une baisse significative des performances des modèles standard lors du traitement de données hors distribution. Une analyse qualitative met en évidence des schémas d'erreurs spécifiques induits par ces entrées inconnues, fournissant ainsi une base pour évaluer des stratégies de génération plus robustes.

Nous utilisons ensuite cette analyse pour proposer plusieurs moyens d'améliorer la génération KG-to-text en russe en incorporant des informations supplémentaires afin de cibler les erreurs détectées. Nous évaluons l'affinage, le prompting de modèle linguistique à grande échelle (Large Language Models, LLM) et la génération augmentée par recherche d'information (Retrieval Augmented Generation, RAG), en explorant d'autres moyens d'intégrer des connaissances externes, tels que la mise à disposition de traductions d'entités possibles ou de verbalisations de graphes similaires. Les résultats indiquent que les modèles affinés ont une capacité de généralisation limitée sur des données inconnues. Si la RAG permet d'obtenir une plus grande fluidité, elle est moins fidèle à la KG d'entrée que le LLM-prompting. En fin de compte, le LLM-prompting et la RAG s'avèrent tous deux plus performants que l'affinement traditionnel sur tous les ensembles de tests inconnus, ce qui suggère que les stratégies de prompting et de RAG sont plus efficaces pour gérer les changements de domaine.

Nous étendons ensuite notre analyse et nos recherches à d'autres langues, notamment l'anglais, le français, le chinois, le russe et l'espagnol. Après avoir créé des ensembles d'entraînement multilingues « silver » et des données de test « gold », nous comparons trois approches architecturales distinctes : un modèle hybride combinant la génération KG-to-text en anglais avec la traduction automatique, la traduction automatique directement affinée et le few-shot prompting de LLM. Les résultats démontrent que les stratégies de few-shot prompting surpassent systématiquement les approches de traduction hybrides et affinées dans les cinq langues cibles. De plus, nous poussons plus loin l'approche de prompting LLM pour générer des textes dans des langues peu dotées, montrant que le prompting peut combler le fossé pour celles-ci.

Mots-clés: Génération KG-to-text, génération de texte multilingue, évaluation, LLM

Abstract

With vast amounts of knowledge accessible as Knowledge Graphs (KGs), Knowledge Graph to text generation (KG-to-text) has become an important part of many applications using that knowledge to improve dialogue systems, translation and question answering. However, while a lot of research has been done for English, KG-to-text generation for other languages remains under-explored.

In this thesis we address this by first focusing on Russian, analysing generalisation capabilities to out-of-distribution data of existing KG-to-Russian text generation methods. By introducing novel benchmarks for English and Russian that incorporate varying ratios of unseen entities and properties from both DBpedia and Wikidata, the research first identifies a significant performance decay in standard models when processing out-of-distribution data. A qualitative analysis further highlights specific error patterns induced by these unseen inputs, providing a foundation for evaluating more robust generation strategies.

We then use this analysis to propose several ways to improve KG-to-Russian text generation by incorporating additional information to target found errors. We evaluate Fine-Tuning, Large Language Model (LLM) prompting, and Retrieval-Augmented Generation (RAG), exploring alternative ways of integrating external knowledge, such as providing possible entities translations or similar graphs verbalisations. The findings indicate that fine-tuned models demonstrate limited generalisation capacity on unseen data. While RAG achieves higher fluency, it is less faithful to the input KG than LLM-prompting. Ultimately, both LLM-prompting and RAG are found to outperform traditional fine-tuning across all unseen test sets, suggesting that prompting and RAG strategies are more effective for handling domain shifts.

We then extend our analysis and research to other languages, covering English, French, Chinese, Russian, and Spanish. After creating silver multilingual training sets and gold test data, the work compares three distinct architectural approaches: a hybrid model combining KG-to-English text generation with machine translation, direct fine-tuned machine translation, and few-shot LLM prompting. The results demonstrate that optimized few-shot prompting strategies consistently outperform both hybrid and fine-tuned translation approaches across all five target languages. Moreover, we further challenge the LLM prompting approach to generate texts in low-resource languages, showing that prompting can bridge the gap for them.

Keywords: KG-to-text generation, multilingual text generation, evaluation, LLM

Contents

Génération de textes multilingues à partir de graphes de connaissances **xxiii**

1 Questions de Recherche et Contributions xxiv

2 Plan de la thèse xxvi

3 Liste des publications xxvii

Introduction **1**

1 Research Questions 2

2 Thesis Outline 3

3 List of Publications 4

Chapter 1 Background **5**

1.1 Introduction to Natural Language Generation 6

1.1.1 Text-to-text generation 6

1.1.2 Data-to-text generation 7

1.2 Data for KG-to-text generation 10

1.2.1 Automatically aligned datasets 10

1.2.2 Human annotated datasets 10

1.2.3 Unsupervised datasets 12

1.3 Natural Language Generation Methods 12

1.3.1 Pre-neural methods 12

1.3.2 Neural methods 13

1.3.3 Prompting methods 15

1.3.4 RAG 15

1.4 Evaluation 16

1.4.1 Automatic Evaluation 16

1.4.2 Human Evaluation 20

1.4.3 LLM-as-a-judge for Evaluation 22

1.5 Conclusion . . . 23

Chapter 2 Evaluating Knowledge Graph-to-text Generation Models 25

2.1 Introduction . . . 26
2.2 Creating New Benchmarks for English and Russian . . . 27
2.3 Creating Kelm Benchmarks . . . 28
2.4 Creating WebNLG Benchmarks for Russian . . . 32
2.5 Assessing Generalisation . . . 33
2.6 Quantitative Analysis . . . 33
2.6.1 Models . . . 33
2.6.2 Metrics . . . 34
2.6.3 Results . . . 34
2.7 Qualitative Analysis . . . 36
2.7.1 Error Annotation . . . 36
2.7.2 Error Analysis . . . 37
2.8 Conclusion . . . 38

Chapter 3 How do Knowledge Graph-to-Russian Text Generation Models generalise? 41

3.1 Introduction . . . 42
3.2 Data . . . 43
3.2.1 Test Data . . . 43
3.2.2 Fine-Tuning and Few-Shot Data . . . 44
3.3 Fine-tuning on Labeled data. . . . 44
3.4 LLM Prompting . . . 45
3.4.1 Input . . . 46
3.4.2 Generation strategies . . . 47
3.4.3 Few-shot selection strategies . . . 47
3.4.4 Results . . . 50
3.5 Retrieval Augmented Generation for LLM . . . 51
3.5.1 Paragraph Retrieval . . . 51
3.5.2 Chunks . . . 51
3.5.3 Queries . . . 53
3.5.4 Few-shots . . . 54
3.5.5 Results . . . 54

3.6 Comparing the Three Approaches . . . 55
3.6.1 Automatic Evaluation . . . 55
3.6.2 Human Evaluation . . . 56
3.6.3 Comparing the Three Approaches: Case Study . . . 58
3.7 Conclusion . . . 59

Chapter 4 Multilingual Verbalisation of Knowledge Graphs **61**
4.1 Introduction . . . 62
4.2 Data . . . 63
4.3 Models . . . 65
4.3.1 Supervised Approaches . . . 66
4.3.2 LLM-based Generation . . . 66
4.4 Automatic Evaluation . . . 67
4.4.1 Metrics . . . 67
4.4.2 Results . . . 68
4.5 Error Analysis . . . 71
4.5.1 Analysing the FewShot Model . . . 71
4.5.2 Comparing Models . . . 72
4.6 Extension of the Experiments . . . 72
4.6.1 Data . . . 72
4.6.2 Models . . . 74
4.6.3 Results . . . 75
4.7 Conclusion . . . 78

Chapter 5 Conclusion **81**
5.1 Main Contributions . . . 81
5.2 Limitations and Future Research . . . 82
5.3 Final Remarks . . . 83

Appendix A Appendix for Chapter 2 **85**
A.1 Human Annotation Details . . . 85
A.2 Full Automatic Evaluation Results . . . 85
A.3 Error Annotation Examples . . . 85
A.4 Full Qualitative Analysis Results . . . 86

Appendix B Appendix for Chapter 3 **95**
B.1 Example Prompts . . . 95
B.2 Wikidata SparQL Queries . . . 100
B.3 Automatic Evaluation Results . . . 101
B.4 Hyperparameters . . . 105

Appendix C Appendix for Chapter 4 **107**
C.1 Comparing Different Prompting Strategies . . . 107
C.2 Performance of the three KG-to-Text models on the 5L-GOLD-100 testset 108
C.3 Impact of the input Graph Size on the Performance of the FewShot Model 109
C.4 Evaluation of 6 MT models on the EN-SILVER-1K testset . . . 110
C.4.1 High-Resource Languages . . . 110
C.4.2 Low-Resource Languages . . . 110
C.5 Evaluation Results . . . 113
C.5.1 English Subsets Results . . . 113
C.5.2 Chinese Subsets Results . . . 114
C.5.3 Spanish Subsets Results . . . 115
C.5.4 French Subsets Results . . . 116
C.5.5 Russian Subsets Results . . . 117
C.6 Annotator recruitment . . . 118
C.7 Annotation Website . . . 123
C.8 Full Human Evaluation results . . . 124
C.9 Prompting Options . . . 126
C.10 Extended Evaluation Results . . . 134
C.10.1 English Results . . . 134
C.10.2 Chinese Results . . . 134
C.10.3 Spanish Subsets Results . . . 135
C.10.4 French Results . . . 135
C.10.5 Russian Results . . . 136
C.10.6 Breton Results (Br) . . . 136
C.10.7 Irish Results (Ga) . . . 137
C.10.8 Maltese Results (Mt) . . . 137
C.10.9 Welsh Results (Cy) . . . 137
C.11 Hyper Parameters . . . 139

Bibliography **141**

List of Figures

1 Un exemple du graphe de connaissances tiré de la base de connaissances Wikipedia et ses verbalisations en anglais et en français. xxiii

1 Example Knowledge Graph from the Wikipedia Knowledge Base and its verbalisations in English and French. 1

1.1 RDF triple and graph examples 9
1.2 WebNLG entry example 11
1.3 Error hierarchy used by Freitag et al. (2021) 21

2.1 Screenshot of a human annotation website 29
2.2 BLEU scores for each model on English (Left) and Russian (Right) Test Sets. 33
2.3 BLEU score for the different ratios of unseen elements (properties or entities) on English (Left) and Russian (Right) 34
2.4 Error ratios per language 37
2.5 Error ratio per test set. English (Left) vs Russian (Right). 38
2.6 Error ratio per model. English (Left) and Russian (Right). 38

3.1 A graph example from NG_{500} test set, with a graph which provides an advantage to cumulative generation approach. 50
3.2 Distribution of sentence length for each graph size in number of symbols (left) and number of tokens (right) 53
3.3 BLEU Scores for the Best Fine-Tuning, Prompting and RAG models. . . . 55
3.4 Human annotation GUI example 57

4.1 Global Min-Max Normalized Scores 69
4.2 Human evaluation results on the full test set and on each subset by source. 70
4.3 BLEU scores for KG-to-Text Generation on in- (Right) and out-of-domain (Left) data. 76

A.1 BLEU scores for different graph sizes in KELM-E+Pfor English (left) and Russian (right) 86
A.2 Each error progression on KELM-E+P dataset for English. 92
A.3 Each error progression on KELM-E+P dataset for Russian. 93

B.1 SparQL query to get labels of entities 100

B.2 SparQL query to get labels of properties 100

C.1 Global Min-Max Normalized Scores of Subsets by Length. 109
C.2 A screenshot of the website with an annotation example 123

List of Tables

2.1 Examples from KELM dataset . . . 28
2.2 Statistical information on KELM Extracted Subsets . . . 30
2.3 Modification statistics between MT translations and final human translations for KELM and WebNLG test sets . . . 31
2.4 KELM and WebNLG Unseen Benchmarks Statistics. . . . 32
2.5 Graph sizes distribution in each ratio range in KELM-E+P . . . 35
2.6 Error annotation examples for English . . . 39

3.1 Distribution of graph lengths across different categories in the newly created NG_{500} dataset. . . . 44
3.2 Availability of labels and descriptions in Russian across datasets. . . . 45
3.3 Results of (Kazakov et al., 2023)'s Pretrained Model fine-tuned on WebNLG 2020 training data, tested with and without input enrichment . . . 46
3.4 Short Description of the Prompting Options . . . 47
3.5 BLEU Scores for the various LLM-Prompting configurations (FS/I/NLG). 49
3.6 Query variants . . . 53
3.7 BLEU scores for RAG. . . . 54
3.8 Human Evaluation Results . . . 57
3.9 Fleiss' kappa . . . 58
3.10 An example of a graph and three systems' outputs . . . 60

4.1 Selected Translation Models for Each Target Language. . . . 64
4.2 Statistics of the 5L-GOLD-100 testset. . . . 65
4.3 Prompting Approaches . . . 66
4.4 Example One Shot Prompt . . . 67
4.5 P-value of the statistical significance test. *Student's t-test. **Wilcoxon signed-rank test. . . . 68
4.6 Ratio of error type per language and per model . . . 73
4.7 Selected MT models per language. . . . 74
4.8 Best-performing *Instruct* LLM for each target language in the few-shot setting. . . . 75
4.9 Few-shot BLEU on Russian. . . . 76
4.10 Example illustrating high scores for FewShot on English. . . . 79

A.1 Automatic evaluation results on Russian benchmarks . . . 87

A.2 Automatic evaluation results on English benchmarks . . . 88
A.3 Error annotation examples for Russian: addition, omission, repetition and entity distortion . . . 89
A.4 Error annotation examples for Russian: property understanding, topic change, complex text and garbage . . . 90
A.5 Full results of qualitative analysis. . . . 91

B.1 Results of LLMs with different prompting methods and RAG on the seen WebNLG test set . . . 101
B.2 Results of LLMs with different prompting methods and RAG on WebNLG-based unseen test sets . . . 102
B.3 Results of LLMs with different prompting methods and RAG on KELM-based test sets . . . 103
B.4 Results of LLMs with different prompting methods and RAG on NG_{500} test set . . . 104

C.1 Evaluation results on Russian WebNLG Test dataset with various metrics. 107
C.2 All results for all metrics for all languages. . . . 108
C.3 Results on the 1K English test set for translation models across four target languages (Chinese, French, Russian, Spanish). . . . 110
C.4 Similarity metric support for selected low-resource languages. . . . 111
C.5 Support of many-to-many (M2M) MT models for selected low-resource languages. . . . 112
C.6 Precision (%) of Language Identification (LID) for generated translations in low-resource languages. . . . 112
C.7 Synthetic Score Results on the EN-GOLD-1K English test set for translation models across three target languages (Irish (GA), Welsh (CY)), Maltese (MT). . . . 112
C.8 Subset results for English categorized by dataset source and graph length, with all metrics. . . . 113
C.9 Results for Chinese subsets, categorized by dataset source and graph length, with all metrics. . . . 114
C.10 Results for Spanish subsets, categorized by dataset source and graph length, with all metrics. . . . 115
C.11 Results for French subsets, categorized by dataset source and graph length, with all metrics. . . . 116
C.12 Results for Russian subsets, categorized by dataset source and graph length, with all metrics. . . . 117
C.13 Full test set. . . . 124
C.14 WebNLG . . . 124
C.15 Kelm . . . 125
C.16 Wikidata . . . 125
C.17 Few-shot and baseline results on English WebNLG and 5L-GOLD-OOD. . 134
C.18 Few-shot and baseline results for Chinese on 5L-GOLD-OOD. . . . 134
C.19 Few-shot and baseline results on Spanish WebNLG and 5L-GOLD-OOD. . 135

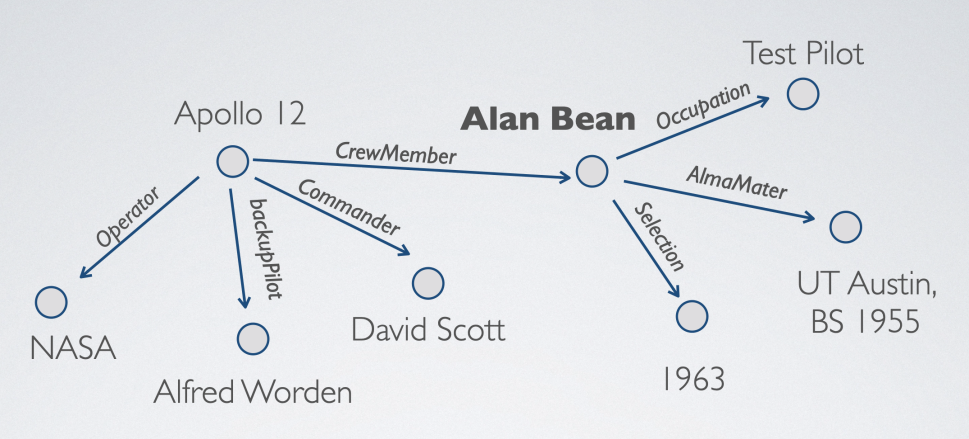
C.20 Few-shot and baseline results for French on 5L-GOLD-OOD. 135
C.21 Few-shot and baseline results on Russian WebNLG and 5L-GOLD-OOD. . 136
C.22 Few-shot and baseline results on Breton WebNLG. 136
C.23 Few-shot and baseline results on Irish WebNLG. 137
C.24 Few-shot and baseline results on Maltese WebNLG. 137
C.25 Few-shot and baseline results on Welsh WebNLG. 138
C.26 Hyper parameters of the different models. 139

List of Abbreviations

AMR	Abstract Meaning Representations
BART	Bidirectional and Auto-Regressive Transformer
BERT	Bidirectional Encoder Representations from Transformers
BLEU	Bilingual Evaluation Understudy
BLEURT	Bilingual Evaluation Understudy with Representations from Transformers
ChrF	Character n-gram F-score
D2TG	Data-to-text Generation
GEC	Grammatical Error Correction
IAA	Intern Annotator Agreement
KELM	Knowledge Enhanced Language Model Pre-training
KG	Knowledge Graph
LLM	Large Language Model
LSTM	Long Short Term Memory unit
MT	Machine Translation
NLG	Natural Language Generation
NN	Neural Network
OOD	Out-of-Distribution
QA	Question Answering
RAG	Retrieval Augmented Generation
RDF	Resource Description Framework
RNN	Recurrent Neural Network
T2TG	Text-to-text Generation
T5	Text-to-Text Transfer Transformer
TER	Translation Edit Rate

Génération de textes multilingues à partir de graphes de connaissances : de la création de données à l'évaluation humaine

Les graphes de connaissances (KG) codifient les connaissances sur le monde en enregistrant des faits qui peuvent être exprimés par des triplets $(e1, p, e2)$, où $e1$ et $e2$ sont des entités ou des valeurs et p désigne une relation binaire également appelée propriété. Les entités peuvent désigner différents lieux, personnes, événements ou concepts. La génération KG-to-Text consiste à convertir des graphes de connaissances en texte, ce qui fait partie du domaine plus large de la génération de langage naturel (NLG). Figure 1 montre un exemple de graphe KG et les verbalisations possibles de ce graphe en anglais et en français.



en Alan Bean graduated from UT Austin in 1955 with a Bachelor of Science degree. He was hired by NASA in 1963 and served as test pilot. He was a crew member of Apollo 12. The backup pilot for Apollo 12 was Alfred Worden, and the commander was David Scott.

fr Alan Bean a obtenu une licence en sciences à l'université du Texas à Austin en 1955. Il a été recruté par la NASA en 1963 et a occupé le poste de pilote d'essai. Il faisait partie de l'équipage d'Apollo 12. Le pilote de réserve d'Apollo 12 était Alfred Worden, et le commandant était David Scott.

Figure 1: Un exemple du graphe de connaissances tiré de la base de connaissances Wikipedia et ses verbalisations en anglais et en français.

La génération KG-to-text peut être intégrée à de nombreuses applications différentes. Elle est utilisée pour faciliter la réponse aux questions (question answering, QA) (Fan et al., 2019; Han and Gardent, 2023), pour générer automatiquement des articles Wikipédia (Mille et al., 2024a), pour améliorer la qualité de la traduction automatique (Conia et al., 2024), ou pour soutenir la génération de dialogues (Zhou et al., 2018). Toutes ces applications diverses rendent importante l'amélioration de la génération de texte à partir d'un KG.

Cependant, les travaux antérieurs sur la génération de texte à partir d'un KG se sont principalement concentrés sur des approches monolingues, la plupart d'entre elles ciblant l'anglais (Schmitt et al., 2021, 2020; Yuan and Faerber, 2023; Yin and Wan, 2022; Schmitt et al., 2021, 2020; Koncel-Kedziorski et al., 2019; Li et al., 2021; Duma and Klein, 2013). Les différentes langues posent des défis différents en matière de génération de texte. Les différentes structures grammaticales, les scripts d'écriture ainsi que les quantités plus faibles de données disponibles dans d'autres langues posent toutes des difficultés dans la génération de textes non anglais.

De plus, de nombreuses recherches ne sont validées qu'à partir de données issues de la même distribution (Yu et al., 2024), ce qui rend difficile l'évaluation de la généralisation des méthodes. La capacité de généralisation est particulièrement importante dans les applications de génération KG-to-text générales, telles que la QA, car les KG peuvent être mises à jour plus facilement et plus rapidement que les méthodes et les modèles entraînés pour générer du texte à partir de celles-ci.

Nous souhaitons donc disposer d'une évaluation fine : premièrement, pour détecter les problèmes exacts liés à la génération de texte dans des langues autres que l'anglais et, deuxièmement, pour détecter les problèmes de généralisation. Nous nous intéressons particulièrement aux aspects du texte généré qui sont affectés par les données non anglaises hors distribution afin de les traiter directement. En termes d'évaluation hors distribution, il existe un défi supplémentaire qui consiste à évaluer les performances des grands modèles linguistiques (Large Language Models, LLM), car les données sur lesquelles ils sont entraînés, en raison de leur volume, présentent souvent des instances similaires (sinon les instances elles-mêmes) aux données de test courantes (Zhou et al., 2023a).

Dans cette thèse, nous explorons la génération de texte à partir de graphes de connaissances dans différentes langues et adoptons une approche d'évaluation qui cherche à comprendre les limites des modèles actuels, y compris les LLM, dans cette tâche.

1 Questions de Recherche et Contributions

Afin d'orienter l'exploration de la génération multilingue KG-to-text et son évaluation, nous avons formulé plusieurs questions de recherche. Nous les présentons dans cette section, et nous répertorions nos contributions pour chaque question de recherche présentée.

RQ1 : Quels types d'erreurs les modèles produisent-ils lorsqu'ils sont évalués sur des données hors distribution pour la génération KG-to-text dans des langues autres que l'anglais ?

Comme de nombreuses recherches sur la génération KG-to-text se concentrent sur l'anglais, les approches existantes ne tiennent souvent pas compte des défis linguistiques posés par d'autres langues que l'anglais. Nous pensons qu'en mettant d'abord en évidence ces défis et en identifiant les erreurs courantes dans la génération vers des langues autres que l'anglais, nous serons en mesure de corriger ces erreurs à l'aide de méthodes appropriées.

Pour répondre à cette question pour le russe, nous avons créé des ensembles de tests basés sur KELM et WebNLG, qui constituent un défi pour les modèles entraînés avec WebNLG en termes de nouvelles entités, propriétés et catégories inconnues. Après avoir effectué une analyse qualitative, nous avons constaté que les modèles russes ont particulièrement du mal à verbaliser les entités issues des graphes de connaissances, ainsi qu'à replacer les propriétés dans le contexte approprié de ces entités (erreur de changement de sujet tirée du chapitre 2).

Par la suite, afin de voir comment cela s'applique à d'autres langues, nous avons créé un autre ensemble de tests inédits basé sur Wikidata, avec des graphes qui ne sont pas encore présents dans d'autres ensembles de données. En l'utilisant, nous avons constaté que pour les langues autres que l'anglais, les omissions sont particulièrement fréquentes pour un LLM centré sur l'anglais sur des données hors distribution. Nous avons également constaté que les modèles de traduction automatique affinés ont généralement du mal à verbaliser les propriétés, tandis que l'approche hybride de génération de texte en anglais et de traduction automatique produit des textes peu fluides, en plus d'avoir du mal à verbaliser les entités et les propriétés.

RQ2 : Comment pouvons-nous corriger ces erreurs avec différentes architectures et en fournissant des informations supplémentaires pour le russe ?

Nous émettons l'hypothèse que fournir des informations supplémentaires au modèle et modifier l'architecture neuronale peuvent aider à cibler les erreurs qui surviennent fréquemment lors de la génération de texte non anglais à partir de graphes de connaissances hors distribution. En prenant l'exemple du russe, nous cherchons à déterminer si le réglage fin, le prompting d'un LLM et le RAG (génération à enrichissement contextuel), tout en mettant à disposition des traductions d'entités ou le contexte dans lequel elles apparaissent, peuvent combler l'écart de performance sur les données hors distribution.

Afin de comparer correctement les capacités de généralisation de ces méthodes, nous introduisons un autre ensemble de tests inédits, contenant cette fois-ci des graphes de connaissances qui n'étaient pas présents dans les autres ensembles de données, ce qui aurait dû représenter un défi plus important pour les méthodes basées sur les LLM, à savoir le prompting et le RAG.

Pour le prompting, nous constatons que la variante la plus performante est une stratégie de sélection en quelques essais (few-shots) basée sur des propriétés similaires, combinée à des informations sur les étiquettes (labels, traductions possibles des entités et des propriétés) et fournissant des few-shots dans la langue cible plutôt qu'en anglais. Pour

le RAG, la même stratégie à few-shots était essentielle pour la variante la plus performante, ainsi que le contexte récupéré sous forme de phrases similaires aux paires d'entités dans le graphe d'entrée. Nous constatons que le prompting et le RAG s'améliorent tous deux en comparant avec un affinage en termes de capacité de généralisation, le prompting ayant tendance à être plus fidèle aux données et le RAG produisant un résultat plus fluide.

RQ3 : Comment les différentes approches multilingues se comparent-elles dans la génération de texte à partir de graphes de connaissances sur des données hors distribution ? Les LLMs peuvent-ils faciliter la génération dans des langues à faibles ressources ?

La prochaine étape consiste à passer du russe à d'autres langues. Nous explorons comment différentes approches multilingues, telles que l'affinage d'un modèle de traduction automatique (Machine Translation, MT), la combinaison d'un modèle de KG-to-texte pour anglais avec MT, ou l'utilisation de LLM prompting, se comparent sur des données en distribution et hors distribution. Nous expérimentons d'abord avec l'anglais, le français, l'espagnol, le chinois et le russe. Nous explorons ensuite les capacités de tous ces modèles pour les langues peu dotées, à savoir le gallois, l'irlandais, le breton et le maltais.

Pour évaluer ces approches, nous créons des données de test inédites de qualité « gold » pour les langues riches en ressources. À l'aide de cet ensemble de test, nous démontrons que l'approche LLM prompting surpasse les autres approches multilingues, tant sur les données en distribution que hors distribution. Sur des données hors distribution inédites, nous avons également pu constater que les LLM bénéficient d'un avantage particulier sur les langues pour lesquelles ils ont été entraînés. En revanche, nous avons constaté que les méthodes basées sur la traduction subissent des pertes d'évaluation substantielles lorsque l'on compare leurs performances dans des contextes en distribution et hors distribution.

Nous avons également testé nos méthodes sur des langues peu dotées. Cependant, pour celles-ci, nous ne disposions que de données WebNLG en distribution pour les ensembles de test. Si toutes les méthodes se sont révélées moins performantes sur les langues peu dotées, les méthodes « few shot » ont enregistré la baisse la moins importante, ce qui indique que les LLMs constituent un moyen prometteur de combler le fossé pour les langues peu dotées.

2 Plan de la thèse

Le chapitre 1 présente le domaine du NLG et fournit les informations nécessaires à la compréhension de la thèse. Nous abordons le NLG en termes d'applications existantes, de méthodes et de méthodologie d'évaluation.

Le chapitre 2 vise à répondre à la question de recherche 1. Pour cela, nous créons de nouvelles données de génération KG-to-text hors distribution et effectuons une analyse qualitative et quantitative sur plusieurs modèles, y compris ceux qui ont obtenu les meilleures performances dans les WebNLG Challenges pour l'anglais et le russe. Nous nous concentrons particulièrement sur les différences de performances induites par les variations dans la distribution des données et la langue.

Dans le chapitre 3, nous abordons principalement la question RQ2 en comparant plusieurs approches, à savoir l'affinage, le prompting et le RAG, sur des données hors distribution. Nous explorons également différents types d'informations supplémentaires, telles que les traductions courantes des entités ou des exemples de phrases dans la langue cible, afin de faciliter la génération ciblant spécifiquement les erreurs de verbalisation des entités et de changement de sujet qui sont fréquentes en russe, comme indiqué au chapitre 2. Afin de comparer ces approches, nous créons un ensemble de tests hors distribution diversifié, en veillant à ce qu'il soit également inédit pour les LLM. Nous effectuons également une évaluation comparative humaine qui fournit des informations sur les points forts et les points faibles des méthodes présentées.

Nous élargissons encore le champ de notre recherche à la génération multilingue de texte à partir de graphes de connaissances (KG) dans le chapitre 4, contribuant ainsi à la question de recherche 3. Nous proposons trois approches différentes pour la génération multilingue KG-to-text, en nous concentrant sur cinq langues riches en ressources (anglais, français, espagnol, russe et chinois), puis en étendant l'analyse à quatre langues peu dotées (gallois, maltais, irlandais et breton). Nous présentons un ensemble de données multilingues pour l'évaluation de la génération de texte à partir de graphes de connaissances hors domaine. En plus de l'évaluation automatique de ces données, nous effectuons une évaluation humaine et une analyse des erreurs, qui contribuent également à la RQ1, sur une plus petite partie des données pour les langues riches en ressources.

Enfin, dans le chapitre 5, nous résumons les résultats et les contributions de ce travail et discutons de ses limites. Nous proposons également des pistes de recherche futures possibles et formulons des remarques finales.

3 Liste des publications

Le contenu de cette thèse repose principalement sur les publications suivantes, évaluées par des pairs :

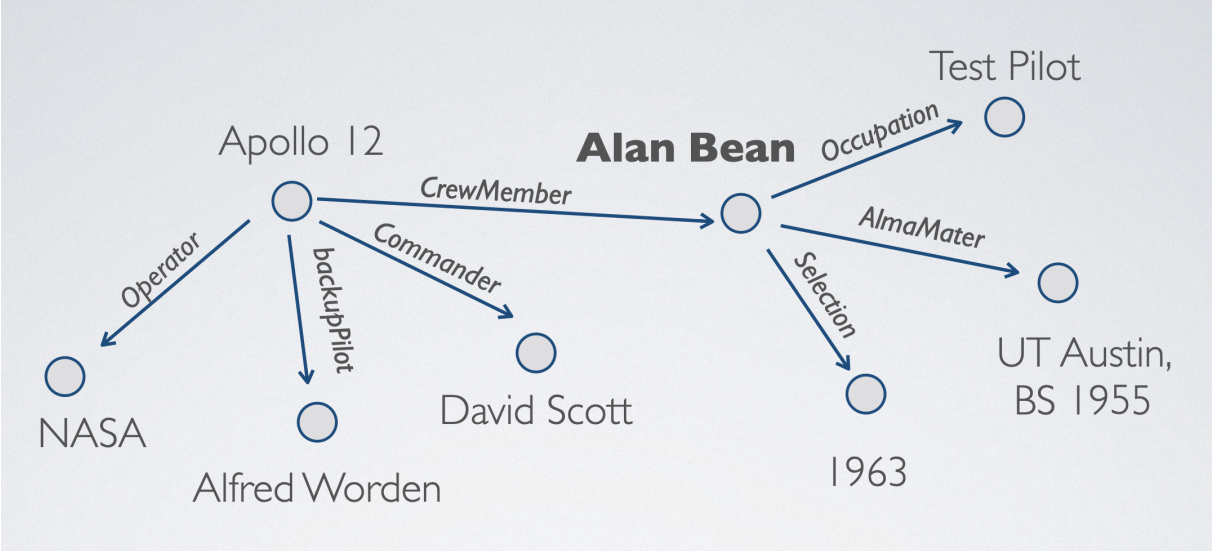
- Anna Nikiforovskaya and Claire Gardent. 2024. Evaluating RDF-to-text generation models for English and Russian on out of domain data. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 134–144, Tokyo, Japan. Association for Computational Linguistics

- Anna Nikiforovskaya, William Eduardo Soto Martinez, Evan Parker Kelly Chapple, and Claire Gardent. 2025. Fine-tuning, prompting and RAG for knowledge graph-to-Russian text generation. how do these methods generalise to out-of-distribution data? In *Proceedings of the 18th International Natural Language Generation Conference*, pages 419–448, Hanoi, Vietnam. Association for Computational Linguistics

- Yifei Song, William Soto Martinez, Anna Nikiforovskaya, Evan Parker Kelly Chapple, and Claire Gardent. 2025. Multilingual verbalisation of knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1111–1162, Suzhou, China. Association for Computational Linguistics

Introduction

Knowledge Graphs (KG) encode knowledge about the world by recording facts where facts can be expressed by $(e1, p, e2)$ triples with $e1, e2$ being some entities or values and p denoting a binary relation also called a property. Entities can denote different places, people, events, or concepts. KG-to-Text generation is the task of converting Knowledge Graphs into text, which is a part of a broader domain of Natural Language Generation (NLG). Figure 1 shows an example KG graph and possible verbalisations of that graph in English and French.



en Alan Bean graduated from UT Austin in 1955 with a Bachelor of Science degree. He was hired by NASA in 1963 and served as test pilot. He was a crew member of Apollo 12. The backup pilot for Apollo 12 was Alfred Worden, and the commander was David Scott.

fr Alan Bean a obtenu une licence en sciences à l'université du Texas à Austin en 1955. Il a été recruté par la NASA en 1963 et a occupé le poste de pilote d'essai. Il faisait partie de l'équipage d'Apollo 12. Le pilote de réserve d'Apollo 12 était Alfred Worden, et le commandant était David Scott.

Figure 1: Example Knowledge Graph from the Wikipedia Knowledge Base and its verbalisations in English and French.

KG-to-text generation can be part of many different applications. It is used to facilitate question answering (QA) (Fan et al., 2019; Han and Gardent, 2023), to automatically

generate Wikipedia articles (Mille et al., 2024a), to enhance the quality of machine translation (Conia et al., 2024), or to support dialogue generation (Zhou et al., 2018). All these diverse applications make it important to improve upon KG-to-text-generation.

However, previous work on KG-to-text generation has mostly focused on monolingual approaches, most of them targeting English (Schmitt et al., 2021, 2020; Yuan and Faerber, 2023; Yin and Wan, 2022; Schmitt et al., 2021, 2020; Koncel-Kedziorski et al., 2019; Li et al., 2021; Duma and Klein, 2013). Different languages pose different challenges in their text generation. The various grammatical structures, writing scripts as well as smaller amounts of available data in other languages all pose difficulties in non-English text generation.

Additionally, a lot of research is only validated against same-distribution data (Yu et al., 2024), making it difficult to understand how well the methods generalise. Generalisation ability is especially important in KG-to-text generation applications like QA, as KGs can be updated easier and faster than the methods and models trained to generate from them.

We are thus interested in having a fine-grained evaluation: first, to detect what are the exact problems in text generation into non-English languages and second to detect issues of generalisation. We are especially interested in what aspects of the generated text are affected by out-of-distribution non-English data to further address them directly. In terms of out-of-distribution evaluation there is an additional challenge of evaluating Large Language Models (LLMs) performance, as the data on which they are trained due to its volume often have similar instances (if not those instances themselves) to common test data (Zhou et al., 2023a).

In this thesis, we explore KG-to-text generation into various languages and adopt an evaluation approach which seeks to understand limitations of current models, including LLMs, on this task.

1 Research Questions

To help guide the exploration of multilingual KG-to-text generation and its evaluation, we have formulated the following research questions.

RQ1: What kind of errors do models produce when evaluated on out of distribution data for KG-to-text generation into non-English languages?

As a lot of research in KG-to-text generation focuses on English, existing approaches often do not account for the linguistical challenges introduced by other, non-English, languages. We believe that by first highlighting those challenges and identifying common errors in generation into non-English languages, we will be able to address those errors with corresponding methods. We first focus on exploring this question while focusing on Russian. However, we further revisit the question by analysing the performance of several models in Spanish, French, Chinese, Russian, and English.

RQ2: How can we address those errors with different architectures and providing additional information for Russian?

We hypothesize that providing additional information to the model as well as varying the neural architecture can help target the errors which occur frequently during non-English out-of-distribution KG-to-text generation. With Russian as an example, we explore whether fine-tuning, prompting and Retrieval Augmented Generation (RAG), while providing entity translations and context where the entities occur, can bridge the gap in performance on out-of-distribution data.

RQ3: How do different multilingual approaches compare in KG-to-text generation on out-of-distribution data? Can LLMs facilitate generation into low-resource languages?

The next step is to move from Russian to other languages as well. We explore how different multilingual approaches, such as fine-tuning a machine translation (MT) model, combining a KG-to-English text with MT, or using LLM prompting compare on in-distribution and out-distribution data. We first experiment with English, French, Spanish, Chinese and Russian. We then explore the abilities of all of those models for low-resource languages, namely Welsh, Irish, Breton and Maltese.

2 Thesis Outline

Chapter 1 introduces the field of NLG and provides the necessary background for the thesis. We cover NLG in terms of existing applications, methods and evaluation methodology.

Chapter 2 aims to answer RQ1. For that we create new out-of-distribution RDF-to-text generation data and run both qualitative and quantitative analysis on several models including the ones with the best performance in WebNLG Challenges for English and Russian. We particularly focus on differences in the performance induced by variations in the data distribution and language.

In Chapter 3 we primarily address RQ2 by comparing several approaches, namely Fine-Tuning, Prompting and Retrieval Augmented Generation (RAG), on out-of-distribution data. We also explore different types of additional information, such as common translations of the entities or example sentences in the target language to facilitate generation targeting specifically entity verbalisation and topic change errors found to be prevalent in Russian in Chapter 2. To compare these approaches we create a diverse out-of-distribution test set, focusing on making it unseen for LLMs as well. We also perform comparative human evaluation that provides some insights on strong and weak aspects of introduced methods.

We further extend the scope of our research to multilingual KG-to-text generation in Chapter 4, contributing to RQ3. We propose three different approaches to multilingual KG-to-text generation, focusing on 5 high-resource language (English, French, Spanish, Russian, and Chinese), later on also expanding the analysis to 4 low-resource languages

(Welsh, Maltese, Irish, and Breton). We introduce a multilingual dataset for out-of-domain KG-to-text generation evaluation. In addition to automatic evaluation on that data, we run human evaluation and error analysis, which also contributes to RQ1, on a smaller portion of data for high-resource languages.

Finally, in Chapter 5 we summarize the results and contributions of this work and discuss its limitations. We also propose possible future directions of the research, as well as provide final remarks.

3 List of Publications

The content of this thesis is primarily based on the following peer-reviewed publications:

- Anna Nikiforovskaya and Claire Gardent. 2024. Evaluating RDF-to-text generation models for English and Russian on out of domain data. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 134–144, Tokyo, Japan. Association for Computational Linguistics
- Anna Nikiforovskaya, William Eduardo Soto Martinez, Evan Parker Kelly Chapple, and Claire Gardent. 2025. Fine-tuning, prompting and RAG for knowledge graph-to-Russian text generation. how do these methods generalise to out-of-distribution data? In *Proceedings of the 18th International Natural Language Generation Conference*, pages 419–448, Hanoi, Vietnam. Association for Computational Linguistics
- Yifei Song, William Soto Martinez, Anna Nikiforovskaya, Evan Parker Kelly Chapple, and Claire Gardent. 2025. Multilingual verbalisation of knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1111–1162, Suzhou, China. Association for Computational Linguistics

Thus, Chapter 2 is based on (Nikiforovskaya and Gardent, 2024), Chapter 3 on (Nikiforovskaya et al., 2025) and Chapter 4 extends (Song et al., 2025). In (Nikiforovskaya et al., 2025) I was responsible for data creation, fine-tuning method, and developed RAG approaches, I also conducted evaluations with a focus on human evaluation and statistical analysis. William and Evan were working on prompting variations. In (Song et al., 2025) my work concentrated on out-of-distribution data creation and human evaluation and analysis. In addition, I was responsible for the NLG+MT approach. Yifei was responsible for MT selection and automatic evaluation, William for MT fine-tuning and William and Evan were working on prompting variations.

1

Background

Contents

1.1 Introduction to Natural Language Generation 6

1.1.1 Text-to-text generation 6

1.1.2 Data-to-text generation 7

1.2 Data for KG-to-text generation 10

1.2.1 Automatically aligned datasets 10

1.2.2 Human annotated datasets 10

1.2.3 Unsupervised datasets 12

1.3 Natural Language Generation Methods 12

1.3.1 Pre-neural methods 12

1.3.2 Neural methods 13

1.3.3 Prompting methods 15

1.3.4 RAG 15

1.4 Evaluation 16

1.4.1 Automatic Evaluation 16

1.4.2 Human Evaluation 20

1.4.3 LLM-as-a-judge for Evaluation 22

1.5 Conclusion 23

In this chapter, we provide the necessary background for the thesis. We discuss the tasks of Natural Language Generation (NLG), focusing specifically on the challenges and applications of data-to-text generation. We cover the historically and commonly used methods for NLG, ranging from rule-based systems to modern Large Language Models (LLMs) and Retrieval Augmented Generation (RAG). Finally, we survey the commonly used datasets and detail the quantitative and qualitative evaluation methods, both automatic and human-curated, used to assess the quality of generated text.

1.1 Introduction to Natural Language Generation

The tasks of Natural Language Generation (NLG) concern producing texts in natural languages (e.g. English, Russian, French), which follow specified rules and qualities determined by a particular task. The NLG family of tasks can be divided into two different areas according to the source of generation; text-to-text generation and data-to-text generation (Gatt and Krahmer, 2018).

1.1.1 Text-to-text generation

Text-to-text generation (T2TG) tasks are the tasks which provide natural language texts as a source. One of the oldest and most common examples of such a task is Machine Translation (MT) (Wang et al., 2022a). Aside from the daily use of MT systems for general purposes (Google Translator (Pitman, 2021), DeepL [1], etc), there are many domain-specific applications, such as MT adapted for Social Networks (Gupta et al., 2023), MTs for biomedical texts (Neves et al., 2024) or news translation (Akhbardeh et al., 2021). Although in all cases we expect to have translations which are both faithful (or adequate) to the source and fluent in the target language (Castilho et al., 2018), domain-specific translations may be subject to additional constraints, such as translations of scientific terms as they are commonly used in the needed scientific area, or the same style and emotional tone of the Social Media text is preserved.

Another common task in the area of T2TG is text simplification, with the goals of preserving facts and meaning while simplifying the language (Al-Thanyyan and Azmi, 2021). It is applied to facilitate government communication (Beks van Raaij et al., 2024), make scientific texts accessible to the general public (Engelmann et al., 2023; Ermakova et al., 2022) or to help non-native language speakers (Paetzold and Specia, 2016). All simplification methods are expected to have a degree of faithfulness to the original text, however, the expectations of the level and type of simplification vary on the area of application.

Text summarization task is commonly applied to news articles (Ahuja et al., 2022; Zhang et al., 2024), scientific articles (Cohan and Goharian, 2015; Wang et al., 2022b) and other types of documents (Shukla et al., 2022; Tian et al., 2024) to extract the most important information from one or multiple text documents and to further process it. Specific evaluation criteria vary based on goals of the summarization, favouring specific aspects of the document such as more general points for news summarization, or scientific aspects in scientific summarization, however, in the majority of cases the resulting summaries are expected to be fluent, coherent and faithful to the original documents. It is also important that the summaries are not too long, so the goal becomes to also minimize redundancy.

Widely used tools for text correction such as Grammarly [2], MerciApp [3], or Reverso [4] are examples of application of Grammatical Error Correction (GEC), another T2TG task.

[1] https://www.deepl.com/en/quality
[2] https://www.grammarly.com/grammar
[3] https://www.merci-app.com/
[4] https://www.reverso.net/spell-checker/

GEC is commonly targeted at helping second-language learners (Koyama et al., 2020; Kaneko et al., 2022), however native speakers also use text correction tools to deal with occasional misspellings or to make their texts more appropriate for a specific setting, such as academic writing (Qub'a et al., 2024). Evaluation of the quality of GEC is a big task in itself, as one text may contain various, sometimes overlapping grammatical errors, but also the text cannot be changed too much, GEC is not allowed to introduce or omit facts (Sakaguchi et al., 2016).

In summary, we can see that text-to-text generation can be used in various domains and for multiple tasks, each of which imposes different requirements on the quality and desired features of the produced text.

1.1.2 Data-to-text generation

We define data-to-text generation (D2TG) tasks as tasks which require generation of texts based on non-textual data, which might be both unstructured and structured. Here we consider the data to be structured if it is organised in a pre-defined manner, having a pre-defined data model. Thus, we consider sets of images or videos to be unstructured data.

Generation from unstructured data

Image captioning is a task among the family of unstructured D2TG tasks. It can be used to label the pictures to improve their indexing for content-based image retrieval (Hossain et al., 2019) or to facilitate understanding for visually impaired people (Ahsan et al., 2021). Although many different methods were introduced (and in application to specific domains such as news (Liu et al., 2024b)), their evaluation remains difficult, as it is not always clear what aspects of the image should be included in the caption, so people are constantly working on evaluation in the field, and creating new metrics (Cui et al., 2018; Hessel et al., 2021; Lee et al., 2024).

Closed caption generation is another challenging task that can be considered a part of video-to-text generation. Closed captions are a useful tool for deaf and hard-of-hearing people, people watching videos or TV in their second language and even native speakers, desiring to watch videos without disturbing others (Fedorchenko and Alumäe, 2025). The caption have to be not only accurate to the speech and to what happens in the video, but also have to appear at the right time and be in easily digestible chunks for the viewers (Arroyo Chavez et al., 2024). Another important aspect of quality in closed caption generation is the presence of non-speech information, such as animal sounds, music playing in the background, as well as meta-information like who is currently speaking (May et al., 2024). All these aspects need to be accounted for during evaluation.

Generation from structured data

We have seen some examples of unstructured data-to-text generation and their evaluation challenges. Another big part of data-to-text generation is structured data-to-text, which includes generation from organised records, tables and graphs.

The oldest examples of structured D2TG involve generation from structured records. This includes for instance the generation of automatic commentary for games from sportscasting data, such as soccer (Chen and Mooney, 2008) and the generation of weather forecasts based on the set of records (Sripada et al., 2004; Liang et al., 2009). These tasks are challenging in addition to the complexity of generating fluent and cohesive text. For example, in automatic sports commentary it is crucial to properly decide what events to include in the commentary, while in weather forecast there is a need to draw complex conclusions from several records. These are the content selection problems that are a common issue in D2TG (Gkatzia, 2016). Thus, even though those tasks were formulated a long time ago, there is still work produced in those areas, both expanding the domain of application, for example, for basketball review generation (Thomson et al., 2020) or news generation (Leppänen et al., 2017), and in terms of methods, such as incorporating temporal information (Gong et al., 2019). Evaluation criteria in generation from structure records includes, aside from quality of the produced text itself, faithfulness to the data and aspects related to content selection, such as newsworthiness for news generation or sports commentary.

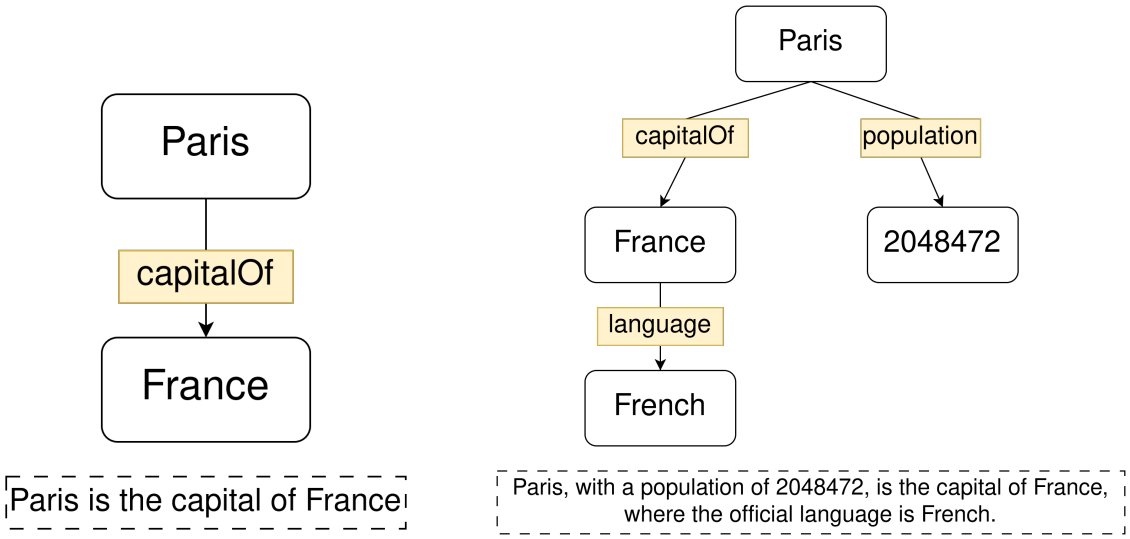
Originally introduced as a means to express text meaning in a machine-readable graph format, Abstract Meaning Representations (AMR) (Banarescu et al., 2013) has encouraged people to use it as an intermediate step for various tasks, such as text summarization, question answering and machine translation (Tohidi and Dadkhah, 2022). As a result, AMR-to-text generation became another important direction of D2TG (Pourdamghani et al., 2016; Konstas et al., 2017; Song et al., 2018), with recent works focusing on multilingual generation (Fan and Gardent, 2020; Soto Martinez et al., 2024). In terms of quality expectations, text generated from AMR is required to be as close to AMR graphs as possible.

Another way to represent texts in a machine-readable graph format is using dependency trees. While AMR graphs have a focus on semantic representation, dependency trees focus on syntactic representation. The Universal Dependencies (UD) project (Nivre et al., 2016) standardized the format of dependency trees for cross-lingual application and gathered treebanks for multiple languages. This led to the Multilingual Surface Realisation Shared Tasks (Mille et al., 2018, 2019), which popularized text generation from dependency trees. Evaluation, similarly to AMR-to-text generation, focuses not only on readability of the generated text but also its closeness to the dependency tree.

A very common example of structured D2TG involves generation specifically from Wikipedia tables, as Wikipedia contains a lot of fairly reliable knowledge in multiple languages, making it an attractive source of knowledge for D2TG systems. While some aim to generate short one-sentence biographies (Lebret et al., 2016), others aim to generate Wikipedia sections based on those tables (Chen et al., 2021). While in these two examples the text is expected to be accurate to the data, it is not expected to verbalise the tables in full. Many question answering (QA) systems use additional sources of knowledge, such as Wikipedia tables, to facilitate reliability and faithfulness (Pasupat and Liang, 2015). But there are also other sources of information and formats for how this additional knowledge can be represented.

Resource Description Framework (RDF) was introduced by Lassila and Swick (1999) as a means to represent data on the Web in a machine-understandable format. They proposed a format which allows to represent data as a set of statements, each of which

can be represented as a triple of a subject, predicate and object, with predicate denoting the relationship between subject and object. Examples of an RDF triple, as well as an RDF graph with their possible lexicalisations can be seen in Figure 1.1. The simplicity of the proposed framework led to creation of several ontologies, and such huge projects as DBPedia (Auer et al., 2007) and Wikidata (Vrandečić and Krötzsch, 2014) covering vast amounts of information in knowledge bases in RDF format.



(a) (b)

Figure 1.1: RDF triple example (1.1a) and RDF graph example (1.1b). Predicates are highlighted with yellow.

Similarly to Wikipedia tables, the availability of large amounts of human-verified data makes it an attractive source of data for multiple applications, such as providing additional context in QA systems (Han and Gardent, 2023), automatic generation of Wikipedia articles (Mille et al., 2024a) or to enhance MT systems (Conia et al., 2024). As such, text generation from knowledge graphs (KGs) became another direction of structured data-to-text generation. In order to stimulate KG-to-text generation, WebNLG challenge (Colin et al., 2016) was introduced, which requires the text to be the exact verbalisation of the small RDF graphs.

Structured knowledge data can also be domain specific. For example, E2E Challenge (Novikova et al., 2017) focuses on restaurant domain, asking systems to generate text summarizing given information about restaurants. They represent the data using their own text-based meaning representations.

While all generation from structured data usually requires the generated text to be accurate in regards to input data, some tasks are in effect akin to summarization, calling for summarizing the input, while other tasks are set to verbalise the input in full, without omitting anything. In this thesis, we focus on such full verbalisation of RDF data, thus putting an emphasis on semantic faithfulness of the text to the data (i.e. how closely the text follows information given in the data).

1.2 Data for KG-to-text generation

As we discussed in the previous section, data can be of different forms and from different sources for data-to-text generation. In this section, we will discuss different datasets that exist for KG-to-text generation, covering their creation processes, as well as discussing their strong and weak points. As tables are also commonly used as knowledge representation and can often be represented in a graph form (and vice versa, the graphs can be represented in a table form), we also include in our discussion some datasets for table-to-text generation.

We group the datasets by the alignment of the texts to KG. Specifically, we first discuss automatically aligned data, then human annotated data, and finally we discuss existing unsupervised datasets, which contain the knowledge graphs, but not the references.

1.2.1 Automatically aligned datasets

Some datasets are created by automatic alignment, using various heuristics to assign texts to the corresponding data. For example, the WikiBio dataset (Lebret et al., 2016) is created by taking the first sentences of the biography articles and aligning them with the Wikipedia infoboxes of those articles. This dataset was later extended upon by automatically aligning whole abstracts of biography articles to the Wikipedia infoboxes, resulting in WikiGen dataset (Perez-Beltrachini and Lapata, 2018).

Another example of automatically aligned dataset is WikiTableT (Chen et al., 2021), which, in addition to Wikipedia tables and infoboxes, aligns additional information to sections in the Wikipedia article, such as hyperlinks, named entities, and Wikipedia article structure. Although automatic filtering may help with the quality of the data, such automatic alignment is far from perfect alignment, as the information in the texts does not ideally correspond to the information in the data.

In T-REx (Elsahar et al., 2018) Wikipedia abstracts are aligned with graphs from Wikidata using entities information and an assumption that all sentences in the same paragraph implicitly mention the core entity from the beginning of a paragraph. Similarly, GenWiki (Jin et al., 2020) is constructed by aligning DBPedia graphs automatically based on hyperlinks and entities found in the Wikipedia article.

Knowledge graphs can also be aligned to automatically generated texts, as in KELM dataset(Agarwal et al., 2021). To obtain it, the authors first created a TeKGeN dataset, where Wikidata graphs were automatically aligned to sentences from English Wikipedia. Then, they trained a KG-to-text model on that data, fine-tuned it on WebNLG (Gardent et al., 2017a) and created KELM by automatically generating texts based on knowledge graphs from Wikidata. This results in large yet silver data, which can work well for training.

1.2.2 Human annotated datasets

A common way to actually achieve a perfect alignment between data and text is to use human annotation. One way is to ask the workers to write the needed information for the data, which was done, for example, by Pasupat and Liang (2015) to obtain questions

```
<entry category="Food" eid="Id1" shape="(X (X) (X) (X))" shape_type="sibling" size="3">
  <originaltripleset>
    <otriple>Ajoblanco | country | Spain</otriple>
    <otriple>Ajoblanco | alias | "Ajo blanco"@en</otriple>
    <otriple>Ajoblanco | ingredient | Garlic</otriple>
  </originaltripleset>
  <modifiedtripleset>
    <mtriple>Ajoblanco | country | Spain</mtriple>
    <mtriple>Ajoblanco | alternativeName | "Ajo blanco"</mtriple>
    <mtriple>Ajoblanco | ingredient | Garlic</mtriple>
  </modifiedtripleset>
  <lex lang="en" lid="Id1">Ajoblanco, also know as "Ajo blanco" , is made with garlic and comes from Spain.</lex>
  <lex lang="ru" lid="Id1">Ахобланко, также известный как "ахобланко", сделан из чеснока и родом из Испании.</lex>
  <lex lang="en" lid="Id2">Ajoblanco (alternatively known as Ajo blanco) comes from Spain and has garlic as an ingredient.</lex>
  <lex lang="ru" lid="Id2">Ахобланко (также известный как ахобланко) родом из Испании и имеет чеснок как ингредиент.</lex>
  <lex lang="en" lid="Id3">Ajoblanco originates from Spain, is also called "Ajo blanco" and has garlic as one of it's ingredients.</lex>
  <lex lang="ru" lid="Id3">Ахобланко родом из Испании, также называется Ахобланко и имеет чеснок как один из ингредиентов.</lex>
  <dbpedialinks>
    <dbpedialink direction="en2ru">Ajoblanco | sameAs | Ахобланко</dbpedialink>
    <dbpedialink direction="en2ru">Spain | sameAs | Испания</dbpedialink>
    <dbpedialink direction="en2ru">Garlic | sameAs | Чеснок</dbpedialink>
  </dbpedialinks>
  <links>
    <link direction="en2ru">Ajo blanco | sameAs | ахобланко</link>
  </links>
</entry>
```

Figure 1.2: En entry example from the Russian dataset in WebNLG 2020. In addition to RDF triples themselves and lexicalisations in English and Russian, the dataset also contains translation information (via dbpedia links), and information on the shape of the graph and its category.

and answers corresponding to the tables. For their E2E dataset Novikova et al. (2017) have also used an intermediate step of visualising the data in a picture before giving it to the crowdsource workers to produce the text based on it. However, crowdsource workers may omit or add extra information when producing texts based on data covering multiple entities. Dušek et al. (2019) showed that cleaning E2E with a slot-matching algorithm improves the performance of D2TG models, even though the cleaning is still not perfect.

In 2017 Gardent et al. (2017a) introduced a dataset for a WebNLG challenge (Gardent et al., 2017b), which further became a foundation for a grand portion of RDF-to-text research as well as for the inverted task of text-to-RDF semantic parsing. The dataset consists of graphs taken from DBPedia and their lexicalisation was obtained via crowdsourcing.

Later, the dataset was extended to include Russian lexicalisations via MT and their crowd-sourced verification (Shimorina et al., 2019). The dataset also contained information on common Russian translations of entities and properties, obtained from DBPedia. An example of a Russian entry in the WebNLG xml format can be seen in fig. 1.2.

After that, the WebNLG dataset has obtained translations of the lexicalisations into several low-resource languages, namely Breton, Irish, Maltese, and Welsh, for the WebNLG 2023 challenge (Cripwell et al., 2023). In addition, translations into other languages of the WebNLG dataset appear in the community, such as translation into Spanish (Ramón-Ferrer et al., 2025) and Italian (Oliverio et al., 2025).

An interesting way to combine automatic alignment with human annotation and verification can be shown by ToTTo dataset (Parikh et al., 2020), which consists of tables with highlighted parts and sentences which can be derived from those tables focusing on highlighted parts. The authors first align Wikipedia tables and sentences automatically, then they ask annotators to highlight parts in the table corresponding to the sentence and finally they ask annotators to edit the sentence.

To create DART dataset, Nan et al. (2021) combine several existing datasets: two open-domain question answering datasets (WikiSQL (Zhong et al., 2018) and WikiTable-

Questions (Pasupat and Liang, 2015)), Cleaned E2E dataset (Dušek et al., 2019) and WebNLG 2017 English dataset (Gardent et al., 2017b). They convert all the datasets to set of triples aligned with lexicalisations. For that the table-based datasets are adapted to triples using tree ontology derived from those tables after extensive human annotation. All the properties are then unified across different sources and the corresponding lexicalisations are obtained via crowdsourcing when they are not directly obtainable from the source data.

1.2.3 Unsupervised datasets

GEM shared task 2024 (Mille et al., 2024c) has also introduced new RDF test data, containing in addition to factual data from Wikidata contrafactual and fictional data, to test the generation of the texts in separation from the world knowledge obtained by many language models in the pre-training stage. However, the datasets they introduced do not contain gold text references.

1.3 Natural Language Generation Methods

Numerous approaches to text generation have been introduced. In this section, we are going to cover them historically, starting from pre-neural methods, which include statistical and rule-based methods and then going on to neural methods, starting from fine-tuning-based methods and ending with prompting and Retrieval Augmented Generation (RAG). While discussing those methods, we will focus on their applications to data-to-text generation and specifically to RDF-to-text generation.

1.3.1 Pre-neural methods

The first methods introduced for D2TG were based on patterns or rules manually extracted from the target language (Goldberg et al., 1994; Reiter et al., 1995). Implementation of these rule-based systems usually consisted of several modules, each performing its own task towards the text generation, starting from content planning and ending with surface realisation. Reiter and Dale (1997) elaborated on 6 commonly found modules of rule-based systems: content determination, discourse planning, sentence aggregation, lexicalisation, referring expression generation, and linguistic realisation. Extracted patterns depended not only on a target language but also on a communication goal, for example, in automatic instruction generation (Reiter et al., 1995) the grammatical structures used are limited due to the common specifics of the language in manuals. Moreover, these rules may be complex for their surface realisation, or lexicalisation, which led to people trying different approaches such as introducing different types of templates (McRoy et al., 2003).

Principal drawbacks of rule-based systems are low linguistic variability, which can result in perceived lack of fluency, and extensive labour needed to mine the patterns and rules based on requirements and the language. Nevertheless, rule-based systems are still often used on their own or as part of larger NLG systems to help deal with low-resource

languages or lack of training data (Mille et al., 2023; Mahlaza et al., 2024; Mille et al., 2024b).

One way to deal with extensive human labour to produce language patterns and rules is to deduct those rules automatically using statistical approaches (Gatt and Krahmer, 2018). For example, Langkilde and Knight (1998) introduce their Nitrogen grammar to generate from AMR graphs, and they use statistical information to make various linguistic choices. Similarly, Belz (2008) also uses corpus-level statistical information in the introduced grammar, but more intrinsically, making the built grammar probabilistic.

Another way to use statistical information is to treat generation itself as a stochastic process. As such Liang et al. (2009) model the generation process as a Markov process, making decisions depend on the current state, combining high-level (such as content planning) and low-level (specific word choice) information.

Text generation can also be treated as an optimisation problem, for example Dethlefs and Cuayáhuitl (2011) combines Hidden Markov Model (HMM), which was first introduced for speech-to-text generation (Jelinek, 1976), with reward optimisation, where rewards are induced from human data.

While statistical models have vastly improved scalability and made generalisation to different domains easier, they often fail to incorporate longer context, leading to fluency and coherence problems. However, some ideas introduced by statistical methods are used in neural-based methods.

1.3.2 Neural methods

Even though the idea of neural networks for NLG was introduced as early as 1987 (Gatt and Krahmer, 2018), it got a massive boost with the development of computational resources and with the availability of data much later.

One of the first examples of use of neural networks to model the language was introduced by Bengio et al. (2003), who have also proposed a way of learning a distributed representation for words, aka word embeddings. However, Feedforward Neural Networks (FNN) as introduced by Bengio et al. (2003) have a drawback of having a fixed-length context. So, later on, there were introduced Recurrent Neural Networks (RNN) for language modeling which theoretically could store limitless context making use of a recurrent structure (Mikolov et al., 2010). The idea behind the recurrent structure is that the reuse of certain ‘hidden‘ vectors in the architecture to produce new state of the NN can help pass the information to account for a longer context. Later Sutskever et al. (2011) introduced a character-level RNN to generate the text, thus showing the potential of RNNs for the NLG.

One of the most important advancements in RNN application was the introduction of a sequence-to-sequence (Seq2Seq) model (Sutskever et al., 2014). In their work Sutskever et al. (2014) showed the potential of RNN models, as they attained good quality of Machine Translation using the new Seq2Seq model with Long Short Term Memory units (LSTMs) (Hochreiter and Schmidhuber, 1997). Specifically, they used one LSTM to encode the input sequence and another LSTM to "decode" into the output sequence. A similar RNN based encoder-decoder model was also proposed by Cho et al. (2014) for MT. In both of those approaches, however, a fixed-length context vector is used to

compress all the information in the input sequence, which is a bottleneck when dealing with longer sequences. To counter that drawback Bahdanau et al. (2014) introduced attention mechanism, generating not a single context vector but a sequence of them focusing on the important subset of them for the decoder.

These ideas first used for MT were later applied to D2TG by transforming input data into a sequence. For example, in WebNLG Challenge 2017 there were several participant systems using Seq2Seq architecture with attention mechanism (Gardent et al., 2017b), where they linearize the graph in different ways, e.g. by introducing special tokens to signify the start of each part of a triple. Similarly, when generating from tables, each table can be linearized as a set of pairs (field name, field content) (Sha et al., 2018).

The encoder-decoder architecture can also be generalized to introduce more data-specific encoders without the need for linearization. As such Ribeiro et al. (2019) use a Graph Neural Network (GNN) encoder to generate texts from AMR. Tables can also have their specific encoders, for example Liu et al. (2019) introduced an attribute-level LSTM to account for the table structure.

The earlier introduced attention mechanism laid the ground for the development of a transformer model (Vaswani et al., 2017), which eliminates recurrence in favour of using a multi-head attention mechanism. In turn, this led to the development of several powerful pre-trained language models (PLMs), such as GPT (Radford et al., 2018), BERT (Devlin et al., 2019), BART (Lewis et al., 2020), T5 (Raffel et al., 2020). The pre-trained models differ in what kind of pre-training tasks and strategies are used as well what kind of modifications to the original transformer structure are implemented. While the original transformer is an encoder-decoder architecture, some pre-trained models opt to use only the decoder, like GPT, while others utilise solely the encoder part, e.g. BERT.

The pre-trained language models opened a possibility to adapt them to various tasks by fine-tuning on smaller amounts of data without losing the possibility to generate cohesive text. One way is a full fine-tuning, as did Agarwal et al. (2020) by fine-tuning a pre-trained T5 model on multiple tasks to aid RDF-to-text generation for English and Russian. Another way is parameter efficient fine-tuning, which either does not alter the original weights of the PLM at all introducing additional elements to adapt it, or which only alters some layers of the PLM leaving the rest unchanged. An example of the former is the work of Clive et al. (2022), where they introduce control prefixes, which are incorporated into the base PLM (T5 or BART) to include the input-dependent information. The latter can be illustrated by Gheini et al. (2021), where they fine-tune only cross-attention layers instead of the full PLM fine-tuning for machine translation.

PLMs were later developed into Large Language Models, by introducing larger models, larger datasets and varying tasks as well as training methods. Models such as LLama (Touvron et al., 2023; Grattafiori et al., 2024), Mistral (Jiang et al., 2023) and GPT (Brown et al., 2020b) have shown great performance on various NLG tasks (Kumar, 2024; Zhang et al., 2025). While some text generation methods opt to fine-tune LLMs for down stream tasks (Chung et al., 2024), a straightforward use of LLMs via providing a prompt can also produce good results.

1.3.3 Prompting methods

As LLMs are trained on a variety of tasks and instructions, LLMs can produce good-quality results on many tasks without fine-tuning but by providing the instructions to them. This approach is called prompting and it became wide-spread, being used for different tasks and in different domains.

There are multiple ways of writing the prompt with the same meaning for a human, and it was shown LLMs are generally sensitive to the phrasing (Yin et al., 2023) and the required output format (Long et al., 2025). Moreover, order of information in the prompt (Pezeshkpour and Hruschka, 2024) and position of key information for the output (Liu et al., 2024a) can influence the performance of LLMs. Razavi et al. (2025) have even introduced a task on prompt sensitivity prediction to facilitate research on the effects of the prompt variation.

To improve the outcome, many different approaches to prompting LLMs were introduced in NLG. Some approaches focus on optimizing or generating the prompt automatically (Deng et al., 2022; Zhou et al., 2023b). Other approaches focus on incorporating additional information or utilising complex pipelines resulting in multi-step instructions. Few-shot examples were shown to reduce LLM sensitivity to prompt variations (Zhuo et al., 2024). Another introduced approach is Chain-of-Thought (Wei et al., 2022), which provides reasoning examples, thus guiding the LLM to produce right answers through reasoning.

Several studies worked on applying LLM prompting to KG-to-text generation. Axelsson and Skantze (2023) experimented with providing prompts followed by linearized RDF graphs to gpt-3.5 (Brown et al., 2020a) without providing any other information (i.e. in a 0-shot setting). They showed the potential of LLM prompting in KG-to-text generation, while highlighting the issue that the LLM tends to add information not present in the graph.

Schneider et al. (2024) compare different LLMs in zero-shot and few-shot settings, the latter including several examples of a desired communication between the user and an LLM, specifically providing the knowledge graph and its desired lexicalisation. They showed that few-shot prompting outperforms zero-shot prompting.

He et al. (2025) improve on few-shot selection for KG-to-text generation by proposing a method for a difficulty-diversity balanced few-shot selection, where they assign difficulty levels to different instances in the dataset based on the graph similarity to the desired output texts, and the diversity is measured via different k-means clusters.

In this thesis, we experiment with different prompting strategies and providing additional information to apply LLMs for multilingual RDF-to-text generation.

1.3.4 RAG

As LLMs struggle to produce domain-specific responses or to incorporate new knowledge, many works leverage external knowledge databases to augment LLMs, leading to Retrieval Augmented Generation (RAG) (Gao et al., 2023; Wu et al., 2024).

The idea of RAG has appeared before the popularity of PLMs and LLMs to make the language model focus on the specific knowledge. For example, Chen et al. (2017) retrieved

paragraphs from Wikipedia similar to the questions and trained an RNN model to use those paragraphs to produce answers for open-domain QA. Some RAG-based approaches using LLMs still opt for extra fine-tuning, to better adapt the model for a specific task and domain (Asai et al., 2024; Lin et al., 2024).

Other RAG-approaches are used directly to aid LLM prompting. To improve medical question answering without extra training, Xiong et al. (2024) compared different settings of RAG, such as different source medical corpora, retrieval algorithms, and the backbone LLM. They showed the importance of each setting; for instance, they found that the best retriever for their task was the one specifically trained for biomedical area (Jin et al., 2023).

While many works in RAG focus on QA, there is also work done with RAG in other areas. For example, in MT there are approaches of using RAG to introduce similar translation examples (Hoang et al., 2023), add domain-specific knowledge with KGs (Chen et al., 2024), or by providing unstructured knowledge (Wang et al., 2024a).

For KG-to-text generation Jobanputra and Demberg (2024) use a symbolic retriever to aid the LLM with graphs and their lexicalisations based on similar properties in those graphs.

We experiment in our work with RAG to improve KG-to-text generation in Russian, we augment graphs with example sentences taken from Wikipedia, which are similar to those graphs.

1.4 Evaluation

As we have seen in the above written sections, different text generation tasks require different aspects of quality. To estimate whether those quality requirements are met, different evaluation techniques have been proposed. First, we discuss the currently used automatic metrics, then how people are used to verify the quality, and finally we talk about LLM-as-a-judge evaluation.

1.4.1 Automatic Evaluation

We divide all automatic metrics into several categories, based on the principles of their work. First, all metrics can be divided into input-independent and input-dependent metrics. The former requires references to compare the generated text to, while the latter are the metrics which take into account the input and the generated text often without the need of the gold-quality references.

Some automatic metrics are constructed around surface-level information, such as tokens and characters in the text. Others are semantic-based, i.e. working by first extracting some underlying information from the text, and then using that information to produce the score.

While there is a variety of those metrics in NLG, in this section we will focus on those which are used in KG-to-text generation and which can be used in multilingual setting.

Input-independent surface-based metrics

The metrics introduced in this section are all based on comparison of a generated text to one or several gold-quality texts (references) using surface-level information, such as tokens, n-grams (sequences of n tokens), and characters.

BLEU score, which was first introduced in Papineni et al. (2002a), is a reference-based metric, which compares the n-grams present in the generated text to the n-grams present in the reference. It is commonly used in Machine Translation and other text generation tasks. To retrieve the score, it first involves calculating a modified n-gram precision p_n by counting the number of times each n-gram occurs in the generated text:

$$p_n = \frac{\sum_{C \in \{\text{Candidates}\}} \sum_{\text{n-gram} \in C} Count_{clip}(\text{n-gram})}{\sum_{C' \in \{\text{Candidates}\}} \sum_{\text{n-gram'} \in C'} Count(\text{n-gram'})} \tag{1.1}$$

However, as the metric is precision-based, it favours smaller generated texts, so they have also introduced a brevity penalty.

$$BP = \begin{cases} 1 & \text{if } c \geq r \\ e^{(1-r/c)} & \text{if } c < r \end{cases} \tag{1.2}$$

Thus, the final BLEU score can be expressed as

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log(p_n)\right) \tag{1.3}$$

While BLEU is extremely popular for evaluation in NLG tasks, it has a lot of drawbacks. Designed first as a quick diagnostic evaluation of MT systems, it is criticized for use in other tasks (Reiter, 2018). Additionally, as the metric is precision based, its performance is dependent on number of references (Post, 2018). Besides, the fact it is an n-gram based metric makes it also dependent on the tokenizer used.

Translation Edit Rate (TER) score Snover et al. (2006) introduced an automatic metric that counts the number of edits needed to transform the generated text into a reference text.

$$TER = \frac{\#\text{edits}}{\text{average } \# \text{ of reference words}} \tag{1.4}$$

The edits include the insertion, deletion, and substitution of single words as well as shifts of word sequences (i.e. moving a sequence of words in one location in the hypothesis to another).

ChrF++ The character n-gram F-score (ChrF) was introduced by Popović (2015), the idea was to combine precision and recall on character n-grams, as in the formula below:

$$\text{ChrF}\beta = (1 + \beta^2)\frac{\text{ChrP} \cdot \text{ChrR}}{\beta^2 \cdot \text{ChrP} + \text{ChrR}} \tag{1.5}$$

Later on it was modified (Popović, 2017) to fix $\beta = 2$ and to include both character and word n-grams, which resulted in the metric ChrF++. This metric relies much less on tokenization comparing to BLEU, which made it common in multilingual evaluations.

Another metric sometimes used in KG-to-text generation is **ROUGE**, originally created for summarization (Lin, 2004). Similarly to BLEU, it is an n-gram based metric, however, in contrast to BLEU it focuses on recall instead of precision, which is why it was used, for example, in E2E Challenge (Novikova et al., 2017), as the challenge did not require to verbalise the whole knowledge graph, making it similar to a summarization task.

However, these surface-based metrics often struggle with performance on languages with a less strict word order, they do not account for paraphrases or synonyms. Denkowski and Lavie (2014) addressed it by introducing paraphrase tables and counting tokens in the same synonym group as the same, to calculate their **Meteor score** (Banerjee and Lavie, 2005), which was first introduced a simpler word-matching metric. However, this approach is not easily extended to new languages.

Input-independent semantic-based metrics

Metrics in this section are model based, relying on embeddings or trained models to produce a similarity score between the generated text and references.

SBERT score (Reimers and Gurevych, 2019) is a similarity measure obtained by first computing sentence-level embeddings and then using a cosine similarity of those embeddings to produce a score.

BERT score introduced by Zhang* et al. (2020) is a reference-based automatic metric, which is based on comparison of pre-trained BERT contextual embeddings (Devlin et al., 2019) for the generated and the reference texts. Specifically, the similarity of two sentences is computed as a sum of cosine similarities between their tokens' embeddings. The metric was shown to have high correlation to human judgements in MT and image captioning.

BLEURT score (Sellam et al., 2020) is another reference-based trained metric based on BERT. To obtain the score, BERT is first pre-trained on a diverse set of syntactic reference-candidate pairs and then it is fine-tuned to predict human scores for MT. Originally only trained for English, it was later adapted for other languages (Pu et al., 2021).

NMTScore is a set of similarity measures introduced by Vamvas and Sennrich (2022), which uses the idea of calculating the probability whether one text is a translation of another. To be more precise, they introduce three different scores based on different probabilities. Direct score is based on direct probability of a source text in source language to be translated into a target text in target language. Pivot score refers to the probability of first having one text translated into a pivot language and then from that pivot language to be translated into a target text. Finally, cross-likelihood estimates the probability of a translation into a pre-determined language of a source text to be also a translation of a target text. As the authors use Prism (Thompson and Post, 2020) model, which allows for same language translation, treating it as a paraphrasing task, as the base MT model to calculate those probabilities, all these measures can be applied to the texts in the same language. Vamvas and Sennrich (2022) showed the possibility of using their NMT scores for data-to-text generation.

CometKiwi (Rei et al., 2022) is a trained metric for MT quality estimation. It is built as a combination of a COMET framework (Rei et al., 2020), which was developed for reference-based MT evaluation, and OpenKIWI (Kepler et al., 2019), which was developed for word-level and sentence-level quality-estimation. Shown to have good correlations with human judgements and adapted for multiple languages, it is used in MT evaluation and multilingual reference-based evaluation.

Input-dependent metrics

Metrics in this section use structured input data either as an addition to or instead of references to produce a score for data-to-text generation. We focus on metrics which can be applied to KG-to-text generation.

PARENT (Dhingra et al., 2019) is a surface-based metric introduced for table-to-text generation. It is a an n-gram based metric, where n-gram precision and recall are combined into an F-score. However, each n-gram is counted 'correct' not only if it occurs in the reference, but also on whether it can be entailed by the provided table, which can be estimated with a probability based on two different probabilistic models. One is a word-overlap model (PARENT-W), which checks for different parts of an n-gram whether they occur in the table, and another is a co-occurrence model (PARENT-C), which uses co-occurrence counts from a training set of table-reference pairs. While the authors have shown it can be applied to the WebNLG English dataset, it is not directly applicable when the knowledge graph and the references are in different languages.

There are also various training-based approaches for KG-to-text generation, such as **FactSpotter** which is a trained metric focusing on whether each triple of a graph is represented in the text (Zhang et al., 2023). It does not account for whether the text introduces new information and was only presented for English.

Another semantic-based metric which accounts for the input graph is **EREDAT** (Le Scao and Gardent, 2023), which, drawing inspiration from SBERT, trains a cross-modal model to compute aligned embeddings for graphs and texts to be able to compute a similarity score between a graph and the generated text. However, it was again presented only for English.

Recently there was also introduced a **MuCAL** metric (Song and Gardent, 2025), which, similarly to EREDAT, uses contrastive learning to train a KG/text alignment model. They make the model multilingual, covering in addition to English other languages, namely Arabic, Chinese, French, Russian, and Spanish.

1.4.2 Human Evaluation

As automatic metrics do not capture all the desirable aspects of a generated text and often have low correlation with human judgements, human evaluation became necessary to complement automatic evaluation. one of the most trusted ways to estimate the generation quality. It is commonly used across different NLG tasks as a means to evaluate both quality of the produced text regardless of the specific task, and quality with respect to the task.

There are several parts of human evaluation that can be varied depending on the goals of human evaluation and available resources.

One part is deciding on the criteria and its phrasing itself. Some works decide to create a single question on whether the task was done successfully, for example, (Barrault et al., 2020) ask participants to evaluate the quality of translation via direct assessment comparing source texts or reference texts to the generated translation with a 100-points rating scale. While other works split the quality assessment into several more specific questions. For example, Freitag et al. (2021) follow MQM guidelines (Lommel et al., 2014) and evaluate MT quality asking human participants to identify errors present in each segment by their type and severity. As can be seen in fig. 1.3 they identified 22 different error categories, covering both the errors of MT accuracy and quality of the produced text.

In previous examples the participants are asked to evaluate each generated text or segment on its own, i.e. in an absolute setup. The setup can also be relative, so participants in human evaluation are asked to compare different outputs or systems. Rank-based Magnitude Estimation (Novikova et al., 2018), which helps not only to rank systems based on quality but also estimates the difference in quality, was used in human evaluation for E2E challenge (Dušek et al., 2020). While relative setup can be simpler in terms of criteria phrasing than the absolute one, it, however, makes it difficult to include new systems into comparison, as the whole experiment would need to be rerun. Absolute setup is preferred when dealing with objective criteria to detect and analyse specific drawback of one model or another. Relative setup is often used for comparison especially with more subjective criteria. For a person it is usually simpler, for example, to order sentences in terms of fluency, than to give an exact fluency score.

Depending on a context of the task and goals of evaluation, the evaluation can be done by experts in the domain (Erera et al., 2019; Joshi et al., 2020), crowdsourced

Error Category		Description
Accuracy	Addition	Translation includes information not present in the source.
	Omission	Translation is missing content from the source.
	Mistranslation	Translation does not accurately represent the source.
	Untranslated text	Source text has been left untranslated.
Fluency	Punctuation	Incorrect punctuation (for locale or style).
	Spelling	Incorrect spelling or capitalization.
	Grammar	Problems with grammar, other than orthography.
	Register	Wrong grammatical register (eg, inappropriately informal pronouns).
	Inconsistency	Internal inconsistency (not related to terminology).
	Character encoding	Characters are garbled due to incorrect encoding.
Terminology	Inappropriate for context	Terminology is non-standard or does not fit context.
	Inconsistent use	Terminology is used inconsistently.
Style	Awkward	Translation has stylistic problems.
Locale convention	Address format	Wrong format for addresses.
	Currency format	Wrong format for currency.
	Date format	Wrong format for dates.
	Name format	Wrong format for names.
	Telephone format	Wrong format for telephone numbers.
	Time format	Wrong format for time expressions.
Other		Any other issues.
Source error		An error in the source.
Non-translation		Impossible to reliably characterize distinct errors.

Figure 1.3: Error hierarchy used by Freitag et al. (2021)

(Shardlow and Nawaz, 2019; Castro Ferreira et al., 2020a) or even performed by end users of developed systems (Ghandeharioun et al., 2019). Experts are a common option in domain-specific tasks, like medical domain summarization, which requires strong knowledge of the domain. However, in a general domain setting, experts can introduce their own biases, which is a problem when the system is meant for non-experts (van der Lee et al., 2019). When working with crowdsourcing there are many challenges in terms of participant selection or answers filtering to ensure the quality of evaluation. For example, Karpinska et al. (2021) note the importance of a training phase for crowdsource participants to ensure the validity and meaningfulness of the scores. The end users evaluation is preferable when the goal is to evaluate an interactive system as a whole and is commonly used in dialogue systems.

As any work with humans involve human errors and biases, it is common to have multiple participants in human evaluation studies to evaluate the same set of instances. Those judgements can be aggregated in different ways, such as majority vote, average, or weighted average based on agreement (Raykar et al., 2010). Inter-Annotator Agreement is another important aspect to consider. Low IAA may be a sign of a task subjectivity, poor human evaluation design, or issues with participants selection (Sampson and Babarczy, 2008). In all cases IAA can give an insight on reliability of human evaluation.

Some works opt for inclusion of extra information from human evaluation participants, i.e. they ask to provide rationale behind their answers or provide a long-form description of the problems in the generated text. This extra information can be then analysed either to improve the setup of the human evaluation or to draw non-trivial conclusions. For example, Chaganty et al. (2018) found that asking annotators to highlight regions in the text that support their decision substantially improved the quality of the output.

However, as Howcroft et al. (2020) showed, the variability of human evaluation approaches as well as the lack of details of these approaches in published papers make it

difficult to understand the meaning of these evaluations and compare different generation methods. The reproduction issue of human evaluation experiments is further highlighted by Belz et al. (2023). Thus one of the challenges of the human evaluation is to design the process in a clear and meaningful way. Human evaluation is nevertheless important and brings not only the scores but also insights on various aspects of the models performance. It contributes to the analysis of particular errors made by the models, which could be then directly addressed. Finally, human evaluation is flexible, it does not depend on existing metrics and can be devised for multiple criteria, focusing on the needs of a specific task.

In KG-to-text generation it is common to have task specific criteria for human evaluation which refer to how closely the text follows the information given in the data. For example, in the 2017 WebNLG Challenge human evaluation (Shimorina et al., 2018) this role was played by criteria of 'correctness', for which annotators could give a score on a three-point scale. In 2020 WebNLG challenge (Castro Ferreira et al., 2020a) the quality of data expression was expressed via three different criteria: data coverage (whether all predicates in the data are expressed in the text), relevance (whether all predicates in the text can be found in the data) and correctness (whether for each predicate the subject and object are expressed in a decent way). Finally in 2023 WebNLG Challenge (Cripwell et al., 2023) it was expressed via binary notions of presence of omissions (from the data in the text), additions (in the text compared to data) and unnecessary repetitions. In our work we mostly make use of a system with binary questions on omissions, additions and repetitions as we found them easier to interpret as well as easier to test the potential participants on their understanding of the criteria in the participant selection phase.

1.4.3 LLM-as-a-judge for Evaluation

As human evaluation can be time-consuming, costly, and labour-demanding, while automatic metrics often lack the ability to evaluate specific aspects of quality, with the development of LLMs and their ability to follow instructions, many researchers started to use LLMs to conduct quality assessments.

LLM-as-a-judge has been used to directly give a score based on the description of a criteria (Liu et al., 2023; Fu et al., 2024), rank candidate responses according to given criteria (Li et al., 2025), produce span annotations (Kasner et al., 2025) or even to give a free-form feedback (Kartac et al., 2025).

However, LLM based evaluation suffers from the same problems as all LLM-based methods. For example, they tend to have a position bias (Wang et al., 2024b), favouring the text in a certain position, and are also prone to prompt sensitivity, as well as text familiarity bias (Stureborg et al., 2024). It was also shown that LLMs are biased towards self-generated texts (Xu et al., 2024). Bavaresco et al. (2025) conducted a large study on LLM-as-a-judge approach, covering different prompt strategies, models, and tasks, and they showed that while LLMs can be used reliably as evaluators on some tasks, generally their agreement with human evaluation is not sufficient. They highlight the need for validation and calibration of LLMs against human judgements.

LLMs can be calibrated to improve their judgement quality by targeting specific biases during fine-tuning (Zhu et al., 2025), for example, by augmenting data with examples where the instances to judge are swapped to alleviate position bias.

For the task of verbalisation of RDF graphs, which we explore in this thesis, the criteria are reasonably clear, and thus we believe that common automatic metrics and human evaluation suffice to give a clear picture of the results.

1.5 Conclusion

In this chapter, we have provided a general overview of the NLG domain, focusing on KG-to-text generation, covering various applications, datasets and approaches. We have also discussed different evaluation methods as well as challenges that occur during evaluation when dealing with multilingual KG-to-text generation.

In our work, we use the WebNLG and KELM datasets as source of data for both training and testing. We experiment with various neural models, based on T5 or BART. We later use prompting and RAG to elevate LLMs for multilingual KG-to-Text generation.

For evaluation we focus on input-dependent automatic metrics like BLEU, TER, ChrF++ and BERT Score which work for all our languages but rely on presence of references in the target language. We also use several metrics derived from MT, like NMTScore and CometKiwi when we move on to comparing different approaches in a multilingual setting, as they are semantic based and configured well for multilinguality.

Finally, we ground our human evaluation in the criteria introduced in (Cripwell et al., 2023), focusing on additions, omissions, and repetitions. However, we foreground other errors when performing an error analysis.

2

Evaluating Knowledge Graph-to-text Generation Models for English and Russian on Out Of Domain Data

Contents

2.1 Introduction 26
2.2 Creating New Benchmarks for English and Russian 27
2.3 Creating Kelm Benchmarks 28
2.4 Creating WebNLG Benchmarks for Russian 32
2.5 Assessing Generalisation 33
2.6 Quantitative Analysis 33
2.6.1 Models 33
2.6.2 Metrics 34
2.6.3 Results 34
2.7 Qualitative Analysis 36
2.7.1 Error Annotation 36
2.7.2 Error Analysis 37
2.8 Conclusion 38

While the WebNLG dataset has prompted much research on generation from knowledge graphs, little work has examined how well models trained on the WebNLG data generalise to unseen data and work has mostly been focused on English. In this chapter, we introduce novel benchmarks for both English and Russian which contain various ratios of unseen entities and properties. These benchmarks also differ from WebNLG in that some of the graphs stem from Wikidata rather than DBpedia. Evaluating various models for English and Russian on these benchmarks shows a strong decrease in performance while a qualitative analysis highlights the various types of errors induced by non i.i.d data.

2.1 Introduction

Much research on KG-to-Text generation focuses on the WebNLG dataset (Gardent et al., 2017a) often restricting evaluation to the WebNLG test sets. While these include both seen (in distribution) and unseen (out of distribution, OOD) data for English, no unseen test data is available for Russian. Furthermore, the input graphs all stem from DBpedia and the texts are often stilted as they are either crowd-sourced (English data) or machine translated from the crowdsourced texts and manually verified (Russian data).

To assess how well current NLG models perform on OOD KG-to-Text generation, we create several novel benchmarks for both English and Russian which address these shortcomings and differ from the WebNLG test sets in several ways. First, they include both English and Russian – WebNLG only has unseen test data for English. Second, they include both DBpedia and Wikidata[5] graphs – WebNLG focuses on DBpedia graphs. Third, they contain various ratios of unseen entities and properties – this allows for a detailed analysis of how the type and ratio of unseen data impact performance.

Using these benchmarks, we then assess and compare several KG-to-Text models. The results show a strong decrease in performance for all models compared to results on in domain data. A qualitative analysis highlights the various types of errors induced by OOD data suggesting directions for further research on KG-to-Text.

KG-to-Text Generation. The WebNLG challenges gave rise to different approaches for KG-to-Text generation, with most of the 2020 participating models being fine-tuned version of T5 (Raffel et al., 2020) or BART (Lewis et al., 2020). In the WebNLG 2020 challenge (Castro Ferreira et al., 2020a), human evaluation showed that models which were based on these pre-trained encoder-decoders produce the best texts in terms of fluency (e.g., Yang et al. (2020); Agarwal et al. (2020)) but lacked adequacy on unseen test sets exposing a noticeable drop in performance regarding Relevance (not all information mentioned in the text is present in the input graph) and Data Coverage (not all information present in the input graph is verbalised by the text).

For Russian, the two best performing models are Kazakov et al. (2023) and Kumar et al. (2023). Both models fine-tune a pre-trained model on the WebNLG data with Kazakov et al. (2023) fine-tuning the pre-trained FRED (Full-scale Russian Enhanced Denoiser, 1.7M Parameter) model and Kumar et al. (2023) mT5_{base}. Neither of these models were evaluated on unseen data.

Evaluation. Recent work has focused on creating better evaluation benchmarks for data-to-text generation. In particular, Mille et al. (2021) introduced various subtests (subpopulations) for different data-to-text generation tasks including WebNLG. They developed subpopulations based on input size and the uniqueness of subjects, objects, and properties present in the data. Their study showed that each of these properties influences the results and that the level of impact differs between Russian and English. Similarly, in 2024, a new GEM challenge on Data-to-Text generation was launched which includes par-

[5] https://www.wikidata.org/wiki/Wikidata:Main_Page

allel datasets to WebNLG featuring counterfactual and fictional data.[6] This challenge also evaluates data-to-text generation models on graphs from Wikidata (Axelsson and Skantze, 2023). These new test sets consist solely of automatically combined graphs without any reference verbalizations, which excludes reference-based evaluation and necessitate human evaluation.

Different from these works, we provide new unseen test sets for KG-to-Text generation which include references in both English and Russian. We then used these test sets to evaluate the ability of existing models to generalise to OOD data and to analyse the types of errors that arise in their output texts.

2.2 Creating New Benchmarks for English and Russian

We aim to create benchmarks which support a fine-grained assessment of how various types of unseen items impact generation.

Terminology. An *unseen element* is a KG element (entity or property) not seen in the WebNLG training/dev data. An *unseen category* is a DBpedia category which is not part of the 16 categories[7] used in WebNLG to create the training data.

We create separate benchmarks depending on whether the input graph contains unseen entities, unseen entities and properties or unseen category. While the latter two benchmarks permit assessing how well models perform on out of domain data, the former helps evaluating how much performance degrades with varying ratios of unseen entities.

For English and Russian, we derive these benchmarks from the KELM dataset (Agarwal et al., 2021), a large dataset of (graph,text) pairs created using distant supervision. For Russian, we additionally derive benchmarks from the WebNLG data following a methodology similar to that used to create the WebNLG unseen test set for English.

KELM. Agarwal et al. (2021) created the KELM dataset in several steps as follows. First, Wikidata triples were heuristically aligned to Wikipedia sentences yielding a dataset of approximately 6M noisily aligned (graph, sentence) pairs and covering 1,041 Wikidata properties. Second, 15M Wikidata graphs where created based on relation co-occurrence counts and the corresponding text was generated from these graphs using a T5 model fine-tuned on the silver 6M (graph,sentence) pairs. The semantic adequacy (semantic match between graph and text) and the fluency of 200 randomly selected KELM (graph,text) pairs were annotated by human judges (8 annotators, 2 judgements per instance) on a 1-5 scale, yielding an average rate of 4.36 for semantic adequacy and 4.60 for fluency. Examples of KELM instances are shown in table 2.1.

WebNLG. The WebNLG dataset is a dataset of (graph,text) pairs where graphs were extracted from DBpedia and texts were crowdsourced to match the input graph. For

[6] https://gem-benchmark.com/shared_task

[7] The 16 categories used to anchor WebNLG data are: Airport, Astronaut, Building, City, ComicsCharacter, Food, Monument, SportsTeam, University, WrittenWork, Athlete, Artist, CelestialBody, MeanOfTransportation, Politician, Company.

Text	Graph
The redshift of NGC 266 is 0.015537.	(*NGC 266* , *redshift*, *0.015537*)
Bowditch is a lunar crater which is located at LQ22 on the Moon and named after Nathaniel Bowditch.	(*Bowditch_crater*, *located on astronomical location*, *Moon*), (*Bowditch_crater*, *instance of*, *Lunar craters*), (*Bowditch_crater*, *location*, *LQ22*), (*Bowditch_crater*, *named after*, *Nathaniel Bowditch*)

Table 2.1: Examples from KELM dataset

English, the training data covers 16 DBpedia categories and the test set has three subsets: Seen (490 instances), a test set where graphs include only entities and properties present in the training data; Unseen Entities (393 instances), where graphs include entities not present in the training data; and Unseen Categories (896 instances), a test set where graphs are rooted in entities whose category does not belong to the 16 categories present in the training data.[8] For Russian, the training data only covers nine categories[9] and all instances in the test set (1,200 instances) are from the seen categories.

2.3 Creating Kelm Benchmarks

To create the KELM unseen test sets (KELM-E, KELM-E+P), we first select subsets of KELM that contain unseen entities and properties. We then ask human annotators to verify the semantic adequacy of the (graph, text) pairs (does the text match the graph?) filtering out all pairs which are not validated by the annotators. This yields novel unseen test sets for English. We create corresponding test sets for Russian using machine translation and manual correction by professional translators. An alternative would be to create a Russian dataset from Wikidata and Wikipedia using Agarwal et al. (2022) methodology. We adopted the MT approach instead because it is less computationally intensive and it allows for the creation of a parallel (graph, English text, Russian text) dataset.

In what follows, KELM refers to the dataset created by Agarwal et al. (2022) while KELM-E, KELM-E+P refers to the two benchmarks we derived from KELM.

Selecting a Subset of KELM. We extract a subset of KELM such that (i) graph and text embeddings have high similarity, (ii) the dataset is balanced across graph size and (iii) the distribution of the Wikidata properties present in the KELM dataset is preserved.

[8]For each test set there are two versions, one for generation and the other for semantic parsing. Here we only consider the generation test sets.

[9]These nine DBpedia categories are: Airport, Astronaut, Building, CelestialBody, ComicsCharacter, Food, Monument, SportsTeam, and University.

The latter point helps in ensuring that our dataset has a wide variety of topics and is not skewed towards frequent properties.

To extract this subset, we proceed as follows. First, we compute graph and text embeddings using Le Scao and Gardent (2023) cross-modal KG-Text model and we only keep those pairs whose graph and text embeddings have a cosine similarity greater than 0.9. We then remove quadruples (i.e., Wikidata facts that are not triples) and graphs that have more than six triples as these are a minority (less than 1%) and tend to have repetitive or unintelligible texts. Creating a dataset for larger graphs is possible but would require developing an alternative content selection procedure to ensure that the selected subgraphs yield text that are coherent and readable. We further compute the ratio of unseen elements for each graph text pairs. Finally, we select two types of unseen data: instances where all properties are known but some entities are not (unseen entities, KELM-E) and instances which contain various ratio of unseen entities and properties (unseen entities and properties, KELM-E+P).

Human Validation on English Data. A manual inspection of 100 random instances shows that approximately one third of the data is poorly aligned, i.e. text and graph convey different content. We use crowd sourcing to filter out badly aligned (graph,text) pairs. We use the Potato annotation tool (Pei et al., 2022) to create a website for annotation and Prolific[10] to find participants for the study. We provide a screenshot of the built website in fig. 2.1, where you can see an example to annotate with the questions presented to the workers. The participants were paid 14€ for annotating 100 instances and 2€ for the qualification task (even if failed) which averages to 10.5€ per hour. Further details about the human annotation protocol are given in Section A.1.

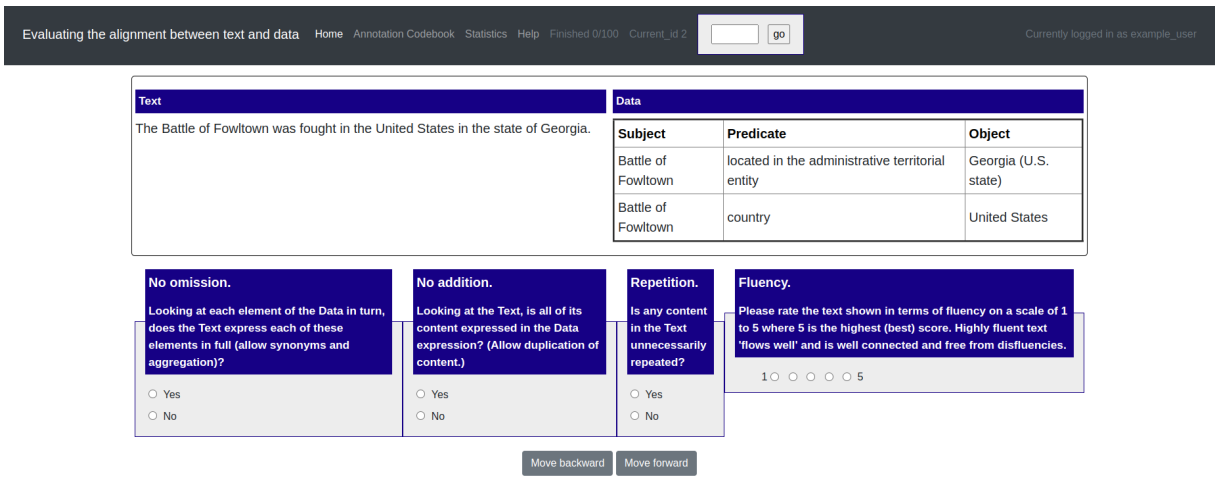


Figure 2.1: A screenshot from the annotation website we built. Evaluation of a text-data pair.

To evaluate the quality of each pair, we used the WebNLG Challenge 2023 criteria for human evaluation (Cripwell et al., 2023) whereby for each item, the annotators were

[10] https://prolific.com/

asked to answer the following four questions (with binary yes/no answer for the first three questions).

No omission. "Looking at each element of the graph in turn, does the text express each of these elements in full (allow synonyms and aggregation)?".

No addition. "Looking at the text, is all of its content expressed in the graph? (Allow duplication of content.)".

No unnecessary repetition. "Is any content in the text unnecessarily repeated?".

Fluency. " Please rate the text shown in terms of fluency on a scale of 1 to 5 where 5 is the highest (best) score. Highly fluent text 'flows well' and is well connected and free from disfluencies.".

To ensure a good understanding of these criteria, we made available an annotation codebook with explanations and examples for each criterion. We also run a prestudy consisting of 15 (graph,text) pairs where 10 examples were taken from KELM and 5 easier examples were created manually. We made sure that the examples covered all possible answers for each yes/no criteria and were of different level of fluency. However, as it is hard to evaluate fluency, we only verified if the participant answered correctly to all yes/no criteria. To pass this prestudy a participant must have annotated 10 out of 15 examples correctly. Only around 10% of participants managed to pass the prestudy and the data was annotated by 14 annotators. Table 2.2 shows the number of instances created for each category of unseen data before and after human validation. One may notice the imbalance of the graph sizes for KELM-E. This is a consequence of a condition that all properties should be seen in WebNLG training/dev data. The more triples there are, the more properties there are in a graph and thus the less the possibility that all of them are seen. The results are consistent with our preliminary analysis with about 2/3 of the automatically extracted data being deemed correct by the annotators.

	E		E+P	
	B	A	B	A
# instances	4,167	2,126	3,800	1,312
# entities	7,801	4,038	11,264	4,078
# properties	57	53	394	296
# 1-triple G	3,725	1,917	374	176
# 2-triple G	334	172	326	127
# 3-triple G	53	27	647	295
# 4-triple G	9	1	755	256
# 5-triple G	3	0	782	240
# 6-triple G	43	9	916	218

Table 2.2: **KELM Extracted Subsets for English and Russian** Before (B) and After (A) human validation (E: graphs with unknown entities, E+P: graphs with unknown entities and properties).

Creating the Russian Benchmark. We create KELM-based benchmarks for Russian by automatically translating the texts of the English KELM benchmarks and manually verifying the resulting translations. For Machine Translation, we use the NLLB neural Machine Translation model (NLLB Team et al., 2022). For human validation, we hired four professional translators. As entities were shown to raise translation issues (Shimorina et al., 2019), we collected the Russian names of graph entities by querying DBpedia for their Russian label using the property 'rdfs:label' and provided the translators with (i) the English text from KELM, (ii) its translation into Russian and (iii) the Russian translation of the KG entities present in the input graph. Translators could copy and paste the NLLB translation and modify it afterwards. The translators also had the possibility to mention any kind of mistakes they notice.

Table 2.3 shows statistics on the changes introduced by the translators to convert the machine translated texts into valid Russian. To measure the differences between the two texts, we use the Levenshtein ratio. [11] We see a low similarity ratio indicating that, for Russian, machine translated texts need correcting.

	KELM	WebNLG
Translator	**Mean (STD)**	**Mean (STD)**
1	0.29 (0.15)	0.28 (0.16)
2	0.33 (0.15)	0.38 (0.16)
3	0.26 (0.15)	0.24 (0.18)
4	0.24 (0.15)	0.25 (0.16)
5		0.34 (0.16)
Total	0.28 (0.15)	0.30 (0.17)

Table 2.3: **Modification statistics between MT translations and final human translations for KELM and WebNLG test sets**. Levenshtein ratio distance mean and STD values for each translator separately and together.

Out of the 230 comments left by the translators, 214 concerned minor issues such as texts including + in front of positive numbers (the way they appear in the data). In two cases, the graph did not match a meaning of the text and we removed either the whole instance or a triple from the graph. Finally, there were 14 instances where we modified both the English and the Russian sentences as these contained mistakes regarding the gender of a person (like a scientist was described as a man by default) or the lexicalisation of field specific terms (like 'taxon' in Biology).

[11] The Levenshtein distance indicates the minimum number of insertion, deletion or substitution of individual characters that are required to transform one sentence into another and the Levenshtein ratio normalises this distance by the length of the two sentences and inverts the score so that a perfect match will have a score of 1.0, and completely dissimilar strings will be assigned a value of 0.0 (LDistance: Levenshtein Distance, LRatio: Levenshtein Ratio): $LRatio(a,b) = 1 - \frac{LDistance(a,b)}{len(a)+len(b)}$

2.4 Creating WebNLG Benchmarks for Russian

We derive two WebNLG Russian benchmarks from the WebNLG English test set by first selecting graphs with unseen categories or unseen entities and then translating the corresponding texts into Russian.

As explained above, English WebNLG differs from Russian WebNLG in that it covers 16 categories (vs. 9 for Russian) and the test set includes an Unseen Category and an Unseen Entities test set. To create an Unseen Category test set for Russian (WebNLG-C, 1,251 instances), we simply select from the English test set all instances which belong to the 7 categories not included in Russian WebNLG training and dev data. The second test set (WebNLG-E, 192 instances) consists of the instances that are from seen categories in Russian WebNLG train or dev set, but the entities are unseen.

These two subsets were then translated from English to Russian by 5 professional translators, who have Russian as a native language. Similarly to the validation of the KELM translations, the translators were provided with the English text, the NLLB translation and the DBpedia Russian labels of the graph entities and again we observe a high ratio of changes introduced by the translators (Table 2.3).

Comparing the English texts to the corresponding graphs, the translators spotted a few errors (165 instances were highlighted out of the whole test set). Those errors include references to female scientists or politicians by he/him, subject and object interchanged in the text comparing to the KG data.

After cleaning the data after translators, we carefully transferred data into the WebNLG xml format.

Table 2.4 summarises the created benchmarks indicating the number of test instances for each language and for each type of unseen data.

Benchmark	**Nb. of Instances**	
	Russian	**English**
KELM		
KELM-E+P		
50/60	146	146
60/70	211	211
70/80	328	328
80/90	265	265
90/100	361	361
Total	1311	1311
KELM-E	2126	2126
WebNLG		
WebNLG-C	1251	N/A
WebNLG-E	192	N/A

Table 2.4: **KELM and WebNLG Unseen Benchmarks.** Number of (graph,text) pairs in each test set (E: Entities, E+P: Unknown Entities and Properties, X/Y: the min and max ratio of unknown elements, C: Unknown Category)

2.5 Assessing Generalisation

We evaluate current pre-trained Encoder-Decoders on our benchmarks. Since the best approaches in the 2020 edition of the WebNLG shared task were based on T5 or mT5 (Yang et al., 2020; Castro Ferreira et al., 2020a), we consider various versions of this model fine-tuned on the WebNLG English/Russian training data. We also include in our evaluation the Control-Prefixes (Clive et al., 2022) model, a state-of-the-art model for KG-to-Text generation as well as the models for Russian submitted to the WebNLG 2023 Challenge (Cripwell et al., 2023). We evaluate the models using automatic metrics and run a qualitative analysis to identify the most common errors occurring when assessing current models on out of domain data.

2.6 Quantitative Analysis

2.6.1 Models

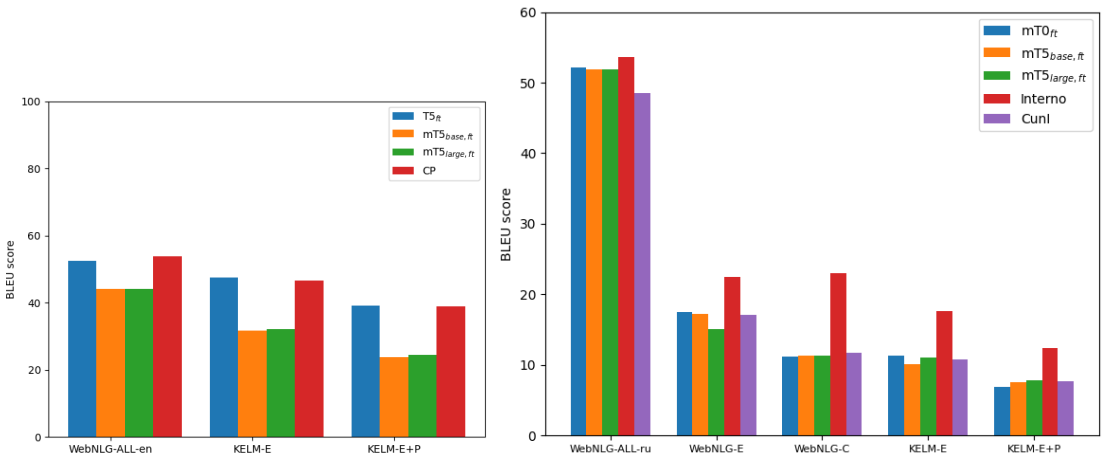


Figure 2.2: BLEU scores for each model on English (Left) and Russian (Right) Test Sets.

English. We evaluate four models on the English benchmarks: the $T5_{base}$ model fine-tuned on the WebNLG 2020 training data for English ($T5_{ft}$); the $mT5_{base}$ and $mT5_{large}$ models fine-tuned on the WebNLG 2020 training data for English and Russian ($mT5_{base,ft}$, $mT5_{large,ft}$); and CP, a state of the art model for KG-to-Text generation (Clive et al., 2022)[12] which uses tasks-specific soft prompts (Control Prefixes, CP). We train this model for 40 epochs on WebNLG 2020 English training data with all the parameters provided by the authors and using their code.[13] When running the finetuned model on new KELM test sets, we pass categories (which are used as part of the prefix) all equalled to 1.

[12] https://paperswithcode.com/sota/data-to-text-generation-on-webnlg?p=control-prefixes-for-text-generation

[13] https://github.com/jordiclive/ControlPrefixes

Russian. We also evaluate $\text{mT5}_{base,ft}$ and $\text{mT5}_{large,ft}$ fine-tuned on WebNLG Russian training data on the Russian benchmarks. In addition, we evaluate the mT0 pre-trained model (mT5 fine-tuned on crosslingual tasks, (Muennighoff et al., 2023)) fine-tuned on the WebNLG training data for Russian (mT0_{ft}) and two models for Russian that participated in the WebNLG 2023 challenge. The first model is Interno, a model based on FRED-T5 (Full-scale Russian Enhanced Denoiser, 1.7M Parameters, (Zmitrovich et al., 2024)) and fine-tuned on WebNLG training data (Kazakov et al., 2023). We used the final checkpoints submitted to the WebNLG 2023 challenge. The second model is CunI, a mT5_{base} model which was fine-tuned on multilingual data created by machine translating (using NLLB) WebNLG training data into Maltese, Irish Gaelic and Welsh and including the original Russian data (Kumar et al., 2023).[14]

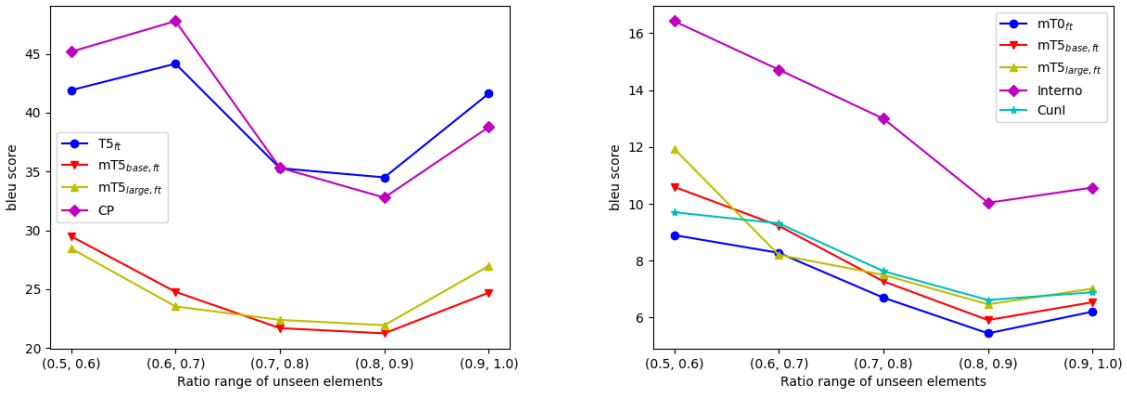


Figure 2.3: BLEU score for the different ratios of unseen elements (properties or entities) on English (Left) and Russian (Right)

2.6.2 Metrics

All models were evaluated using the WebNLG-toolkit[15] which includes the SacreBLEU implementation for BLEU (Papineni et al., 2002b), the pyter implementation for TER (Snover et al., 2006), and the official implementations of chrF++ (Popović, 2017) and BERTScore (Zhang* et al., 2020).

2.6.3 Results

Figure 2.2 shows the BLEU scores for each model on each of the benchmarks. The results for the other metrics show similar trends, so they are not discussed in the main body of the chapter but can be found in the Appendix A.2.

[14]Unfortunately, we did not manage to reproduce the original results using the authors code(https://github.com/knalin55/CUNI_Wue-WebNLG23_Submission) and communicating with them. Possible difference: did not use the fp16 while it seems the authors used it (gpu available did not support it).

[15]https://github.com/WebNLG/webnlg_toolkit/

Strong Degradation on the new Benchmarks. For all models and for both languages, we observe a strong degradation on our benchmarks with a drop in BLEU score with respect to the initial WebNLG test sets ranging from 5 to 20 BLEU points for English and 31 to 45 points for Russian. On English, the models that degrade least are the state-of-the art CP model and the monolingual T5 model fine-tuned on WebNLG. We observe a similar trend on Russian, where the degradation for the four multilingual models (mT0$_{ft}$, mT5$_{large,ft}$, mT5$_{base,ft}$, CunI) is worse than for Interno, a model based on FRED-T5 (Full-scale Russian Enhanced Denoiser), a monolingual model pre-trained on Russian. This suggests that multilingual models are more sensitive to out of distribution data than monolingual ones.

Stronger Degradation on OOD Graphs. Comparing results on KELM and the WebNLG benchmarks (KELM-E/WebNLG-E and KELM-E+P/WebNLG-C), we find a stronger degradation on KELM benchmarks indicating that, even though there is a large overlap between DBpedia and Wikidata properties and entities, models trained on DBpedia graphs and crowdsourced text do not generalise well to Wikidata graphs.

Stronger Degradation when both Properties and Entities are unseen. Unsurprisingly, we see that results are lower for graphs that contains both unseen properties and unseen entities (KELM-E+P, WebNLG-C) than only unseen entities (KELM-E, WebNLG-E).

Impact of the ratio of unseen elements. Figure 2.3 shows that performance mostly decreases as the ratio of unseen elements increases. However, there is a surprising peak at the 0.9/1.0 ratio. We conjecture that this is due to the high proportion of small graphs for this ratio (48% of these graphs are of size 1, as can be seen in table 2.5), which makes the generation task easier.

Ratio \ Graph size	1	2	3	4	5	6
(0.5, 0.6)	0	0	86	33	21	6
(0.6, 0.7)	1	47	91	29	22	21
(0.7, 0.8)	0	0	42	82	109	95
(0.8, 0.9)	0	42	42	49	49	83
(0.9, 1.0)	175	38	34	64	37	13

Table 2.5: Graph sizes distribution in each ratio range in KELM-E+P

We also see that, while for lower ratios of unseen elements, the mT5 base model (mT5$_{base,ft}$) outperforms the large one (mT5$_{large,ft}$), the inverse is true for ratios greater than 70%. This suggests that smaller models overfit to the data. As the ratio of unseen elements is low, performance does not decrease too much as the remaining seen elements have been memorised by the model and can be generated correctly. Conversely, when the ratio is high, the advantage gained through memorisation of seen elements is reduced and performance decreases compared to larger models.

2.7 Qualitative Analysis

To get a better understanding of the type of errors made by generation models on OOD data, we run a qualitative analysis on the models outputs.

2.7.1 Error Annotation

For each model and each benchmark, we select the five instances with the lowest BLEU scores. This yields a total of 320 instances, 200 for Russian (8 benchmarks × 5 models × 5 instances) and 120 for English (6 benchmarks × 4 models × 5 instances). We then manually annotated the selected data for different types of errors including three error types previously used in the evaluation of KG-to-Text models (Belz et al., 2023) and six additional error types we found in the data. Specifically, we identified the following 9 types of errors.

Addition (A). The text contains information that is not present in the input graph.

Omission (O). The text misses information present in the input graph.

Repetition (R). The text has unnecessarily repeated parts.

Entity distortion (ED). An entity is mentioned in the generated text, but its name is partially incorrect. This can manifest in different ways for Russian and English. For Russian it includes entities copied over from the input data, entities mixing different scripts or just mistranslated. For English it mostly includes misspellings and incorrect numbers.

Property understanding (PU). The property is verbalised incorrectly (e.g., "instance of" is verbalised as *"is a part of"*).

Topic change. (TC) The text treats a property and its arguments as if they were from another topic, for instance referring to buildings as if they were people and using expressions like *"was born on"* instead of *"was built in"*. This category differs from the "Property Understanding" category in that the lexicalisation of the property is correct out of context but incorrect for the given triple i.e., when taking its arguments into account.

Complex text (CT). The generated text is unnecessarily complex. This includes cases where each triple is verbalised but natural means of aggregation (ellipsis, coordination, pronouns) are not exploited resulting in unnatural text. E.g., *"Peter Slater (ornithologist) is a human and speaks, writes or signs in English. His given name is Peter."* rather than *"Peter Slater is an English speaking ornithologist."*). This error category also includes other over complications such as using *"is an instance of"* instead of directly saying *"is"*. This category is only assigned to cases that have neither additions nor omissions.

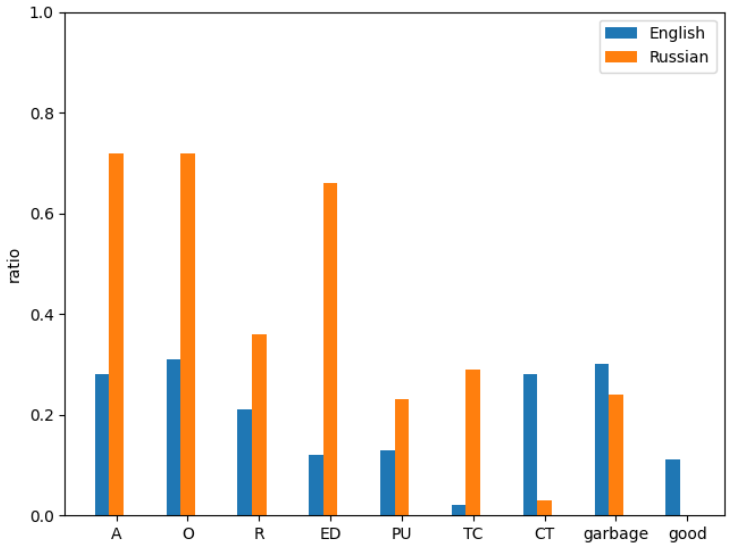


Figure 2.4: Error ratios per language

Garbage (G). Instances which consisted of just unrelated symbols or words which do not form any meaningful statements. If an instance is annotated as *Garbage*, no other annotation is assigned to it.

Good. Instances which in fact were good verbalisations of the input but received a low BLEU score because they paraphrased the reference text.

It is worth noting that one instance can contribute to several error annotations. E.g. *Property Understanding* often leads to one of the triples being not verbalised, and in this case we would also annotate the instance to have an *Omission*. In table 2.6 we provide examples of generated text to illustrate different types of errors for English. In the appendix we also provide examples for Russian (tables A.3 and A.4).

2.7.2 Error Analysis

We report error ratios per language, per model and per benchmark in the appendix (sec. A.4. In this section we provide our observations and analyse the trends in errors.

The error rate is markedly higher for Russian. Figure 2.4 shows a higher error ratio on Russian than on English overall highlighting a high level of degradation when the BLEU score is lowest. The high ratios for almost all error types indicate that the output texts contain multiple errors.On English, the error ratio is lower overall indicating a lower level of degradation even for output with lowest BLEU. In fact, some of the English texts although they have low BLEU scores, are good quality but different from the reference (category *Good*).

Domain change increases Topic Change errors. Interestingly, Figure 2.5 shows that topic change errors are more frequent on OOD data (KELM-E+P, WebNLG-C)

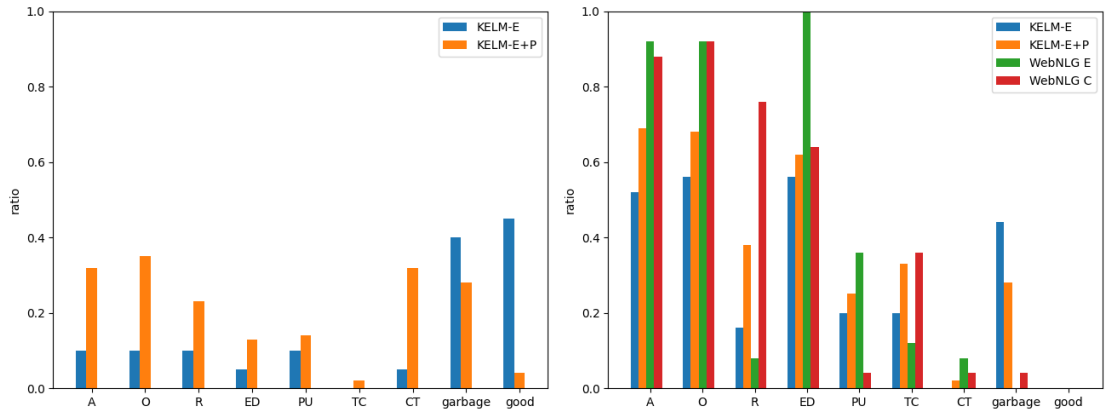


Figure 2.5: Error ratio per test set. English (Left) vs Russian (Right).

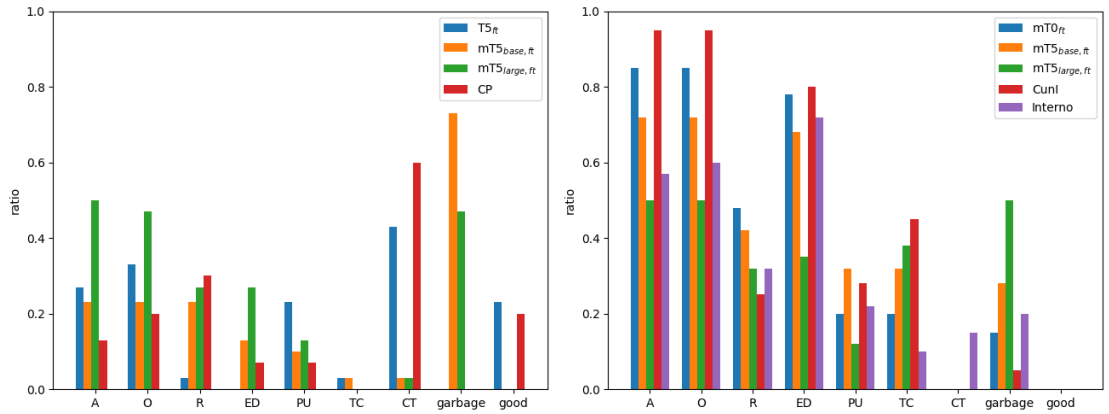


Figure 2.6: Error ratio per model. English (Left) and Russian (Right).

highlighting the fact that neural models fails to adapt property verbalisation to the domain of discourse.

Custom Models show less errors overall. Figure 2.6 shows that for both Russian (Interno model) and English (CP model), custom models yield fewer errors overall than mT0 and mT5 fine-tuned on the WebNLG data.

2.8 Conclusion

We created challenging benchmarks for KG-to-Text generation into English and Russian, quantitatively demonstrated the effects of applying models trained on one distribution (e.g., WebNLG data) to a new distribution (e.g., unseen entities and/or properties) and identified eight error types which arise in this setting.

Contributing to RQ1, we find that Russian models especially struggle with entities verbalisation on out-of-distribution data, which is why in the next chapter we are going to directly address this issue.

Graph	Generated text
Addition	
(*Drenova (Prijepolje)*, *instance of*, *Human settlement*), (*Drenova (Prijepolje)*, *country*, *Serbia*), (*Drenova (Prijepolje)*, *located in area*, *Prijepolje*)	The city of Prijepoli in Serbia is located in the city of Prijepoli. It is a settlement located in the area of Prijepoli.
Omission	
(*Rybakowo, Warmian-Masurian Voivodeship*, *instance of*, *village of Poland*), (*Rybakowo, Warmian-Masurian Voivodeship*, *country*, *Poland*), (*Rybakowo, Warmian-Masurian Voivodeship*, *located in area*, *Gmina Srokowo*)	Rybakowo, Warmian-Masurian Voivodeship is located in Gmina Srokowo, Poland.
Repetition	
(*Travis Heights, Austin, Texas*, *instance of*, *Neighbourhood*), (*Travis Heights, Austin, Texas*, *country*, *United States*), (*Travis Heights, Austin, Texas*, *located in area*, *Austin, Texas*)	Travis Heights, Austin, Texas is a neighborhood in the U.S.A. and is located in Austin, Texas.
Entity distortion	
(*Sandy Lloyd*, *occupation*, *Umpire (cricket)*), (*Sandy Lloyd*, *sport*, *Cricket*), (*Sandy Lloyd*, *occupation*, *Cricketer*), (*Sandy Lloyd*, *sex*, *Male gender*), (*Sandy Lloyd*, *given name*, *Sandy (given name)*), (*Sandy Lloyd*, *family name*, *Lloyd (surname)*)	Sandy Elliot is a Male who was given the nickname 'Sandy'...
Property understanding	
(*Elliott, Illinois*, *instance of*, *Village (United States)*), (*Elliott, Illinois*, *country*, *United States*), (*Elliott, Illinois*, *located in area*, *Illinois*)	Elliott is located in Illinois in the United States. It is part of the Village in the United States.
Topic change	
(*James Baines (clipper)*, *instance of*, *Ship*), (*James Baines (clipper)*, *country*, *United Kingdom*), (*James Baines (clipper)*, *location of final assembly*, *Boston*)	The clipper James Baines, who was a member of the shipyard of the United Kingdom, was assembled in Boston.
Complex text	
(*Palaistra, Florina*, *instance of*, *Human settlement*), (*Palaistra, Florina*, *country*, *Greece*), (*Palaistra, Florina*, *located in area*, *Florina Municipality*)	Palaistra, Florina is located in Florina Municipality, Greece and is an example of human settlement.
Garbage	
(*Henuttawy (princess)*, *sibling*, *Menkheperre*), (*Henuttawy (princess)*, *family*, *Twenty-first Dynasty of Egypt*), (*Henuttawy (princess)*, *sibling*, *Masaharta*)	The twelid of the twelid of the twelid of the twelid of the twelid of the twelid of the twelid of the twelid...
Good	
(*Brian Mallette*, *weight*, *+84 kilogram*)	Brian Mallette's weight is +84 kilogram.

Table 2.6: Error annotation examples for English

3

Fine-Tuning, Prompting, RAG: How do Knowledge Graph-to-Russian Text Generation Models generalise to Out-of-Distribution Data?

Contents

3.1 Introduction 42
3.2 Data 43
3.2.1 Test Data 43
3.2.2 Fine-Tuning and Few-Shot Data 44
3.3 Fine-tuning on Labeled data. 44
3.4 LLM Prompting 45
3.4.1 Input 46
3.4.2 Generation strategies 47
3.4.3 Few-shot selection strategies 47
3.4.4 Results 50
3.5 Retrieval Augmented Generation for LLM 51
3.5.1 Paragraph Retrieval 51
3.5.2 Chunks 51
3.5.3 Queries 53
3.5.4 Few-shots 54
3.5.5 Results 54
3.6 Comparing the Three Approaches 55
3.6.1 Automatic Evaluation 55
3.6.2 Human Evaluation 56
3.6.3 Comparing the Three Approaches: Case Study 58

3.7 Conclusion 59

In this chapter, we focus on Russian as a target language and assess how various generation methods perform on out-of-distribution, unseen data, trying to address errors found in the previous chapter. We especially try to address errors relating to entities and properties verbalisation, as previous work has shown that enriching the input with target language verbalisations of the entities and properties present in the input graph significantly improves the performance of fine-tuned models for Russian. Considering both LLM prompting and Retrieval Augmented Generation (RAG) and comparing multiple variants of each of these methods, we explore alternative ways of integrating such external knowledge into the generation process. Using both automatic metrics and human evaluation, we find that, on unseen data, the fine-tuned model consistently underperforms, indicating limited generalisation capacity; that while it outperforms RAG by a small margin on most datasets, prompting generates less fluent text; and, conversely, that RAG generates text that is less faithful to the input. Overall, LLM-prompting and RAG outperform Fine-Tuning across all unseen testsets.

3.1 Introduction

In this chapter, we investigate the generalisation capacity of three main approaches to generation, namely fine-tuning, LLM prompting and Retrieval Augmented Generation, taking Russian as a target language and evaluating on out-of-distribution (OOD) datasets.

Compared to generation into English, generating from knowledge graphs into Russian faces the challenge of decoding into a language with a different script, a rich verb morphology and a complex nominal declension system. Prior work shows that lexicalising entities into Russian is a difficult task (Chuklin et al., 2022) and as the previous chapter showed, pre-trained models fine-tuned on the Russian WebNLG dataset show a strong performance decrease on non iid data.

Starting with the state-of-the-art KG-to-Russian models presented by Kazakov et al. (2023), we begin by showing that a pre-trained model fine-tuned on the WebNLG data under-performs on both seen and unseen data but that the same model achieves much better results when the input graph is enriched with external knowledge extracted from Wikidata namely, Russian labels (lexicalisations) for the entities and properties present in the input graph.

Next, we explore various LLM-based, Few-Shot Prompting approaches (Brown et al., 2020c). Generating with an LLM benefits from the parametric knowledge acquired from their training on large amounts of data. Compared to Kazakov et al. (2023)'s model which is a fine-tuned version of T5 (Raffel et al., 2020), and therefore trained on much smaller quantities of data than LLMs, the knowledge encapsulated in an LLM might help generalise to new test data. As in the fine-tuning approach just described, we enrich the input prompt with Russian lexicalisations (Wikidata labels) of the entities and properties present in the input graph, and we experiment with various ways of selecting few-shots. Intuitively, the selected few shots should help provide two types of information: (i) how the properties and entities present in the input graph are verbalised into Russian and (ii) how a knowledge graph is verbalised into a text. Accessing these two sources of knowledge

can be viewed as learning both entity/property verbalisation and a generation template. Sample templates can be identified from the associated text and property verbalisation by jointly accessing graph properties and their textual mention. We introduce a few-shot selection method which systematically improves generation results by selecting few shots whose graph size and properties resemble the size and properties present in the input graph.

Retrieval Augmented Generation (Gao et al., 2024), a method which extends the generation input with retrieved information, is another common way to integrate external knowledge into the generation process. Unlike the other two generation approaches we explored, we use retrieval, not to extract Wikidata labels, but to retrieve textual data from Wikipedia. This avoids having to rely on a KB, which makes available multilingual labels, allowing for an approach that could adapt to other knowledge bases more easily. We explore a variety of strategies for querying, chunking and few-shots selection and we show that the best performing variant performs almost on par with the LLM-prompting approach while being less restrictive in terms of resources.

3.2 Data

3.2.1 Test Data

We evaluate our models on three datasets of (KG, Russian Text) pairs.

Seen (in domain) data. This data is available on the web and is therefore likely to have been included in the LLM training data. The dataset consists of the 1,102 instances used for the evaluation of the WebNLG 2020 and 2023 campaigns. In this dataset, the KGs contain seen entities and predicates i.e., entities and predicates that are present in the Russian WebNLG training data.

Unseen (Out of Domain) Data. This is the data set created in a previous chapter that is not available on the web. Derived from the WebNLG and KELM (Agarwal et al., 2021) datasets, it contains the following test sets.

From the WebNLG dataset, two datasets are derived. The Unseen Category (WebNLG-C, 1,251 instances) contains WebNLG graphs from the English training data which root DBPedia category does not belong to the set of categories present in the Russian WebNLG training data. In contrast, the Unseen Entity testset (WebNLG-E, 192 instances) contains graph instances that are from seen categories (i.e., graphs root category of which belong to one of the 16 categories present in the Russian WebNLG train or dev set) but which entities are unseen.

KELM is a silver dataset of (Wikidata graph, English text) pairs created using distant supervision. Similarly to the WebNLG unseen testsets, there are two gold test sets from KELM: a dataset where all graph properties are present in the WebNLG training/dev data for Russian but some entities are not (KELM-E, 2126 instances) and a dataset which graphs contain both unseen entities and unseen properties (KELM-E+P, 1311 instances).

Unseen (Out of Domain) New Graphs Data. For both KELM and WebNLG, the graphs are available on the web and might have been included in the LLM training data. We create a dataset of 472 (Unseen New Graphs, Russian Text) pairs by extracting subgraphs from Wikidata and manually editing the Russian texts that were automatically generated from these graphs by our best LLM prompting approach (Cf. Section 3.4, prompting configuration PRU/+L/D.). The graphs were constructed by first identifying non-existing or rare properties in KELM and WebNLG; manually deciding on relevant Wikidata categories (i.e., categories where subsets of these properties often co-occur); defining a set of possible graph schemas for each category; and using these schemas to extract matching subgraphs from Wikidata. This new dataset consists of 472 (Wikidata graphs, Russian text) pairs with graphs containing from 1 to 10 triples. The categories are Airlines, Chemicals, Conferences, Statistics, Dancers, Diseases, and Tournaments. Each of these categories contains graphs of 1 to 7 triples. We also combined some of the graphs from Statistics and Airlines categories to get a Mixed category containing only graphs of size 10. Table 3.1 provides the statistics of our test set, which we further call NG_{500}.

Category	**Graph Length**								**Total**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**10**	
Airlines	9	9	10	10	10	9	9	0	66
Chemicals	9	9	10	10	10	9	9	0	66
Conferences	9	9	10	10	10	9	9	0	66
Dancers	9	9	10	10	10	9	9	0	66
Diseases	9	9	10	10	10	9	9	0	66
Statistics	9	9	10	10	10	9	9	0	66
Tournaments	9	9	10	10	10	9	9	0	66
Mixed	0	0	0	0	0	0	0	10	10
Total	63	63	70	70	70	63	63	10	472

Table 3.1: Distribution of graph lengths across different categories in the newly created NG_{500} dataset.

3.2.2 Fine-Tuning and Few-Shot Data

For fine-tuning, we use the WebNLG training data for Russian, an aligned corpus of 6,339 (KG, Russian text) pairs. We also use this dataset as a retrieval database for few-shots selection, selecting few shots (KG,text) pairs from either of these two datasets based on the input graph.

3.3 Fine-tuning on Labeled data.

The state-of-the-art KG-to-text model for Russian is Interno (Kazakov et al., 2023), which is a FRED-T5 model (Zmitrovich et al., 2024) fine-tuned on the WebNLG training data for

Russian. FRED-T5 (1.7B parameters) is based on the T5 architecture (Raffel et al., 2020) and trained on a large corpus of Russian texts using a mixture of denoising objectives.

To facilitate entity verbalisation, the authors modify the input in the training and test sets to include, in addition to the knowledge graph, the Russian entity and property labels which are provided by the WebNLG data. As our unseen test sets do not contain such labels, we extract Russian entity and property labels from the Wikidata Knowledge Base using the SparQL queries shown in section B.2. Retrieved labels are added to the generation input under the key "additional links" using the format "English_name=Russian_name".

We then run the Interno model on both seen and unseen data, comparing results when the input is or is not enriched with Russian labels.

Dataset	Descs	Labels
WebNLG Testset	35%	46%
KELM-E	19%	27%
KELM-E+P	42%	59%
WebNLG-E	34%	41%
WebNLG-C	34%	43%
Test-500	11%	12%

Table 3.2: Availability of labels and descriptions in Russian across datasets.

Results. Table 3.3 shows the results and Table 3.2 indicates the ratio of entities and properties present in the input graphs for which a Wikidata label could be retrieved. The results clearly show that, when a sufficient ratio of labels can be retrieved, enriching the input with Russian labels improves performance. Thus, for the three datasets where the ratio of retrieved labels is the lowest (NG_{500}: 12%, KELM-E: 27%, W_E: 41%), the improvement at generation time is not statistically significant while for the other three datasets where this ratio is higher (W_C: 43%, W_S: 46%, KELM-E+P: 56%), it is.

3.4 LLM Prompting

For LLM-Based generation, we use Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and experimented with a few-shots prompting strategy that varies across three dimensions: the input, the prompting strategy and how few-shots are selected. We do not explore 0-Shot as preliminary experiments showed a clear degradation in performance compared to few shots.

We gradually tested and combined the three dimensions, eventually exploring 11 configurations (cf. Table 3.5).

[16]Re-running the authors code on this data set using the DBPedia labels as they did, produced scores similar but not identical to the scores presented in the author's paper.

Model	BLEU		chrF++	B_{F1}
WebNLG Seen				
Interno	24.59	(B)	0.44	0.82
Interno+labels [16]	**53.67**	(A)	**0.69**	**0.92**
WebNLG E				
Interno	22.43	(A)	0.48	0.85
Interno+labels	23.58	(A)	0.50	0.86
WebNLG C				
Interno	23.02	(B)	0.42	0.83
Interno+labels	**24.62**	(A)	**0.46**	**0.84**
KELM-E				
Interno	17.57	(A)	0.36	0.80
Interno+labels	17.74	(A)	0.44	0.84
KELM-E+P				
Interno	12.39	(B)	0.36	0.78
Interno+labels	**16.10**	(A)	**0.45**	**0.81**
Unseen-500				
Interno	14.93	(A)	0.34	0.77
Interno+labels	14.75	(A)	0.36	0.79
All Unseen				
Interno	17.52	(B)	0.37	0.80
Interno+labels	**18.89**	(A)	**0.44**	**0.83**
All				
Interno	18.72	(B)	0.39	0.81
Interno+labels	**24.83**	(A)	**0.49**	**0.84**

Table 3.3: **Results of (Kazakov et al., 2023)'s Pretrained Model fine-tuned on WebNLG 2020 training data, tested with and without input enrichment** The letters A/B indicate whether the difference between two scores occurring in the same cell is or not statistically significant. We assess statistical significance using Paired Bootstrap Resampling as described by Koehn (2004) and consider a difference as significant if confidence is at least 90%.

3.4.1 Input

We consider different types of input depending on whether the input consists solely of a linearised graph or of the graph augmented with Wikidata extracted information. When enriching the input with Wikidata information, we consider 3 variants: adding property descriptions for all properties present in the graph; adding entity and property descriptions for all entities and properties present in the graph; and adding target language labels for all properties and entities present in the graph.

Property and entity descriptions are extracted from Wikidata (when available) using the Wikidata *Descriptions* property. To only keep the most informative part of the description, we remove all text within parentheses and any text that appears after the first end-of-clause punctuation (full stop, colon, or semicolon). Similarly, labels are extracted using the Wikidata *Label* property and the resulting dictionary of entity/label pairs is inserted into the input under the key *labels*.

As shown in Table 3.2, not all Wikidata entities and properties have labels and descriptions, and their availability varies depending on the dataset.

Dimensions	Options	Explanation
Input	Graph	The input is the RDF graph
	+P	RDF Graph and Wikidata Property Descriptions
	All	RDF Graph, Wikidata Property Descriptions and Entity Descriptions
	+L	RDF Graph, Wikidata Entity Labels
Prompting	Direct	Generate a text verbalising this graph
	Cumulative	Generate a text for each fact in turn then combine the resulting texts into one (two consecutive prompts).
Few-Shot Selection	Random	Random examples
	Size-Based	Examples whose graph size is the same as that of the input graph
	Overlap	Examples whose graph maximise property overlap, size and semantic similarity

Table 3.4: Short Description of the Prompting Options

3.4.2 Generation strategies

We compare two generation strategies: direct and cumulative.

The direct generation strategy (Direct) simply instructs the model to verbalise the input. The cumulative generation strategy (Cumulative) proceeds in multiple steps as follows. First, each individual triple in the input graph is lexicalized separately using the direct approach. The result of each inference call is then collected and passed as input to the LLM with the instruction to combine the triple lexicalisations into a coherent output text. When the input graph is of size one, there is no need to continue to the cumulative generation step since there will be only one sentence and the combination step is omitted.

3.4.3 Few-shot selection strategies

Few shots are retrieved from the Russian WebNLG dataset (20K graph/text instances). To select the few-shots, we explore 3 strategies: random, size-based and property-based.

Random. We randomly select KG/Text examples making sure that the selected few-shots contain graphs of different size. This approach provides the model with examples of how triples are lexicalised for graphs of different sizes – this helps illustrate the impact of context on the lexicalisation of triples.

Size based. We randomly select KG/Text examples which have the same KG size as that of the input KG.

Property based. The property-based approach seeks to maximise the property overlap between the input graph and the selected few shots. To this end, all available KG/Text instances are sorted according to their property overlap with the input graph. Top examples are then used as few shots. If there are properties from the input graph that do not appear in the set of selected examples, their LaBSE (Feng et al., 2022) encoding is used to search for the most similar properties in WebNLG and the corresponding properties are used to find relevant few shots. The whole process is repeated until the target number of few-shots is reached. Algorithm 1 specifies this selection process in more detail.

Briefly, *the input* is either the knowledge graph (G) or the KG enriched with Wikidata property descriptions (+P), Wikidata Russian entity labels (+L), Wikidata Russian property descriptions and entity labels (+P+L) or Wikidata property and entity descriptions

Algorithm 1 Property-Based Few-Shot Selection

Require: G ▷ Input graph
Require: P_G ▷ Set of properties from G
Require: D ▷ WebNLG dataset
Require: k ▷ Number of few-shots
Require: I ▷ FAISS index of encoded properties
Ensure: FS ▷ Set of few-shot examples

Initialize $FS \leftarrow \emptyset$
Initialize $P \leftarrow P_G$
while $|S| < k$ **do**
    if $|P| = 0$ **then**
        $P \leftarrow P_G$
    end if
    Sort D according to:
    1. Number of overlapping properties with P (descending order)
    2. Size similarity to G (descending order from graphs of size $|G|$ to largest graph size in D)
    if graphs in D overlap with P **then**
        Select the top example e from D
        $S \leftarrow S \cup \{e\}$
        $P \leftarrow P \setminus \mathrm{P}(e)$
    else
        Encode P using LaBSE
        Query FAISS for similar properties P'
        $P \leftarrow P'$
    end if
end while
return S

	Seen	**Unseen**				
	WebNLG	WebNLG		Kelm		NG_{500}
FS/I/NLG	W_S	W_E	W_C	KELM-E	KELM-E+P	
Interno+labels	53.67	23.58	24.62	17.74	16.10	14.75
Baseline						
R/G/D	16.86	24.38	25.08	20.50	16.16	8.80
Input variants						
R/+P/D	16.79	26.58	24.88	20.52	14.31	8.17
R/+L/D	20.53	25.47	25.88	21.88	16.45	8.01
R/+All/ D	14.85	22.69	22.12	17.94	14.18	5.60
FS variants						
S/G/D	16.91	24.75	23.53	19.18	16.40	8.90
P/G/D	19.76	25.48	23.64	18.17	16.88	8.47
NLG variant						
R/G/ Cum	15.62	21.94	22.55	19.88	13.84	18.16
Combined						
P/+L/D	23.16	23.75	25.04	19.80	**18.33**	8.34
P/+P+L/D	22.77	24.11	24.76	20.52	16.32	6.75
Russian Few Shots						
PRU/+L/D	**32.15** (A)	**30.09** (A)	**25.76** (A)	21.36 (A)	16.19 (A)	20.54 (B)
PRU/G/D	30.81 (B)	29.76 (A)	24.76 (B)	21.54 (A)	16.51 (A)	**21.51** (A)

Table 3.5: **BLEU Scores for the various LLM-Prompting configurations (FS/I/NLG).** FS specifies the few shot selection strategy (R:Random, S:Size, P:Property overlap on fewshots selected from the English WebNLG data, PRU: Property overlap on fewshots selected from the Russian WebNLG data); I, the input (G:Graph, +P:Input augmented with properties descriptions, +All:Input augmented with properties and entities descriptions, +L:Input augmented with Russian labels) and NLG, the generation strategy (D:Direct, Cum:Cumulative). Underlined scores are best score within a block, boldface indicates best score for column.

(+All). The *prompting strategy* is to either prompt the LLM for a direct verbalisation of the entire graph (D) or to first verbalise each fact in the input graph and then aggregate the cumulative verbalisations into a fluent text (Cum). For *few shot selection*, we select 4 few shots[17] and explore three strategies: random (R); size-based, where we select few shots containing graphs which have the same size as the size of the input graph (S)[18]; and property based, which consists in selecting few shots whose graphs contain properties similar to the set of properties present in the input graph (P when the selected few-shots are from the English retrieval base; PRU, when they are in Russian). Some examples of

[17]In the preliminary stage, we experimented with 0, 2 and 4 few shots and found that 4 gave best results.

[18]Since the WebNLG data from which we retrieve the few shots has a maximum graph size of 7, when the input graph is larger we select few shots of size 7

those prompts are given in section B.1.

3.4.4 Results

In Table 3.5 (*Input Variants*), we see that including Wikidata labels in the prompt (R/+**L**/D) helps improve generation, confirming the results of the previous section. Table 3.5 (*NLG Variants*) shows that, except on NG_{500}, the cumulative approach generally underperforms direct generation. A look at the data reveals that, different from the other testsets, the NG_{500} graphs do not easily support linguistic aggregation and therefore for this dataset, simply listing a verbalisation of the input triples often suffice to generate a text that matches the reference. We provide an example of such a graph in Figure 3.1.


```
[
    {
        "subject": "Lufthansa",
        "property": "airline accounting code",
        "object": "220"
    },
    {
        "subject": "Lufthansa",
        "property": "country",
        "object": "Germany"
    },
    {
        "subject": "Lufthansa",
        "property": "callsign of airline",
        "object": "LUFTHANSA"
    },
    {
        "subject": "Lufthansa",
        "property": "official website",
        "object": "https://www.lufthansa.com/de/de/"
    }
]
```


Figure 3.1: A graph example from NG_{500} test set, with a graph which provides an advantage to cumulative generation approach.

The *FS Variants* block indicates that none of the few shot selection methods systematically improve results. However, Table 3.5 (*Combined*) shows that combining property-based few shot selection with an input enriched with labels yields good results overall. Finally, the *Russian Few Shots* variants indicate that augmenting the input with Wikidata labels and selecting Russian rather than English few-shots yields best results for all test sets except for NG_{500}, which, as observed in the previous section is the test set for which the ratio of retrieved labels is the lowest which in turn decreases the impact of adding labels to the input.

In sum, we find that the best prompting configuration (PRU/+L/D) is a configuration where the input graph is enriched with Wikidata entity and property labels and the selected few-shots are example graph/text pairs whose texts are in Russian and whose graphs have maximum similarity in terms of size and properties with the input graph.

3.5 Retrieval Augmented Generation for LLM

RAG provides a natural way to enrich the input by retrieving additional data from a vector base containing task-relevant knowledge and adding the retrieved data to the generator input. We combine RAG with few-shots prompting by including (i) four few-shots and (ii) the retrieved content in the prompt as context for decoding.

Typically, RAG retrieves data based on a given query (here the generation input) and the vector base from which data is retrieved contains data chunks. We explore various options for the query, the vector base chunks and few-shots selection.

We used Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as the base LLM model. Additionally, we ran experiments with Qwen2.5-7B-Instruct (Team, 2024; Yang et al., 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023).

3.5.1 Paragraph Retrieval

We first experiment with a simple approach where the additional data is the first paragraph of the Wikipedia article associated with the graph root entity. As previously observed (Lebret et al., 2016), this paragraph contains much of the information contained in Wikidata graphs. Hence, we explore a simple approach where, instead of retrieving labels from the Wikidata knowledge base or searching a vector base for relevant content, we include in the prompt the first paragraph of the Wikipedia page corresponding to the root entity of the input graph. We use the WikipediaRetriever tool from the LangChain library (Chase, 2022) to retrieve the first paragraph of the Wikipedia article matching the root entity and add it to the prompt.

As shown in Table 3.7 (*BL - Graph + WKP Paragraph*), this approach does not improve results, probably because the provided information is too coarse grained introducing a semantic mismatch between the input graph and the retrieved paragraph. We also experiment with another LLM (Mistral) and with combining paragraph selection with random few-shots but again the results were worse than when including Wikidata labels in the prompt and selecting few shots based on their property overlap with the input graph.

3.5.2 Chunks

We derive text chunks from the Russian Wikipedia dump. We consider two types of chunks for the retrieval data base: sentence and graph size level.

In *the sentence level* version, each chunk in the retrieval base consists of a single Wikipedia sentence. The sentences are pre-extracted by processing Wikipedia dump [19]

[19] https://dumps.wikimedia.org/ruwiki/ accessed in March 2022

and removing all the Wikipedia-specific pages. The parsing was done using mwparserfromhell [20]. Then, the resulting articles are cleaned and split into sentences using the NLTK sentence segmenter (Bird et al., 2009), resulting in more than 60 million sentences. For each triple in the input graph, we retrieve 3 sentences. We preliminarily experimented with retrieving 1, 3 and 7 sentences per triple, 3 yields the best results.

In the *graph-based* version, we create chunks to match the size of the input graphs (graph-size chunks) and for each graph of size k, we retrieve n chunks of size k with $n \in \{3, 7\}$.

To create graph-size chunks, we first compute the character length distribution (see Figure 3.2) of the texts for each graph size (1 to 10 triples) in the graph/text WebNLG data. Then we create text chunks that simulate that distribution using the following algorithm. As we merge the sentences extracted from Wikipedia into larger blocks in terms of size based on character number (as shown in the graphs it has a more pronounced single peak), we try to simulate the character number distribution for a specific graph size, assuming it to be a normal distribution with a mean being the same as a centre of a peak. More precisely, to divide a Wikipedia text into blocks for graph size k we use the following algorithm. Let's say we have started a block b and we decide whether we add the next sentence s to it or not. We then draw two random variables:

$$x_1 \sim U(0, f_{\mathcal{N}(m,\sigma)}(|b|)) \tag{3.1}$$

and

$$x_2 \sim U(0, f_{\mathcal{N}(m,\sigma)}(|b + s|)) \tag{3.2}$$

where m is the average character length of the lexicalisations for graphs of the required size, σ refers to a variance we want to have, f is a probability density function and U is a uniform distribution, while $\mathcal{N}$ is a normal one. Then, if $x_1 < x_2$ we add the sentence s to our current block. Otherwise, we start a new block with the sentence s.

In essence, we use the character length distribution of the texts for each graph size in the WebNLG training data and then we create text chunks that simulate that distribution. This way of simulating the block size distribution allows us to have the average length of the chunks in the vector base close to the average character length of the lexicalisations for graphs of each size while also having some variability in the block lengths.

We also experiment with chunks the text size of which matches the average length of the WebNLG texts associated with single fact graphs (1tblocks). For a graph of size k, we then retrieve $n \times k$ triple-sized chunks.

In both cases (sentence- and graph-based chunks), chunks are embedded using the LaBSE multilingual encoder (Feng et al., 2022) and applying binary quantization to reduce the size of the embeddings. The total size of the original vector base was in total more than 200 gb resulting in high retrieval latency. Binary quantization showed similar results on a small batch with a huge advantage in terms of processing time. These are then indexed using a "flat" binary index from the Faiss library (Douze et al., 2024).

[20] https://github.com/earwig/mwparserfromhell

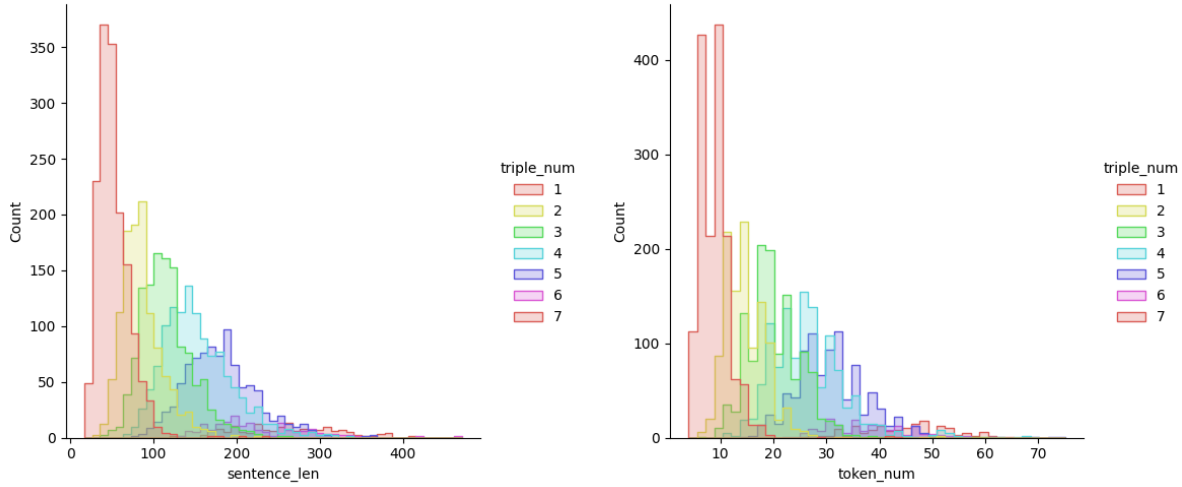


Figure 3.2: Distribution of sentence length for each graph size in number of symbols (left) and number of tokens (right)

3.5.3 Queries

We experiment with the content and the format of the query using either the entities or the facts contained in the input graph. Specifically, the query consists of the entity pairs present in the input graph triples (ents), the same entity pairs but with underscores removed (entsep), the set of triples present in the input graph where triple subject, predicate and object are separated by a space (trps) and the set of triples with underscores removed (trpsep). In table 3.6 we summarise the options for queries we experimented on in RAG.

Query Name	Description	Example
Entities (ents)	Entity pair from a triple	"William_Anders Fighter_pilot".
Processed entities (entsep)	Entities pairs from a triple with extra symbols (like underscores) removed	"William Anders Fighter pilot"
Triples (trps)	Triple subject, predicate and object separated by spaces	"William_Anders occupation Fighter_pilot"
Processed triples (trpsep)	Triple subject, predicate and object separated by spaces with underscores removed.	"William Anders occupation Fighter pilot"
Graph entities (unionentsep)	Linearised graph with repeated entities deduplicated	"William Anders occupation Fighter pilot country_of_citizenship United_States "

Table 3.6: Query variants

3.5.4 Few-shots

By default, we use 4 shots but to assess the impact of these few shots, we also consider a 0-shot variant. Based on the results obtained with prompting (see section 3.4), we select few shots from the Russian WebNLG train and dev data and compare two selection methods: random (R) and based on property and size similarity with the input graph (P).

3.5.5 Results

	Seen	**Unseen**				
	WebNLG	WebNLG		Kelm		NG_{500}
Model	W_S	W_E	W_C	KELM-E	KELM-E+P	
[*BL*] - Graph + WKP paragraph	2.38	3.71	3.35	1.42	1.92	6.64
0-Shot RAG						
[1] - Entity based retrieval (ents)	2.53	3.75	3.54	1.49	2.24	5.85
+ Few Shots						
[2] - [1] + R	18.84 (B)	28.82 (B)	**24.63 (A)**	21.41 (A)	16.60 (B)	19.55 (B)
[3] - [1] + P	**31.03 (A)**	**31.95 (A)**	**24.57 (A)**	19.50 (B)	**17.22 (A)**	**20.54 (A)**
Other LLMs						
[2] using Mistral	16.92	24.45	22.81	16.64	16.34	16.46
[2] using Qwen	16.00	22.47	21.79	15.63	13.59	20.08
Other Queries						
[5] - R + entsep	18.06	28.01	24.00	20.64	16.70	19.71
[6] - R + trps	17.56	26.49	21.78	18.79	15.85	17.43
[7] - R + trpsep	17.71	28.89	21.92	18.75	15.65	17.62
Graph-based Chunks						
[5] + 1tblocks (3)	17.64	28.86	23.36	20.44	16.64	19.34
[5] + graph-size-chunks (3)	18.42	26.54	23.82	20.19	15.96	20.04
[5] + graph-size-chunks (7)	18.33	27.35	23.67	20.99	16.32	19.41

Table 3.7: **BLEU scores for RAG.** The baseline (BL) is a LLama3.1-based 0-shot retrieval model which retrieves a Wikipedia paragraph based on the root entity of the input graph. For all models, except those with graph-based chunks, the vector base chunks are Wikipedia sentences. For the graph-based chunks the number in brackets indicate the number of retrieved chunks per graph (graph-size-chunks) or per triple in the graph (1tblocks).

Table 3.7 shows the results highlighting the impact of the various dimensions we explored. The low results for 0-shot RAG underscore the importance of augmenting the prompt with some few-shots. The results also demonstrate that the most impacting factor is the few-shot selection method. As with prompting, selecting few-shot examples based

on their property and size similarity with the input graph yields the best results. From the various query types investigated, simply using the entity pairs present in the graph triples yields the best results. Similarly, sentence size chunks (the default setting) yield better results than graph size ones. Finally, a comparison with Mistral and Qwen shows an advantage for Llama.

3.6 Comparing the Three Approaches

We compare the three approaches (fine-tuning, prompting, RAG) using the best model we identified for each approach i.e., augmenting the input with Wikidata labels for the Interno model fine-tuned on the Russian WebNLG data (Interno+Labels in Table 3.3); Prompting with, in addition to the input graph, (i) Russian few-shots selected based on their similarity in terms of size and properties with the input graph and (ii) entity and property labels retrieved from Wikidata (PRU/+L/D in Table 3.5); and RAG with the same few-shots, entity based queries and sentence level chunks ([3] in Table 3.7).

3.6.1 Automatic Evaluation

Figure 3.3 shows the BLEU scores (scores for other metrics are shown in section B.3).

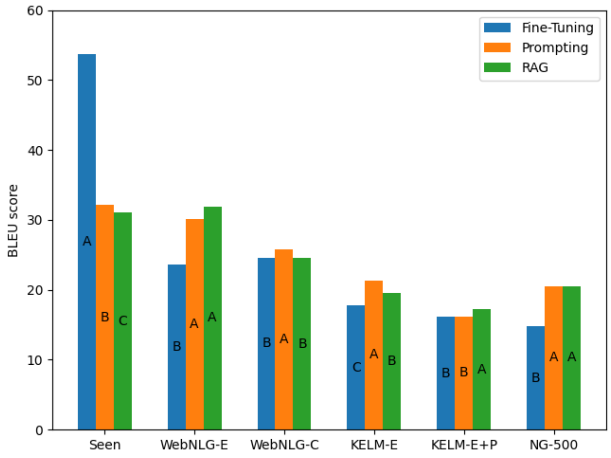


Figure 3.3: **BLEU Scores for the Best Fine-Tuning, Prompting and RAG models.** Fine-tuning underperforms on unseen data.

Differences between approaches. On seen data, the fine-tuning approach outperforms LLM prompting and RAG by a large margin highlighting the effectiveness of domain adaptation on carefully curated data. However, it systematically underperforms on unseen testsets demonstrating limited generalisation capacity.

Compared to RAG where the added information comes from Wikipedia text, the LLM prompting approach, which extends the input with Russian entity and property labels, has a slight advantage outperforming it or performing on par with it on four out of the five unseen test sets. However, it requires that labels be available in the target language

which is a limitation. In contrast, the RAG approach is less constrained simply requiring relevant target language text.

Differences between test sets. Roughly, the scores of the RAG and the prompting models follow a similar trend on the 5 unseen datasets. Notably, scores are highest on W_E and W_C, which is consistent with the fact that the two datasets are available on the web and are likely to have been included in Llama training data. The middle scores on KELM-E and NG_{500} match the low ratio of labels available for these two datasets (27% and 12% respectively) which in turn often correlates with a low availability of Wikipedia texts in Russian for the corresponding graphs (we observed that entities which have no Russian labels in Wikidata are also often entities for which there is no Wikipedia page in Russian [21]). Finally, the lowest score on KELM-E+P reflects the difficulty of correctly verbalising graphs which contain both unseen entities and unseen properties.

3.6.2 Human Evaluation

We also conduct a human evaluation on the three best performing models.

We randomly extract 50 instances from seen WebNLG data, 50 from WebNLG and KELM unseen data, and 50 instances from NG_{500}, totaling 150 samples.

Using the Prolific platform [22] we select 3 annotators who met the following criteria: Russian as their native language, English fluency, and successful completion of a qualification test assessing their ability to identify additions, omissions, and repetitions in text generated from knowledge graphs. To pass the test, annotators were required to provide fully correct responses on at least 10 out of 15 instances. This is the same test as described in the human validation paragraph in section 2.3.

We show the annotators the original graph in table format as well as 3 texts verbalising it, one for each of the three selected models. For that we built an annotation website using potato annotation tool (Pei et al., 2022). An example of GUI can be seen in fig. 3.4. The order of the models, shown to the annotators, is random and does not contain any information about the models themselves. The annotators are then asked to answer the following four questions about the three texts:

- Please, select the text which is the **closest** to the data shown. Consider both additions and omissions.
- Please, select the text which is the **farthest** from the data shown. Consider both additions and omissions.
- Please, select the text which is the **most fluent**.
- Please, select the text which is the **least fluent**.

Then we treat the answers as a ranking of the models. The closest and the farthest information gives us the ranking for faithfulness, while the most fluent and the least fluent

[21] We looked at all the entities having no Russian labels in the unseen test sets and found that less than 5% of them had a related Wikipedia article in Russian.

[22] https://prolific.com/

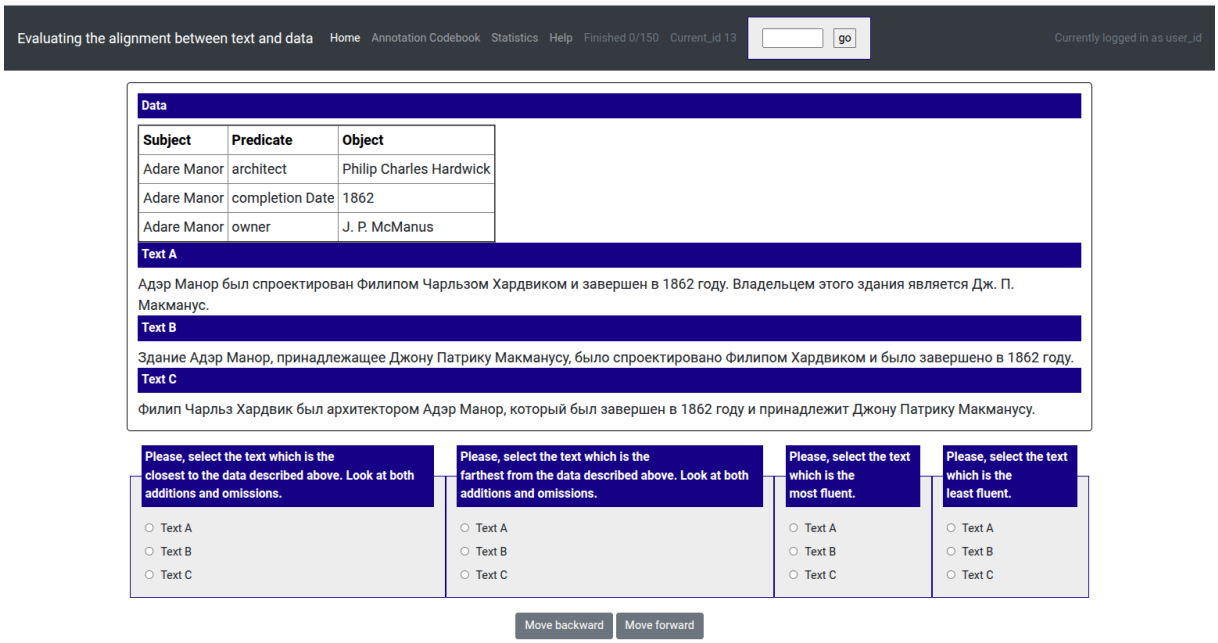


Figure 3.4: Human annotation GUI example

give us the ranking for fluency. We transform each ranking into scores by assigning 3 points to the best model, 1 point to the second, and 0 points to the last. We then accumulate the points by getting their sum and dividing it by the maximum possible score (3 times the number of rankings). Later in subsection 3.6.3 we show some input/output examples illustrating the various types of errors made by the three generation methods.

	Seen	**Unseen**		
		W+K	500	All-U
Faithfulness				
Fine-Tuning	**0.48**	0.37	0.22	0.30
Prompting	0.42	0.47	**0.61**	**0.54**
RAG	0.43	**0.49**	0.50	0.49
Fluency				
Fine-Tuning	**0.45**	0.36	0.29	0.33
Prompting	0.41	0.44	**0.54**	0.49
RAG	0.41	**0.53**	0.50	**0.52**

Table 3.8: **Human Evaluation Results**. We provide the calculated scores for each approach by source of the data.

The results of human evaluation are shown in table 3.8. We also calculated Fleiss' kappa for Inter-Annotator Agreement, which is shown in table 3.9. For fine-tuning, the results confirm the automatic evaluation with the fine-tuned model performing best on seen, and worse on unseen data. On unseen data, prompting is generally judged more faithful but RAG is judged to be more fluent. We conjecture that the better fluency of

	All	By source		
		Seen	W+K	500
closest	0.22	0.25	0.17	0.19
farthest	0.23	0.28	0.13	0.18
most fluent	0.17	0.15	0.18	0.16
least fluent	0.16	0.08	0.24	0.19

Table 3.9: **Fleiss' kappa**. Estimating inter-annotator agreement on each question on all of data, as well as on each part of it by source.

RAG stems from the fact that the human written, Wikipedia texts making up the retrieved chunks provide a better context for generation than the stilted WebNLG crowdsourced texts contained in the prompting few-shots. Conversely, the better faithfulness of the prompting approach is likely due to the fact that the labels included in the prompts help improve the verbalisation of unseen entities and properties – this is consistent with the results in Table 3.5, where augmenting the prompt with Wikidata labels is shown to improve generation performance.

3.6.3 Comparing the Three Approaches: Case Study

In table 3.10 we provide an example, where our annotators have deemed the RAG model as the most fluent, Prompting model as the closest to the data and our Fine-Tuning model (Interno+Labels) as both the farthest produced text from the data and the least fluent. The example is taken from a NG_{500} dataset. To make the analysis of the models easier, we provide translations of the texts to English, trying to keep the same mistakes when possible. We also use a system of different underline styles to show which parts of the texts refer to which fact (triple of the graph).

The simple underlined text signifies that the lexicalisation of the fact is correct. The *emphasized orange underlined text* signifies that the lexicalisation partly describes the fact. The *emphasized orange text* shows unnecessary repetition in the text, and the **bold blue text** shows information not present in the data. Different underline styles (like, one line, dot line, wave line) highlight text fragments which are supposed to be verbalisations of the corresponding triples.

In the example, we can see that the Prompting method stays closest to the described data, however, it mostly states the facts without connecting them properly. This results in a loss of perceived fluency. RAG model verbalises all the facts correctly, however, it additionally states a fact that can be derived from the chemical formula. It is correct, but it can be considered a slight difference from the data. Finally, we can see that the Fine-Tuning model did not verbalize all of the facts, and overall produced a poor text with repetitions and information not present in the data.

3.7 Conclusion

Focusing on Russian as a target language, we examine how well different models can convert English-centric knowledge graphs into Russian text on datasets where the input graphs contain unseen entities and/or properties.We compare three common methods for generating text: fine-tuning a pre-trained model on task-specific data, prompting a large language model and Retrieval Augmented Generation. For each method, we design and evaluate multiple variants, selecting the one that performs best.

Across datasets we find that the fine-tuning approach excels on in distribution data but systematically underperforms on unseen data, exposing limited generalisation capacity. Contributing to RQ2, we find that LLM-prompting with labels information and few-shot examples has a slight advantage over RAG in terms of automatic metrics; and that, based on a human evaluation, RAG produces more fluent, but less faithful text than LLM-prompting.

In the next chapter, we extend our LLM prompting technique to see how it applies in comparison to other multilingual approaches for different languages. We will also address the use of LLM prompting for low-resource languages.

Graph	[["guanethidine", "has use", "medication"], ["guanethidine", "ATC code", "S01EX01"], ["guanethidine", "canonical SMILES", "C1CCCN(CCC1)CCN=C(N)N"], ["guanethidine", "InChI", "InChI=1S/C10H22N4/c11-10(12)13-6-9-14-7-4-2-1-3-5-8-14/h1-9H2,(H4,11,12,13)"], ["guanethidine", "chemical formula", "$C_{10}H_{22}N_4$"]]
RAG	Гуанетидин представляет собой препарат, который используется в качестве лекарства. Его АТС-код - S01EX01. Химическая формула гуанетидина - C10H22N4. *В его структуре присутствуют атомы углерода, водорода, азота.* Вещество имеет канонический SMILES-код C1CCCN(CCC1)CCN=C(N)N и InChI-код InChI=1S/C10H22N4/c11-10(12)13-6-9-14-7-4-2-1-3-5-8-14/h1-9H2,(H4,11,12,13).
Translation	Guanethidine is a product that is used as a medicine. Its ATC code is S01EX01. The chemical formula of guanethidine is C10H22N4. *Its structure contains carbon, hydrogen, and nitrogen atoms.* The substance has the canonical SMILES code C1CCCN(CCC1)CCN=C(N)N and InChI code InChI=1S/C10H22N4/c11-10(12)13-6-9-14-7-4-2-1-3-5-8-14/h1-9H2,(H4,11,12,13).
Prompting	Гуанетидин является лекарственным средством, имеет АТС код S01EX01, химическую формулу $C_{10}H_{22}N_4$, а также SMILES-код C1CCCN(CCC1)CCN=C(N)N и InChI-код InChI=1S/C10H22N4/c11-10(12)13-6-9-14-7-4-2-1-3-5-8-14/h1-9H2,(H4,11,12,13).
Translation	Guanethidine is a medication, has ATC code S01EX01, chemical formula $C_{10}H_{22}N_4$, and also SMILES code C1CCCN(CCC1)CCN=C(N)N and InChI code InChI=1S/C10H22N4/c11-10(12)13-6-9-14-7-4-2-1-3-5-8-14/h1-9H2,(H4,11,12,13).
Fine-Tuning	*1S/C10H22N4* - лекарственное средство, которое имеет *C1CCCN(CCC1)CCN=C(N)N* и canonical smiles C1CCCN(CCC1)CCN=C(N)N. Оно имеет АТС код S01EX01 и **может быть использовано как для внутреннего, так и для наружного применения.**
Translation	*1S/C10H22N4* is a medication that has *C1CCCN(CCC1)CCN=C(N)N* and canonical smiles C1CCCN(CCC1)CCN=C(N)N. It has ATC code S01EX01 and **can be used for both internal and external use.**

Table 3.10: An example of a graph and three systems' outputs, where our annotators have marked Prompting as the closest model to the data, Fine-Tuning as the farthest text from the data, RAG as the most fluent text and Fine-Tuning as the least fluent text. The simple underlined text signifies that the lexicalisation of the fact is correct. The emphasized orange underlined text signifies that the lexicalisation partly describes the fact. The emphasized orange text shows unnecessary repetition in the text, and the bold blue text shows information not present in the data.

4

Multilingual Verbalisation of Knowledge Graphs

Contents

4.1 Introduction 62
4.2 Data 63
4.3 Models 65
4.3.1 Supervised Approaches 66
4.3.2 LLM-based Generation 66
4.4 Automatic Evaluation 67
4.4.1 Metrics 67
4.4.2 Results 68
4.5 Error Analysis 71
4.5.1 Analysing the FewShot Model 71
4.5.2 Comparing Models 72
4.6 Extension of the Experiments 72
4.6.1 Data 72
4.6.2 Models 74
4.6.3 Results 75
4.7 Conclusion 78

In this chapter, we investigate different ways of extending KG-to-Text generation to multiple languages. Focusing on five target languages (Chinese, English, French, Russian, and Spanish), we start by creating silver multilingual (graph, text) training data and gold test data. We then compare three KG-to-Text generation approaches: a hybrid model combining a state of the art KG-to-English model with machine translation (MT) from English into the target languages (NLG+MT); direct machine translation where selected MT models fine-tuned on our silver multilingual training data are directly applied to the input graph (FTMT); and few-shot LLM prompting (FewShot).

We explore different prompting strategies, and show that our best prompting strategy outperforms both the FTMT and the NLG+MT approach for all 5 languages.

4.1 Introduction

Previous work on KG-to-Text generation has mostly focused on monolingual approaches, most of them targeting English (Schmitt et al., 2021, 2020; Yuan and Faerber, 2023; Yin and Wan, 2022; Schmitt et al., 2021, 2020; Koncel-Kedziorski et al., 2019; Li et al., 2021; Duma and Klein, 2013). A key reason for this is the lack of naturally occurring parallel (KG, text) data and the fact that KG-to-Text generation models are either trained or fine-tuned on such data. To extend KG-to-Text generation to multiple languages, we address both of these issues. On the one hand, we create multilingual silver training and gold test data for KG-to-Text generation. On the other hand, we explore three alternative models for KG-to-Tex generation: a hybrid model which does not require multilingual training data as it combines KG-to-English generation with machine translation (MT) from English into the target languages; direct machine translation where selected MT models fine-tuned on our silver multilingual training data are directly applied to the input graph; and few-shot LLM prompting, which does not require training data but necessitates a retrieval base for the few shot which is akin to training data.

We make the following contributions.

Data. We provide a methodology for creating a multilingual silver dataset of (graph,text) pairs where each graph is aligned with a matching text verbalising this graph into multiple languages. We apply this methodology to create 30K multilingual (graph,text) pairs for English, Spanish, French, Chinese and Russian. We exploit the resulting dataset both as a retrieval base for few shot selection and for fine-tuning text-to-text machine translation models on the KG-to-Text task. To support evaluation, we also provide a gold test set of (graph,text) pairs for each of our 5 target languages.

Models. We develop and compare three approaches for multilingual KG-to-Text generation: direct machine translation with selected MT models fine-tuned on the silver data (FTMT); a hybrid NLG+MT model which pipelines KG-to-English generation with Machine Translation from English into the target languages (NLG+MT); and few-shot LLM prompting (FEWSHOT).

Results. We examine the performance of the different approaches using automatic metrics, human evaluation and a qualitative analysis. We show that the LLM prompting approach works best overall but that they are marked performance differences between languages. While English, Spanish and French show good results, scores are much lower for Russian and Chinese.

4.2 Data

We first explain how we created the datasets used for fine-tuning, testing and few-shot retrieval. We then describe the three KG-to-Text generation models we investigated.

We create the following three datasets:

- EN-GOLD-1K: a gold test set of 1,000 (graph, English text) pairs. We use this dataset together with multilingual text similarity metrics to select, for each of our target language, the best MT model among 6 high performing MT models.
- 5L-GOLD-100: a gold, manually verified, test set of 100 (graph,text) pairs where each graph is aligned with its verbalisation into our 5 target languages. This will be used to evaluate and compare our 3 approaches.
- 5L-SILVER-30K: a multilingual dataset of 30K (graph,text) pairs where each graph is aligned with its verbalisation into our 5 target languages. This dataset is used both as a retrieval base for few shot selection and as silver training data for fine tuning our machine translation models on the KG-to-Text task.

Selecting MT Models

As mentioned above, we use MT models both to create silver multilingual (graph,text) training data and for the hybrid NLG+MT models. To optimise the quality of both the silver data and the hybrid models, we first seek to select, for each of our target languages, the best available MT model. We do this in several steps as follows.

EN-GOLD-1K. This is a test set of 1,000 KG/English text pairs which we use, together with multilingual text similarity metrics, to select for each of our target languages the best MT model among 6 high performing MT models. We create this dataset by sampling 1K instances from two existing (KG, English Text) gold datasets where the English texts are verbalisations of the graphs. The first dataset is the English WebNLG testset (Gardent et al., 2017b). The second dataset is a dataset derived from KELM (Agarwal et al., 2021) and manually edited to ensure that the text matches the graph, which was created in section 2.3. When sampling from these two datasets, we ensure a balanced distribution of graph sizes (1 to 5 facts) and property diversity with exactly 200 instances for each graph size. This stratified sampling preserves 89.4% of the properties present in the original source dataset, thus capturing a wide range of relational structures.

Translating. We apply several different MT models to translate the English side of each pair into our four target languages: Chinese, French, Russian and Spanish. The used MT models are: Helsinki-NLP (Tiedemann and Thottingal, 2020), M2M100-418M (Fan et al., 2020), NLLB-200-600M (Costa-jussà et al., 2022), mBART-large-50 (Liu et al., 2020), M2M100-1.2B, and NLLB-200-3.3B.

Translation Model Evaluation and Selection. We evaluate the MT models on the EN-GOLD-1K dataset taking the English text as reference and using six multilingual text similarity metrics: NMT_Direct, NMT_Pivot, NMT_Cross, SBERT, BERTScore and Cometwiki. To rank the models, we combine the six metric scores of each MT model for a given language into a single score by normalising the scores and averaging the normalised scores over all metrics.

Let $\mathcal{M}$ be the set of metrics used and S the set of MT models (systems) being compared, then for each target language L and each MT system $s \in \mathcal{S}$, each metric $m \in \mathcal{M}$ produces the normalised score:

$$\widetilde{\text{sc}}_s^m = \frac{\text{sc}_s^m - MIN_S^m}{MAX_S^m - MIN_S^m} \tag{4.1}$$

where sc_s^m is the average score of the MT model s for the metric m on the test set and MIN_S^m (resp. MAX_S^m) is the minimal (resp. maximum) value for metric m across all systems S. For each language, we then average the normalised scores of each model s over all metrics m:

$$\widetilde{\text{sc}}_s = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \widetilde{\text{sc}}_s^m \tag{4.2}$$

Table C.3 in the Appendix presents the results for each model, target language and metrics together with the average normalised scores per model and per language.

For each target language, we select the MT model whose overall performance as captured by the normalised score is the strongest. Table 4.1 shows which model we use for each language as a result of this evaluation.

Model	Chinese	French	Russian	Spanish
Helsinki-NLP		✓		✓
M2M100-418M				
NLLB-200-600M				
mBART-large-50				
M2M100-1.2B	✓		✓	
NLLB-200-3.3B				

Table 4.1: Selected Translation Models for Each Target Language. Models are selected based on their results on the average normalised scores over 6 metrics. Cf. Table C.3.

5L-GOLD-100

To create the multilingual test data necessary to compare our three KG-to-Text approaches, we first collected 50 seen graph-text pairs from publicly available sources: 25 from WebNLG and 25 from the gold data derived from KELM-based benchmarks from section 2.3. These seen pairs are uniformly distributed with graphs of sizes 1 to 5 triples. We then extracted 50 new graphs from Wikidata focusing on properties that were either

unseen or infrequent in WebNLG and KELM. These unseen graphs include 4 categories (Airlines, Chemicals, Conferences, and Statistics) with 10 graphs each that are uniformly distributed with graphs of sizes 3 to 7 triples and a fifth category (Mixed), which combines graphs from the Airline and Statistics category into 10 graphs of 10 triples each. The English lexicalisations of these unseen graphs were obtained with the LLM-based generation approach and were then manually corrected by a native speaker. Finally, we machine translated the English lexicalisations using the best translation model for each language as presented in Table 4.1 and had both the original lexicalisations and the translations corrected by native speakers. Table 4.2 provides the statistics of our test set.

	Size							
Source	1	2	3	4	5	6	7	10
WebNLG	5	5	5	5	5	—	—	—
KELM	5	5	5	5	5	—	—	—
Wikidata	—	—	8	8	8	8	8	10
Total	10	10	18	18	18	8	8	10

Table 4.2: Statistics of the 5L-GOLD-100 testset.

5L-SILVER-30K

We create multilingual, silver (graph,text) data by extracting instances from WebNLG and Kelm and machine translating the English text in each of our 4 other target languages.

KELM Filtering. Agarwal et al. (2021) constructed a corpus of 15M (graph, text) pairs from Wikidata and Wikipedia using distant supervision, fine-tuning and generation. Since the dataset is automatically generated, the alignment between graph and text is imperfect. We therefore filter the dataset in several steps as follows. Using the cross-modal RDF-Text encoder by Le Scao and Gardent (2023), we first filter out all (graph, text) pairs with cosine similarity lower than 0.9. We balance the dataset across graph sizes and Wikidata properties to prevent skew towards frequent properties. This reduces the dataset to 80K (graph, text) pairs.

To further improve data quality, we apply the Data QuestEval (DQE) metric (Scialom et al., 2021), designed for data-to-text tasks and leveraging question generation and question answering models to evaluate the alignment between graphs and texts, and retain only the top-scoring quartile (Q1) of the filtered KELM data based on DQE scores. This final filtering step yields 18,723 high-quality (graph, text) pairs. We call this dataset KELM_{Q1}.

Finally, we combine the 18,723 pairs from KELM_{Q1} with the 39,890 WebNLG instances yielding a total of 58,613 (KG, English text) pairs. We then translate the English texts using the selected MT models identified in section 4.2.

4.3 Models

We compare two supervised approaches with a few-shot LLM-based approach.

4.3.1 Supervised Approaches

Dimensions	Options	Explanation
Input	Graph	The input is the RDF graph
	Prop.	RDF Graph and Wikidata Property Descriptions
	All	RDF Graph, Wikidata Property Descriptions and Entity Descriptions
	P+L	RDF Graph, Wikidata Property Descriptions and Entity Labels
Prompting	Direct	Generate a text verbalising this graph
	Cumulative	Generate a text for each fact in turn then combine the resulting texts into one (two consecutive prompts).
	Chain-of-Thought	Verbalise each fact and aggregate the resultings text into a coherent text (one prompt).
Few-Shot Selection	Random	random examples
	Size-Based	Examples whose graph size is the same as that of the input graph
	Overlap	Examples whose graph maximise property overlap, size and semantic similarity

Table 4.3: Prompting Approaches (example prompts and details about each of these approaches are given in section C.1)

NLG-MT Hybrid Model. The hybrid model (NLG+MT) is a two steps pipeline where a KG-to-Text model is first used to generate English text from the input graph and then machine translation is applied to generate text into the 5 target languages.

For the first step (KG-to-English Generation), we use the ControlPrefixes model (CP) (Clive et al., 2022), a KG-to-Text model which uses dynamic prefix-tuning to allow for the inclusion of input-dependent information thereby combining the benefits of prompt tuning and controlled generation. This model was shown to achieve state-of-the-art results on several data-to-text datasets, including WebNLG[23]. We set the control token for the input graphs to English and train the model for 40 epochs on the WebNLG 2020 English training data.

To machine translate the English generated text, we use the best MT model we identified for each of our target language (Cf. section 4.2).

Fine-Tuned Machine Translation. We fine-tune the machine translation models (FTMT) selected in section 4.2 on the multilingual 5L-SILVER-30K datasets of (graph, text) pairs for each of our target languages for 4 epochs. For each language, we fine-tune the corresponding best MT model using only the corresponding monolingual data.

4.3.2 LLM-based Generation

For LLM-Based generation, we experimented with a few shot prompting strategy that varies across three dimensions: the input, the prompting strategy and how few shots are selected. Table 4.3 summarises the possible options for each dimension. More details about each dimension and examples prompts illustrating the various options for each dimensions are given in section C.9.

To decide on the best approach, we gradually tested and combined the three dimensions eventually exploring 9 configurations (cf. Table C.1 in section C.1). We evaluate and compare the various configurations on the WebNLG testset for Russian. We choose to evaluate on Russian because (i) WebNLG provides a gold (graph,text) dataset for that

[23] https://paperswithcode.com/sota/data-to-text-generation-on-webnlg

Prompt	Generate a lexicalisation of the following graph in French. Include all the information from the triples regardless of their type or relevance.
Graph	("subject": "AmeriGas", "property": "country", "object": "United States")
Labels	"country": "pays", "United States": "États-Unis"
Text	"AmeriGas est aux États-Unis."
Graph	("Western Goals Foundation", "property": "country","object": "United States")
Labels	"country": "pays", "United States": "États-Unis"
Text	

Table 4.4: Example One Shot Prompt (One fact graph, *AmeriGas is located in the United States*). Labels are extracted from Wikidata. The few shot is optimised to contain properties present in the input graph.

language and (ii) we conjecture that using a language other than English allows for a better estimation of how the different prompting options would perform for languages which are further away from the English-centric RDF graphs where entities and properties are English based.

The results of our experiments indicated that the best option was the Direct, Overlap approach with RDF Graph and Wikidata labels as input. We therefore use that approach, generalising it to the 5 target languages[24] and retrieving few-shots from the corresponding training data (gold for English, silver for the other languages).

In essence, our prompting strategy consists in prompting the LLM for a verbalisation of the input graph augmented with Wikidata translations of the entities and properties it contains and using few-shots whose text is in the target language and whose graph maximises property overlap with the input graph. We use the Llama-3.1-8B-Instruct model. An example prompt for French is shown in Table 4.4.

4.4 Automatic Evaluation

4.4.1 Metrics

We evaluate the performance of our models on the 5L-GOLD-100 test set (cf. Table 4.2) using 10 metrics which support our target languages. Six of these metrics are reference-based: BLEU (Papineni et al., 2002a), ChrF (Popović, 2015), TER (Snover et al., 2006), BERTScore (Zhang* et al., 2020) and BLEURT (Sellam et al., 2020). The other 4 do not require to have a reference in the target language: NMTScore-direct, NMTScore-pivot,

[24]For English, we do not retrieve labels since the KG graphs contain the entities and properties English labels.

NMTScore-cross-likelihood (Vamvas and Sennrich, 2022) and COMETKiwi (Rei et al., 2020). As in section 4.2, we also compute a normalised score which aggregates all metrics across multiple systems.

4.4.2 Results

Comparing approaches. Figure 4.1a shows the normalised scores for each language and each model type[25]. We see that the few-shot LLM approach outperforms both the hybrid NLG-MT approach and the direct translation approach across the board.

Conversely, we observe that the direct machine-translation approach performs poorly, particularly for Spanish, French and Russian. This highlights the difficulty of bridging the graph between RDF graphs and NL texts: machine translating a graph into a text, even after fine tuning the MT model on 30K (KG,Text) pairs, remains harder than translating from a text.

Given the small size of the test set, we perform statistical significance analysis between the best-performing approach (few-shot) and the second best-performing approach (NLG+MT). We first test if the normalised scores of the sentences follow a normal distribution with a Shapiro-Wilk test. If any of the scores is not normal we applied the non-parametric Wilcoxon signed-rank test, otherwise we apply a Student's t-test. Table 4.5 shows the p-values of these tests, which confirm the statistical significance in English, Spanish, French and Russian. For the Chinese, a larger test set might be necessary to obtain conclusive results.

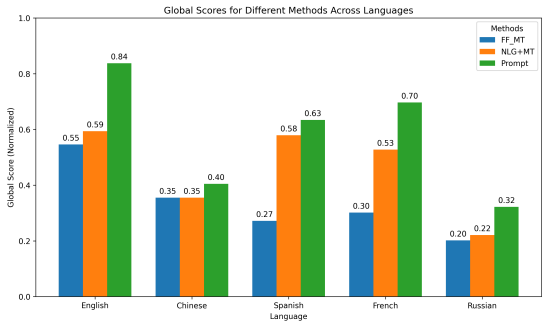
EN*	ZH**	ES**	FR**	RU**
0.0000	0.5092	**0.0114**	**0.0006**	**0.0002**

Table 4.5: P-value of the statistical significance test. *Student's t-test. **Wilcoxon signed-rank test.

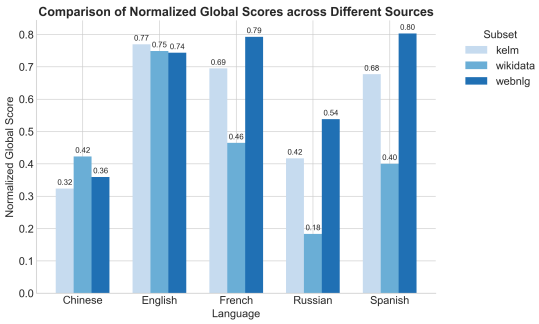
It is also noticeable that while the hybrid NLG+MT approach markedly improves on the MT method for French and Spanish, this is not the case for English, Chinese and Russian. For English, the high scores for both approaches (NLG+MT, FTMT) are likely due to the fact that the RDF graphs are English centric. Hence for this language, bridging the gap between structured (RDF graph) and unstructured (text) data is relatively easy. In contrast, the relatively low results for Russian and Chinese and the similar scores obtained by the NLG+MT and the FTMT approach suggests that for these two languages, whether the input is a graph or an English verbalisation of that graph, the text generated by the MT models remains of low quality overall. This suggest that the MT models for these two languages requires more fine-tuning data for producing good quality KG verbalisations.

Performance on seen vs unseen data. As explained in Table 4.2, our test set comprises three subsets, two subsets derived from existing benchmarks (WebNLG and Kelm) and one subset we created from scratch, extracting subgraphs from Wikidata and manually editing the output generated by the LLM approach. Figure 4.1b shows the performance

[25] Table C.2 provides the detailed results for each metric, language and model type.



(a) Comparing the 3 approaches.



(b) Performance on the different testsets. For each target language, we use the best model, i.e. Few-shot Prompting.

Figure 4.1: Global Min-Max Normalized Scores

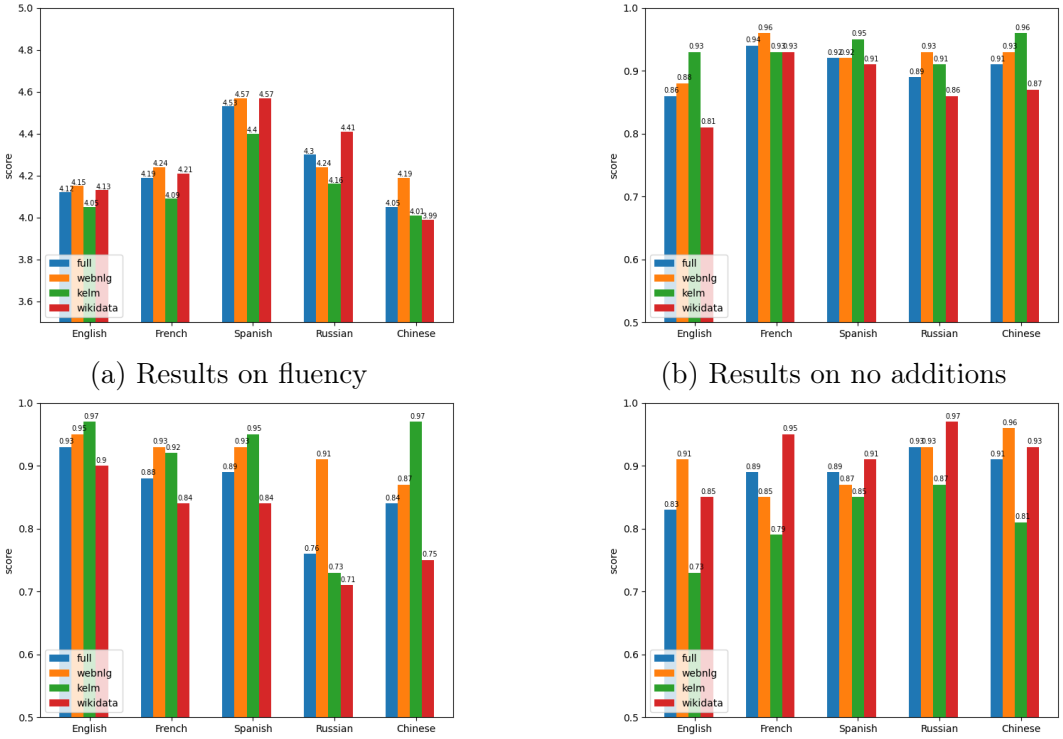
of our best model (FewShot, the LLM based approach) on each of these subsets. We see that the model underperforms on the unseen Wikidata test set for all languages except Chinese, confirming the intuition that LLM-based models perform better on data that is already available on the web and thus is likely to have been included in the LLM training data.

Surprisingly, the results on the WebNLG testset are lower for English than for French and Spanish. However, looking at the scores obtained by surface vs semantic based metrics (Table C.2), we observe that the scores for English are high for surface metrics but low on semantic based metrics indicating an overfitting effect (the model has memorized the WebNLG texts). We conjecture that as the English WebNLG data is available online, the LLM has learned to produce text that is close to the reference and therefore fails to generate paraphrases. As a result, the scores for all paraphrase lenient metrics are lower than for the other languages which in turn, results in a (slightly) lower global normalised score with respect to other languages.

Performance per language. Figure 4.1a shows strong differences between languages on all the approaches. English has the best results, showing stable performance across all test sets (Figure 4.1b) thereby highlighting the higher performance of multilingual LLMs on English.

Compared to Spanish and French, performance markedly decreases for Chinese and Russian for all three models. As noted above, this suggests that MT models for these two languages struggle to verbalise (English verbalisation of) knowledge graphs. For the LLM model, these lower results are likely due to the fact that Llama-8B was not trained on Chinese and Russian. While it was not trained on Spanish either, it was trained on French and Italian and a transfer effect might explain the higher results for that language.

Impact of graph size. Figure C.1 in the appendix shows the impact of graph size on performance. Unsurprisingly, the longer the graph the lower the score.



(a) Results on fluency

(b) Results on no additions

(c) Results on no omissions

(d) Results of no repetitions

Figure 4.2: Human evaluation results on the full test set and on each subset by source. No A – ratio of instances where there is no addition, No O – ratio of instances where there is no omission, No R – ratio of instances where there are no repetitions. Fluency is scaled from 1 to 5.

4.5 Error Analysis

To better understand the differences between languages, subsets and models, we conduct (i) a human evaluation on the output of our FewShot model and (ii) a comparative qualitative analysis of the three models.

4.5.1 Analysing the FewShot Model

We used the WebNLG Challenge 2023 criteria for human evaluation (Cripwell et al., 2023) asking annotators to answer the following four questions (with binary yes/no answer for the first three questions).

No omission. "Looking at each element of the graph in turn, does the text express each of these elements in full (allow synonyms and aggregation)?".

No addition. "Looking at the text, is all of its content expressed in the graph? (Allow duplication of content.)".

Unnecessary repetition. "Is any content in the text unnecessarily repeated?".

Fluency. " Please rate the text shown in terms of fluency on a scale of 1 to 5 where 5 is the highest (best) score. Highly fluent text 'flows well' and is well connected and free from disfluencies.".

To select the annotators we run the test with 15 data-text pairs, which we constructed in English and annotated ourselves. We then had this test set translated into the target languages by native speakers and only retain those annotators which annotated 10 out of 15 instances correctly on omission, addition and repetition. More details on annotator recruitment are given in the section C.6. For absence of addition, omission and repetition, we report the ratio of 'Yes' answers. For fluency, we take the average[26]. Figure 4.2 shows the results (Result tables and Fleiss' kappa for IAA scores are given section C.8).

We see that overall the model for English makes more addition and fewer omission mistakes, which we conjecture is a sign of overfitting as, differently from the other target languages, English (KG, text) data is available on the web. Additions results from a memorisation effect whereby the model learned to identify entities, and generates facts it had memorized about them. Conversely, omissions are fewer as the model has already learned to verbalise Knowledge Graphs into English.

While English is found slightly less fluent than other languages, the difference is small and the kappa for this criteria is low across the board (see section C.8), indicating that fluency is a more subjective notion than addition and omission and so the difference in the human scores between English and the other languages is not necessarily meaningful.

Russian and Chinese show a higher rate of additions and omissions, which is consistent with the automatic evaluation and likely explains the lower automatic scores obtained for these two languages.

Finally, we observe that the model underperforms on the Wikidata non i.i.d. dataset which is consistent with the automatic results and highlights the fact that our LLM does not generalise well to unseen data.

[26]There were three instances, where fluency was not recorded (2 for English and 1 for French), in this case to calculate the average we assumed the missing score to be the average of the available scores for that instance.

4.5.2 Comparing Models

To better understand how the FewShot model differs from the ftmt and the nlg+mt models, we manually inspected 20 instances per model and per language. For each language, we considered 10 instances with low scores and 10 instances with high scores using both a surface based metrics (BLEU) and a semantic one (NMT_{pivot}). For each language, we asked a native speaker to classify the errors into 8 classes: DeGen (degenerate cases i.e., output texts that are disfluent and/or show strong semantic difference with the input content), Om (omitted content), Add (added content), Fluency (the text is correct but lacks fluency), NE/Units (incorrect verbalisation of Named Entities or Units), Gr (the text is ungrammatical), Rep (the text contains unnecessary repetitions) and Prop (a property is incorrectly verbalised). These 8 classes of errors were decided after a preliminary annotation and discussion between native speakers. Table 4.6 shows the results per language and per model. The text below summarises our findings for each language.

While there are differences between languages, the results show that the most frequent errors made by the FewShot models are omissions (this is particularly strong for Chinese and Russian). Other frequent errors found for this model are degenerate output (Chinese, French), additions (Spanish) and some repetitions (Spanish). The ftmt models also frequently omit information but additionally struggle to verbalise properties (except for Spanish). Finally, the nlg+mt model shows strong omissions for Chinese and Russian and a struggle to verbalise both entities and properties. It is also generally less fluent than the other two models which might be due to the fact that the sentence segmentation induced by the KG-to-English model might not be a good fit for these target languages.

We again observe a higher rate of errors for both Chinese and Russian confirming the results of the automatic and the human evaluation.

4.6 Extension of the Experiments

While in previous work we showed the potential of our FewShot approach to deal with unseen data, we do not yet know whether the findings would be similar when the approach is applied to lower-resource languages. It is also important to have more seen and unseen golden test data for a more reliable automatic evaluation. Finally, we wanted to further explore the influence of a size and family of a backbone LLM on FewShot results depending on language. Thus, we further extended the experiments to see how the introduced methods work on more languages, diverse test data, and with larger models of different families.

To cover lower-resource languages, we explored Breton (br), Irish (ga), Maltese (mt), and Welsh (cy), as they all have golden test data, which was presented in 2023 WebNLG shared task (Cripwell et al., 2023).

4.6.1 Data

WebNLG. We collect all WebNLG test sets that were available for our languages. This includes the test sets from the WebNLG 2020 shared task (English, Russian); from the 2023 WebNLG shared task (Breton, Welsh, Irish and Maltese) and a recently published

	F	O	A	Gr	D	NE	Re	Pr
Chinese								
FewShot	2	**26**	2	3	7	4	1	0
ftmt	8	20	0	8	6	12	2	12
nlg+mt	4	17	0	3	5	7	0	6
English								
FewShot	0	0	0	0	0	**3**	0	1
ftmt	**2**	0	**1**	0	**3**	**3**	0	2
nlg+mt	0	**2**	**1**	0	0	0	0	**3**
French								
FewShot	3	**6**	2	0	**2**	1	0	1
ftmt	1	4	3	0	0	5	3	**11**
nlg+mt	9	1	0	0	0	7	1	6
Russian								
FewShot	4	**22**	5	2	0	**13**	3	3
ftmt	4	10	0	0	4	7	0	10
nlg+mt	7	21	0	1	2	10	2	9
Spanish								
FewShot	1	3	**6**	1	1	0	**8**	1
ftmt	0	5	2	0	1	0	0	0
nlg+mt	1	4	1	0	0	8	0	0

Table 4.6: Ratio of error type per language and per model (F(luency), O(mission), A(ddition), Gr(ammar), D(egenerate), NE(ne, units), Re(petition), Pr(operty))

test set for Spanish (Ramón-Ferrer et al., 2025) [27]. These test sets share the same 1,779 KGs (1,102 for Russian). The texts are human-written by crowdworkers, researchers working on the topic (Spanish) or by professional translators. The WebNLG test sets are available on the web and thus likely to be part of the LLMs training data. We use this dataset for an *in-distribution* evaluation of KG-to-Text generation.

5L-GOLD-OOD. To assess our three approaches on *out-of-distribution*, unseen data (i.e., data not present on the internet), we extend the NG_{500} testset created and described in section 3.2.1 to other high-resource languages (English, French, Spanish, Russian and Chinese). To retrieve the lexicalisations we first applied our FewShot approach to generate lexicalisations in English. Then the English lexicalisation were edited by a native English speaker and translated with the best MT model for each of the rest languages. Finally, the translations were post-edited by a native speaker of each language. We did not extend this dataset to low-resource languages as we did not have access to the native speakers necessary to edit the machine translated texts.

9L-SILVER-WebNLG. In our extension experiments we only use WebNLG Challenge 2020 data for English (Castro Ferreira et al., 2020b) for training or few-shot selection, so that the seen test data was well aligned with few-shot or training data. We follow a similar process of MT selection to the one described in section 4.2. However, as not all evaluation metrics were available for each selected low-resource language and in general the MT quality is lower when translating into those languages, we adopt several modifications, which are described in detail in subsection C.4.2. Table 4.7 shows which MT model we use for each freshly introduced language based on this selection. We then translate the WebNLG 2020 English data into all our languages using the best MT model.

Language	H	NLLB	mBART	M2M
Breton	✓			
Irish		✓		
Maltese	✓			
Welsh				✓

Table 4.7: **Selected MT models per language.** Models are selected based on their results on the average normalised scores over 6 metrics (Cf. section C.4). H: Helsinki-NLP, NLLB: NLLB-600M, mBART: mBART-50, M2M: M2M100-1.2B .

4.6.2 Models

The core ideas behind the methods we use do not change.

The NLG+MT approach does not change, we continue to generate first English lexicalisations which we then translate automatically with the selected MT model for each language.

[27] https://zenodo.org/records/15064345

However, some details change for our FTMT method, as the fine-tuning is now performed using different data (9L-SILVER-WebNLG). For each language, we fine-tune the corresponding best MT model for 2 epochs on the corresponding monolingual data.

We take the same configuration of FEWSHOT however, using 9L-SILVER-WebNLG dataset as the source of few-shots. Another difference is that now we also investigate how *model family* (Llama vs. Qwen) and *model size* (7/8B vs. 70/72B) interact with this prompting scheme by comparing four instruction-tuned LLMs: LLAMA-3.1-8B-INSTRUCT, LLAMA-3.1-70B-INSTRUCT, QWEN2.5-7B-INSTRUCT, and QWEN2.5-72B-INSTRUCT. Unless otherwise specified, results reported further for the FEWSHOT setting correspond to the best-performing model among these four. Detailed results for all test sets and all models are given in section C.10.

4.6.3 Results

Language	5L-GOLD-OOD	WebNLG
EN	Llama3-70B	Llama3-70B
FR	Qwen2.5-72B	–
ZH	Qwen2.5-72B	–
ES	Llama3-70B	Llama3-70B
RU	Qwen2.5-72B	Llama3-70B
BR	–	Llama3-70B
GA	–	Llama3-70B
MT	–	Llama3-70B
CY	–	Llama3-70B

Table 4.8: Best-performing *Instruct* LLM for each target language in the few-shot setting. "–" indicates that the language is not present in that test set.

Few-Shot Performance across LLMs. Table 4.8 shows which LLM performs best on which test set. Detailed scores for all LLMs are given in section C.10. Three trends emerge from these scores.

1. *Model size matters*: within the same family the larger variant (70/72B) is always superior to its smaller (7/8B) counterpart, confirming that added capacity is still useful in the KG–to–Text scenario.

2. *Family preference depends on language*: Qwen2.5 dominates on Chinese, French, and, to a lesser extent, Russian, whereas Llama-3.1 excels on English, Spanish, and all low-resource languages. The gap is most pronounced for Chinese, where Qwen's Chinese-centric pre-training corpus gives it a clear advantage (+25.58 BLEU on 5L-GOLD-OOD) over Llama (cf. Table C.18).

3. *Potential data contamination*: for Russian (Table C.21), Llama-3.1 beats Qwen on WEBNLG (27.3 vs 24.9) but is surpassed on the OOD set (26.6 vs 27.8) — see

Model	WebNLG	5L-GOLD-OOD
LLaMA3-8B	**25.6**	23.1
Qwen2.5-7B	24.5	**24.4**
LLaMA3-70B	**27.3**	26.6
Qwen2.5-72B	24.9	**27.8**

Table 4.9: **Few-shot BLEU on Russian.** LLaMA wins on the in-domain WebNLG, while Qwen is stronger on the out-of-domain 5L-GOLD-OOD. The trend holds at both parameter scales (8 B vs. 7 B and 70 B vs. 72 B), suggesting that LLaMA benefited from exposure to the WebNLG Russian data during pre-training.

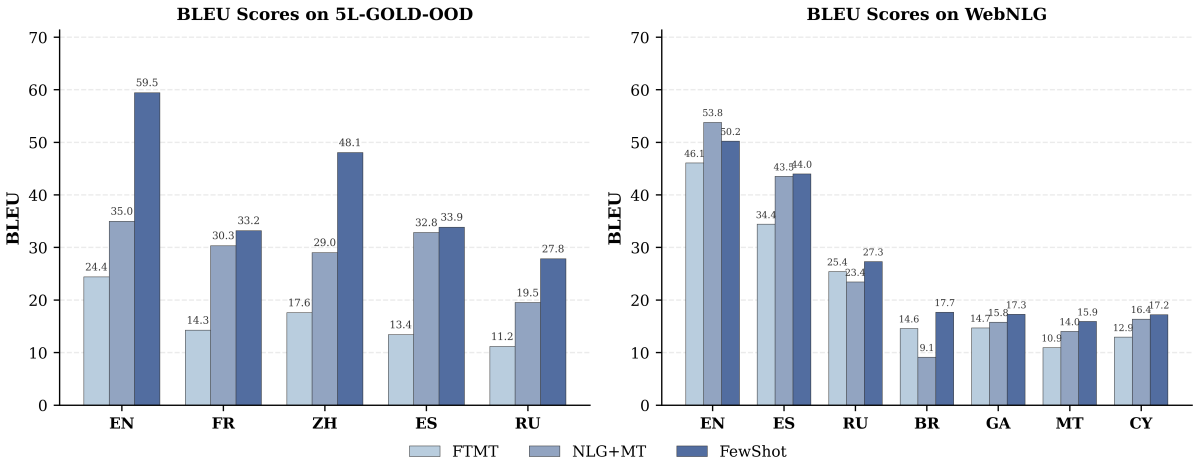


Figure 4.3: **BLEU scores for KG-to-Text Generation on in- (Right) and out-of-domain (Left) data.** Abbreviations: *English (EN)*, *French (FR)*, *Chinese (ZH)*, *Spanish (ES)*, *Russian (RU)*, *Breton (BR)*, *Irish (GA)*, *Maltese (MT)*, and *Welsh (CY)*. FewShot results correspond to the higher score between `Llama-3.1-70B-Instruct` and `Qwen-2.5-72B-Instruct` for each language.

Table 4.9. Given that the WebNLG testset for Russian is publicly available, Llama is likely to have memorised part of them during pre-training, inflating its in-domain score, whereas Qwen generalises better to unseen domains.

Comparing Approaches on In-Distribution. In the in-distribution WebNLG setting (Figure 4.3, right), the FewShot LLM approach outperforms both the hybrid NLG+MT and the direct generation fine-tuned MT method (FTMT) across all evaluated languages except English. English is the one case where the hybrid NLG+MT model slightly surpasses FewShot, which can be attributed to the availability of gold-standard English training data (enabling a directly trained English NLG model). This outcome underscores that, without high-quality parallel data in a target language, prompting a capable LLM with a few examples can still yield superior verbalizations compared to translation-based or weakly fine-tuned solutions. Notably, the FewShot method outperforms both

the NLG+MT and the FTMT approach by a good margin on low-resource languages, indicating a strong benefit from cross-lingual prompting: the LLM leverages knowledge from its multilingual pre-training to generate adequate text even for languages with minimal or no training corpus.

Comparing Approaches on Out-of-Distribution. On out-of-distribution data (5L-GOLD-OOD, Figure 4.3, left) and HR languages, FewShot clearly outperforms the other two methods. Among the 5 HR languages we study, LLM-based generation gains a significant advantage in English and Chinese. This aligns with the language distribution of the LLMs' pre-training corpora. Thus, the LLaMA-based model (used for English) is heavily English-centric[28] while Alibaba's Qwen model (used for Chinese generation) is trained on large Chinese corpora. In other words, the FewShot LLM has a strong advantage on languages it was well-trained on (English, Chinese), whereas in languages less represented in the LLM's training data, the gap between FewShot and the baselines is smaller.

Domain Shift: Seen vs. Unseen Data. The three languages that have both WebNLG (seen) and 5L-GOLD-OOD (unseen) test sets—English, Spanish, and Russian—make the effect of domain shift explicit. Both the NLG+MT pipeline and the FTMT model suffer substantial BLEU drops when moving from the WebNLG domain to the OOD domain, suggesting that these models struggle to verbalize the more complex or specialized triples present in the unseen OOD graphs. Whenever an input graph contains a technically specialised entity or a rare property, these translation-based systems often omit the corresponding triple in the output, trading factual coverage for fluency and thereby incurring substantial evaluation losses (see Table C.21 for an example). The pattern is different for FewShot. In Russian, its BLEU is virtually identical across the two domains, while in English, it is even slightly higher in OOD. The latter "gain" is spurious: As exemplified in Table 4.10, English centric OOD graphs contain many very long entity names, and reproducing them verbatim inflates all reference-based metrics for the English FewShot. FewShot does encounter genuine difficulty on Spanish OOD, where the same entity-omission behaviour emerges and the BLEU drop mirrors that of the translation baselines.

Overall, however, FewShot remains markedly more robust to domain shift than either translation-based approach, particularly on languages (EN, RU) that are well represented in the LLM's pre-training corpus.

High- vs. Low-resource languages. Examining the BLEU scores across different languages in Figure 4.3 (right) reveals a clear trend tied to language resource levels. *High-resource (HR) languages consistently obtain higher BLEU scores than low-resource (LR) languages under all methods.* This indicates that generation quality strongly correlates with the availability of training data and linguistic resources. Even among the HR languages, we see a hierarchy in BLEU performance for the FewShot, the NLG+MT and the

[28]The Llama report indicates that 52.9% of the fine tuning data is from English https://arxiv.org/pdf/2407.21783

FTMT models (English > Spanish > Russian), the degradation of the latter two models reflecting the relative data richness of the corresponding MT models (cf Table C.3). Languages with scarce parallel data or support (the true LR languages) inevitably lag behind in BLEU. *However, the* FEWSHOT *approach narrows the gap between HR and LR languages.* We observe that FEWSHOT prompting yields important gains on lower-resource languages compared to the NLG+MT and FTMT baselines, thereby attenuating the performance disparity. In other words, leveraging a multilingual LLM through cross-lingual few-shot generation helps compensate for the lack of parallel data, elevating the BLEU scores of low-resource languages closer to their high-resource counterparts.

4.7 Conclusion

We explored three ways of extending KG-to-Text generation to multiple languages in the absence of training data: few shot prompting, generate-and-translate and machine translation models fine-tuned on silver graph/text data. For few shot prompting, we compared 9 different ways of prompting and found that a prompting strategy which enriches the input with target language Wikidata labels and target language few-shots selected based on their property overlap with the input graph yields best results. Automatic metrics show that this approach out-performs NLG+MT and FTMT by a large margin. On the downside we note that the performance of the few shot model markedly decreases on unseen data (Wikidata test set) suggesting that the LLM fails to generalise well to non i.i.d data. We also observe marked differences between languages with Russian and Chinese performing poorly with respect to French and Spanish. A qualitative analysis indicates that for Russian named entities are not well verbalised and that for Chinese, Llama-based prompting tends to omit data and often produces degenerate output. In extended experiments we have also shown that the LLM family matters, and Chinese-focused LLM performs and generalises better to unseen data than non-Chinese focused LLM. Finally, we note that while the performance on LR languages markedly decreases for all models, FEWSHOT approach narrows the gap between HR and LR languages.

Input graph	['magnesium oxide | associated hazard | magnesium oxide exposure', 'magnesium oxide | ATC code | A02AA02', 'magnesium oxide | canonical SMILES | O=[Mg]', 'magnesium oxide | InChI | InChI=1S/Mg.O', 'magnesium oxide | chemical formula | MgO', 'magnesium oxide | has use | food additive']
Reference	Magnesium oxide is a compound that has the chemical formula MgO and the InChI InChI=1S/Mg.O. It is associated with the hazard of magnesium oxide exposure and is classified as A02AA02 in the ATC code. The canonical SMILES of magnesium oxide is O=[Mg]. It is primarily used as a food additive.

System	Generated text	BLEU	ChrF	BS	BLEURT	NMT	SBERT
FTMT	Magnesium oxide, which has the ATC code A02AA02, is a food additive. Magnesium oxide, which has the chemical formula MgO, canonical SMILES O=Mg and InChI 1S/Mg.O, is a hazard to the body.	21.3	41.4	92.2	61.2	26.0	90.3
NLG+MT	The ATC code of magnesium oxide is A02AA02 and it has the canonical SMILES O=[Mg]. It has the chemical formula MgO and is used as a food additive.	23.2	45.4	93.1	70.1	31.0	88.5
FewShot	**Magnesium oxide, with the chemical formula MgO, has the associated hazard of magnesium oxide exposure. It has the ATC code A02AA02 and its molecular structure can be represented by the canonical SMILES O=[Mg] and InChI 1S/Mg.O. Magnesium oxide is used as a food additive.**	**41.4**	**64.2**	**94.6**	**76.0**	**40.2**	**92.2**

Table 4.10: **Example illustrating high scores for FewShot on English.** The top block shows the input Knowledge Graph and human reference. The bottom block compares three generation strategies across six automatic metrics. Only the FewShot model succeeds in correctly verbalising all highlighted facts, leading to the highest scores across all metrics. In contrast, both FTMT and NLG+MT either omit or distort key information from the input graph. The long property and entity names, highlighted in orange, correctly verbalised by the FewShot model, result in high scores. (BS refers to BERTScore.)

5

Conclusion

Throughout this thesis, our aim was to improve multilingual KG-to-text generation. We focused a lot on evaluation as a means of understanding the nuances of the approaches and how they can be improved. This final chapter summarizes the contributions as well as states limitations of our work. Based on the limitations we also propose possible future directions of research.

5.1 Main Contributions

We list our contributions by each research question we introduced in introduction.

RQ1: What kind of errors do models produce when evaluated on out of distribution data for KG-to-text generation into non-English languages?

To address this question for Russian we have created KELM and WebNLG based test sets, which are challenging for WebNLG-trained models in terms of new unseen entities, properties and categories. Having run qualitative analysis, we found that Russian models especially struggle with verbalising entities from Knowledge Graphs, as well as to put properties in the right context of those entities (Topic Change error from Chapter 2).

Later, to see how it applies to other languages, we create another Wikidata-based unseen test set, with graphs not yet present in other datasets. Using it we found that for non-English languages, omissions are especially prevalent for an English-centric LLM on out-of-distribution data. We also found that fine-tuned machine translation models generally struggle to verbalise properties whereas the hybrid text-generation and MT approach produce disfluent texts in addition to struggling with verbalising entities and properties.

RQ2: How can we address those errors with different architectures and providing additional information for Russian?

We compared three different approaches to target entity verbalisation and in-context verbalisation of properties for Russian. Specifically, we compared fine-tuning a model with label information containing English-to-Russian translation of entities; several prompting

approaches varying additional information as well as few-shot selection technique; and enhancing LLMs with RAG, providing it with Wikipedia texts context, varying query method as well as chunking method for the context. To properly compare generalisation abilities of those methods we introduce another unseen test set, this time containing graphs not previously present in other datasets, which should have been a bigger challenge for LLM based methods, namely prompting and RAG.

For prompting we find that the best performing variant has a few-shot selection strategy based on similar properties, combined with labels information (possible translations of entities and properties) and providing few-shot examples in the target language rather than English. For RAG, the same few-shot strategy was essential to the best performing variant as well as context retrieved as sentences similar to entities pairs in the graph. We find that both prompting and RAG approve upon fine-tuning in terms of generalisation ability, with prompting tending to be more faithful to the data and RAG producing more fluent output.

RQ3: How do different multilingual approaches compare in KG-to-text generation on out-of-distribution data? Can LLMs facilitate generation into low-resource languages?

We compare three different multilingual approaches to KG-to-text generation: fine-tuning an MT model, a hybrid approach of first using the state-of-the art model for KG-to-English text generation and then translating the outputs with an MT model, and LLM prompting, providing few-shot examples in the target language and label information. To evaluate those approaches we create unseen gold-quality test data for high-resource languages. Using this test set, we show that LLM prompting outperforms other multilingual approaches both on in-distribution and out-of-distribution data. On unseen out-of-distribution data we could also see that LLMs gain special advantage on languages they were trained on. In contrast, we found that translation-based methods suffer substantial evaluation losses when comparing their performance on in-distribution versus out-of-distribution settings.

We have also experimented with our methods applied to low-resource languages. However, for those we only had, in-distribution WebNLG data for test sets. While all methods revealed to be of poorer performance on LR languages, the few shot methods had the least decrease, which indicates that LLMs are a promising way to bridge the gap for LR languages.

5.2 Limitations and Future Research

Our work has several limitations which could be directions for future research.

One of the limitations is KB coverage. We mainly focus on Wikidata and DBPedia, which are both Wikipedia-based Knowledge Bases. While they cover in general diverse domains, they are not specific enough and have their own limitations in terms of knowledge presentation. Future research can address it by either looking for different KB more specialised for their domains, such as ontologies collected for biomedical domain like UMLS (Bodenreider, 2004), or by working on more careful selection of graphs from Wikipedia

KBs, focusing on their informativeness and usability.

In terms of language coverage, while we do cover several different languages from different families, they are still either extremely high-resource languages or of limited diversity (Irish, Breton and Welsh are all Celtic languages, and Maltese is a Semitic language). Future research could extend to incorporate more diverse languages of diverse language families. Additionally, the languages we consider to be low-resource have still reasonable amount of resources available compared to other languages, such as Sami or Udmurt.

In Chapter 4 we draw conclusion for other languages based on one non-English language performance. This raises such questions as what if another prompting strategy or setting can work better on Welsh while performing worse on Russian? Although due to limited resources, such as lack of gold-quality data for some languages, limited time and computational resources, it was not feasible for us to evaluate every single combination on every single language we cover, this issue could be addressed in future research. For example, one might take into account language similarity based on phylogenetic trees to first draw conclusions for a group of languages based on their most high-resource one and then adapt the settings for each language in particular. As Soto Martinez et al. (2024) showed, fine-tuning models using phylogenetic knowledge can help improve performance on lower-resource languages without sacrificing the high-resource languages, so it is an interesting question whether similar method could be adapted to prompting.

While our automatic and human evaluation provides some insights on models' performances, our evaluation often lacks granularity. Precisely, when we are discussing a number of additions or omissions we do not know in what parts of the graph or text they happen and in what specific ways. This could be addressed, for example, by using LLM-as-a-judge, training it to annotate spans with lost or added information and then analysing those spans. As a whole, it is also intriguing to see how LLM-as-a-judge evaluation is adaptable to non-English languages.

As we discussed, KG-to-text generation can be part of many different applications, and thus it is also important to see how our multilingual methods work in the context of those applications.

5.3 Final Remarks

As a result of this thesis we have created several new test sets which could be used to evaluate further KG-to-text generation research. Those test sets are published in a way that prevents LLMs from using them as source of training. Specifically, we have created a website `https://webnlghost.loria.fr/`, where one can download the graphs and upload the generated texts from those graphs, the website then will output the scores for several automatic metrics. We have explored several non-English and multilingual approaches and showed the potential of LLMs for bridging the gap between low-resource and high-resource languages as well as in-distribution and out-of-distribution data. The code for those experiments can be found in `https://github.com/Javanochka/KG-to-text-fine-tuning-prompting-rag.git` for the experiments in Chapter 3 and `https://github.com/MeloS7/Multilingual-KG-Verbalisation.git` for the experiments

in Chapter 4. We have also identified several errors which occur in multilingual generation. We hope that our contributions can help advance multilingual KG-to-Text research.

A

Appendix for Chapter 2

A.1 Human Annotation Details

Payment details. Prolific only allows to set a fixed rate for the task completed by the annotator. So, we first estimated how long it takes to annotate 100 instances and to read all the instruction, and we would set the rates accordingly to get an average of 10.5€ per hour. However, in reality some annotators would be faster than anticipated, so the actual average (total payment divided by total time spent) is about 10.7€ per hour.

A.2 Full Automatic Evaluation Results

In table A.1 we provide all the automatic evaluation results of all the models on all of the test sets for Russian. We can see that for Russian models all metrics favour the Interno model. In table A.2 we provide all the automatic evaluation results of all the models on all of the test sets for English. Only for KELM-E+P we see that different metrics favour different models, however, the difference for TER and BERT metrics is not significant.

As according to table 2.2 KELM-E+P is also balanced regarding the input graph size, we decided to also additionally provide BLEU scores by each graph size for this benchmark for both English and Russian. The graphs in fig. A.1 illustrate those results. We see a well expected drop of performance for all English models with a graph size increase. However, we do not see the same drop for Russian models. This may be due to the overall poorer performance of Russian models on unseen data.

A.3 Error Annotation Examples

We show some of the annotated examples of generated texts in English (table 2.6) and Russian (tables A.3 and A.4). We highlight there the specific issues which led to annotation of the specific error. Note, that normally each generated text may have several problems, in the table we only highlight the one which illustrates the specific error. We also provide translations for Russian examples (except for "garbage" category).

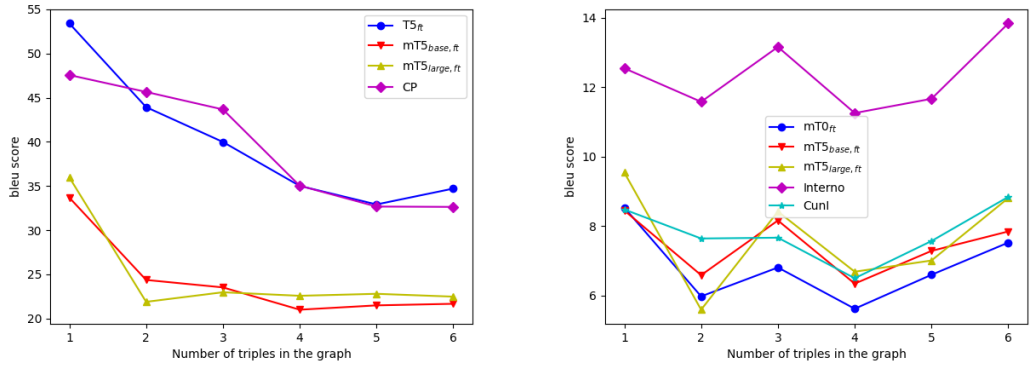


Figure A.1: BLEU scores for different graph sizes in KELM-E+Pfor English (left) and Russian (right)

A.4 Full Qualitative Analysis Results

We present all the ratios of errors and good instances across all models, benchmarks and languages in table A.5. We also show a progression of a ratio of each error for each model on KELM-E+P benchmark for English (fig. A.2) and Russian (fig. A.3).

Model	BLEU ↑	chrF++ ↑	TER ↓	BERT P ↑	BERT R ↑	BERT F1 ↑
ru-WebNLG All						
$mT0_{ft}$	52.227	0.685	0.397	0.915	0.910	0.912
$mT5_{base,ft}$	51.861	0.684	0.393	0.916	0.909	0.911
$mT5_{large,ft}$	51.954	0.686	0.401	0.914	0.908	0.910
Interno	**53.668**	**0.694**	**0.373**	**0.921**	**0.913**	**0.916**
CunI	48.503	0.673	0.439	0.902	0.897	0.898
ru-WebNLG E						
$mT0_{ft}$	17.51	0.457	0.777	0.82	0.821	0.82
$mT5_{base,ft}$	17.212	0.438	0.787	0.814	0.81	0.811
$mT5_{large,ft}$	15.1	0.439	0.87	0.792	0.805	0.798
Interno	**22.427**	**0.486**	**0.649**	**0.858**	**0.851**	**0.854**
CunI	17.016	0.426	0.765	0.804	0.805	0.804
ru-WebNLG C						
$mT0_{ft}$	11.193	0.175	0.931	0.754	0.755	0.753
$mT5_{base,ft}$	11.272	0.177	0.854	0.756	0.755	0.754
$mT5_{large,ft}$	11.289	0.176	0.928	0.746	0.748	0.745
Interno	**23.019**	**0.422**	**0.653**	**0.84**	**0.819**	**0.828**
CunI	11.658	0.289	0.814	0.757	0.758	0.757
KELM-E						
$mT0_{ft}$	11.318	0.311	0.876	0.781	**0.79**	0.785
$mT5_{base,ft}$	10.121	0.296	0.943	0.767	0.781	0.773
$mT5_{large,ft}$	10.989	0.306	0.957	0.761	0.776	0.768
Interno	**17.569**	**0.355**	**0.692**	**0.818**	**0.79**	**0.803**
CunI	10.779	0.307	0.893	0.773	0.788	0.78
KELM-E+P						
$mT0_{ft}$	6.811	0.284	1.314	0.73	0.747	0.738
$mT5_{base,ft}$	7.483	0.284	1.178	0.734	0.749	0.741
$mT5_{large,ft}$	7.768	0.284	1.225	0.723	0.742	0.732
Interno	**12.391**	**0.355**	**1.055**	**0.78**	**0.777**	**0.778**
CunI	7.725	0.294	1.08	0.74	0.756	0.747

Table A.1: **Automatic evaluation results on Russian benchmarks**

Model	BLEU ↑	chrF++ ↑	TER ↓	BERT P ↑	BERT R ↑	BERT F1 ↑
en-WebNLG						
$T5_{ft}$	52.569	0.680	0.411	0.958	0.955	0.956
$mT5_{base,ft}$	44.163	0.627	0.575	0.942	0.941	0.941
$mT5_{large,ft}$	44.019	0.634	0.558	0.942	0.942	0.941
CP	**53.81**	**0.692**	**0.401**	**0.959**	**0.956**	**0.957**
KELM-E						
$T5_{ft}$	**47.554**	**0.764**	**0.346**	**0.96**	**0.967**	**0.963**
$mT5_{base,ft}$	31.694	0.66	0.645	0.938	0.946	0.942
$mT5_{large,ft}$	32.086	0.675	0.541	0.941	0.947	0.944
CP	46.629	0.755	0.368	0.959	0.965	0.962
KELM-E+P						
$T5_{ft}$	**39.037**	**0.685**	0.566	**0.946**	0.955	0.95
$mT5_{base,ft}$	23.794	0.566	1.007	0.913	0.927	0.92
$mT5_{large,ft}$	24.419	0.581	0.928	0.915	0.931	0.923
CP	38.863	0.683	**0.553**	**0.946**	**0.956**	**0.951**

Table A.2: **Automatic evaluation results on English benchmarks**, i.e., WebNLG data (WebNLG: 2020 challenge test set), and two new created benchmarks for text generation from KELM.

Graph	Generated text
Addition	
(*Leipzig-Schönefeld*, *instance of*, *Ortsteil*), (*Leipzig-Schönefeld*, *country*, *Germany*), (*Leipzig-Schönefeld*, *located in area*, *Leipzig*)	Германский город Лейпциг-Шонефельд находится в районе Лейпцига и находится в области Лейпцига. В городе есть Ортстеil. *Translation:* The German city of Leipzig-Schonefeld is located in the district of Leipzig and is in the region of Leipzig. The city has an Ortsteil.
Omission	
(*Hypermarcas*, *industry*, *Pharmaceuticals*), (*Hypermarcas*, *type*, *S.A. (corporation)*), (*Hypermarcas*, *subsidiary*, *Mantecorp*), (*Hypermarcas*, *net income*, *108600000*), (*Hypermarcas*, *number of employees*, *10252*), (*Hypermarcas*, *product*, *Healthcare*)	Компания "Hypermarcas фармацевтическая компания, которая имеет дочернюю компанию "Mantecorp". "Hypermarcas"имеет 10252 сотрудника и годовой доход 108600000. *Translation:* Hypermarcas is a pharmaceutical company which has a subsidiary Mantecorp. Hypermarcas has 10252 employees and has an annual revenue of 108600000.
Repetition	
(*Parliamentary Entitlements Act 1990*, *country*, *Australia*), (*Parliamentary Entitlements Act 1990*, *instance of*, *Act of the Parliament of Australia*), (*Parliamentary Entitlements Act 1990*, *applies to jurisdiction*, *Australia*)	Парламент Австралии принял Парламентский акт 1990, который относится к юриспруденции в Австралии. Он был принят Парламентом Австралии и был опубликован в Акте Парламента Австралии. *Translation:* The Parliament of Australia passed the Parliament Act 1990, which relates to jurisprudence in Australia. It was passed by the Parliament of Australia and was published in an Act of the Parliament of Australia.
Entity distortion	
(*Bedford Aerodrome*, *runway surface type*, *Concrete*)	Тип поверхности взлетно-посадочной полосы в аэропорту "Берддорф" - бетон. *Translation:* The surface type of the runway at Berddorf Airport is concrete.

Table A.3: Error annotation examples for Russian: addition, omission, repetition and entity distortion

Graph	Generated text
Property understanding	
(*Ciudad Ayala*, *government type*, *Council-manager government*), (*Ciudad Ayala*, *population density*, *1604.0*), (*Ciudad Ayala*, *time zone*, *Pacific Daylight Time*), (*Ciudad Ayala*, *time zone*, *Pacific Standard Time Zone*), (*Ciudad Ayala*, *leader*, *Juan Nolasco*)	Лидером Чикаго Аялах является муниципальный правительство, который управляется советом и имеет в качестве лидера Хусейна Нолоссо. Кроме того, в этом районе можно найти длину населенных пунктов 1604,0, а также в настоящее время в этом районе - Международный день-лидер и Период Пасифик. *Translation:* The leader of Chicago Ayalah is a council-managed municipal government that has Hussein Nolosso as its leader. You can also find the length of human settlements is 1604.0 in the area, and currently the International Day Leader and Pacific Period are in the area.
Topic change	
(*Jung Jin-sun*, *height*, *+185 centimetre*)	Высота Юджина Ян-сана составляет -185 метров. *Translation:* The elevation of Eugene Yang-san is -185 metres. *Explanation:* While the word высота in Russian literally means "height", it is not applied to people, only to buildings and geographical objects, like mountains.
Complex text	
(*Uddheden*, *instance of*, *Urban areas in Sweden*), (*Uddheden*, *country*, *Sweden*), (*Uddheden*, *located in area*, *Sunne Municipality*)	Uddheden находится в районе Sunne Municipality в Швеции и является одним из примеров городских районов в Швеции. *Translation:* Uddheden is located in Sunne Municipality in Sweden and is one example of an urban neighbourhood in Sweden.
Garbage	
(*Bootleg Series Volume 1: The Quine Tapes*, *preceded by*, *Squeeze (The Velvet Underground album)*), (*Bootleg Series Volume 1: The Quine Tapes*, *record label*, *Polydor Records*), (*Bootleg Series Volume 1: The Quine Tapes*, *record label*, *Universal Music Group*)	"Пьетро-во-во-во

	#	A	O	R	ED	PU	TC	CT	garbage	good
Results for each model										
CP	30	0.13	0.2	0.3	0.07	0.07	0.0	0.6	0.0	0.2
CunI	40	0.95	0.95	0.25	0.8	0.28	0.45	0.0	0.05	0.0
$T5_{ft}$	30	0.27	0.33	0.03	0.0	0.23	0.03	0.43	0.0	0.23
Interno	40	0.57	0.6	0.32	0.72	0.22	0.1	0.15	0.2	0.0
$mT0_{ft}$	40	0.85	0.85	0.48	0.78	0.2	0.2	0.0	0.15	0.0
$mT5_{base,ft}$	70	0.51	0.51	0.34	0.44	0.23	0.2	0.01	0.47	0.0
$mT5_{base,ft}$ru	40	0.72	0.72	0.42	0.68	0.32	0.32	0.0	0.28	0.0
$mT5_{base,ft}$en	30	0.23	0.23	0.23	0.13	0.1	0.03	0.03	0.73	0.0
$mT5_{large,ft}$	70	0.5	0.49	0.3	0.31	0.13	0.21	0.01	0.49	0.0
$mT5_{large,ft}$ru	40	0.5	0.5	0.32	0.35	0.12	0.38	0.0	0.5	0.0
$mT5_{large,ft}$en	30	0.5	0.47	0.27	0.27	0.13	0.0	0.03	0.47	0.0
Results for each dataset										
KELM-E en	20	0.1	0.1	0.1	0.05	0.1	0.0	0.05	0.4	0.45
KELM-E ru	25	0.52	0.56	0.16	0.56	0.2	0.2	0.0	0.44	0.0
KELM-E+P en	100	0.32	0.35	0.23	0.13	0.14	0.02	0.32	0.28	0.04
KELM-E+P en (0.5, 0.6)	20	0.35	0.35	0.25	0.15	0.25	0.0	0.45	0.1	0.1
KELM-E+P en (0.6, 0.7)	20	0.35	0.4	0.15	0.2	0.0	0.1	0.35	0.2	0.0
KELM-E+P en (0.7, 0.8)	20	0.5	0.5	0.35	0.15	0.25	0.0	0.2	0.25	0.05
KELM-E+P en (0.8, 0.9)	20	0.25	0.35	0.3	0.1	0.15	0.0	0.2	0.4	0.05
KELM-E+P en (0.9, 1.0)	20	0.15	0.15	0.1	0.05	0.05	0.0	0.4	0.45	0.0
KELM-E+P ru	125	0.69	0.68	0.38	0.62	0.25	0.33	0.02	0.28	0.0
KELM-E+P ru (0.5, 0.6)	25	0.88	0.84	0.12	0.88	0.4	0.28	0.04	0.08	0.0
KELM-E+P ru (0.6, 0.7)	25	0.8	0.8	0.44	0.76	0.4	0.36	0.04	0.12	0.0
KELM-E+P ru (0.7, 0.8)	25	0.64	0.64	0.48	0.56	0.24	0.36	0.0	0.36	0.0
KELM-E+P ru (0.8, 0.9)	25	0.52	0.52	0.4	0.48	0.08	0.2	0.04	0.44	0.0
KELM-E+P ru (0.9, 1.0)	25	0.6	0.6	0.44	0.44	0.12	0.44	0.0	0.4	0.0
ru-WebNLG C	25	0.88	0.92	0.76	0.64	0.04	0.36	0.04	0.04	0.0
ru-WebNLG E	25	0.92	0.92	0.08	1.0	0.36	0.12	0.08	0.0	0.0
Results for each language										
English	120	0.28	0.31	0.21	0.12	0.13	0.02	0.28	0.3	0.11
Russian	200	0.72	0.72	0.36	0.66	0.23	0.29	0.03	0.24	0.0

Table A.5: Full results of qualitative analysis. A: addition, O: omission, R: repetition, ED: entity distortion, PU: property understanding error, TC: topic change, CT: complex text. Ratio – yes / (# of instances in the category). The # column – number of instances to annotate in the category

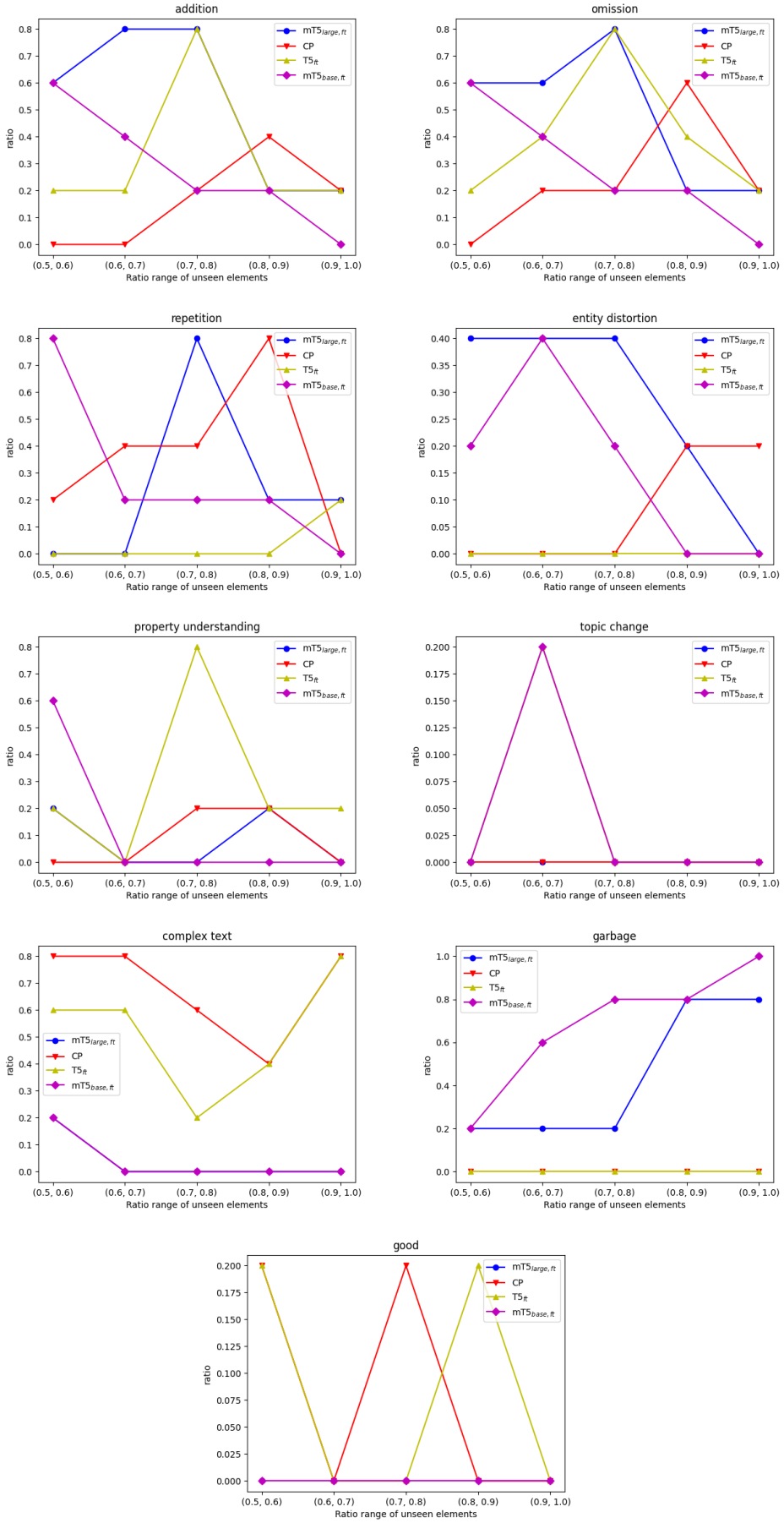


Figure A.2: Each error progression on KELM-E+P dataset for English.

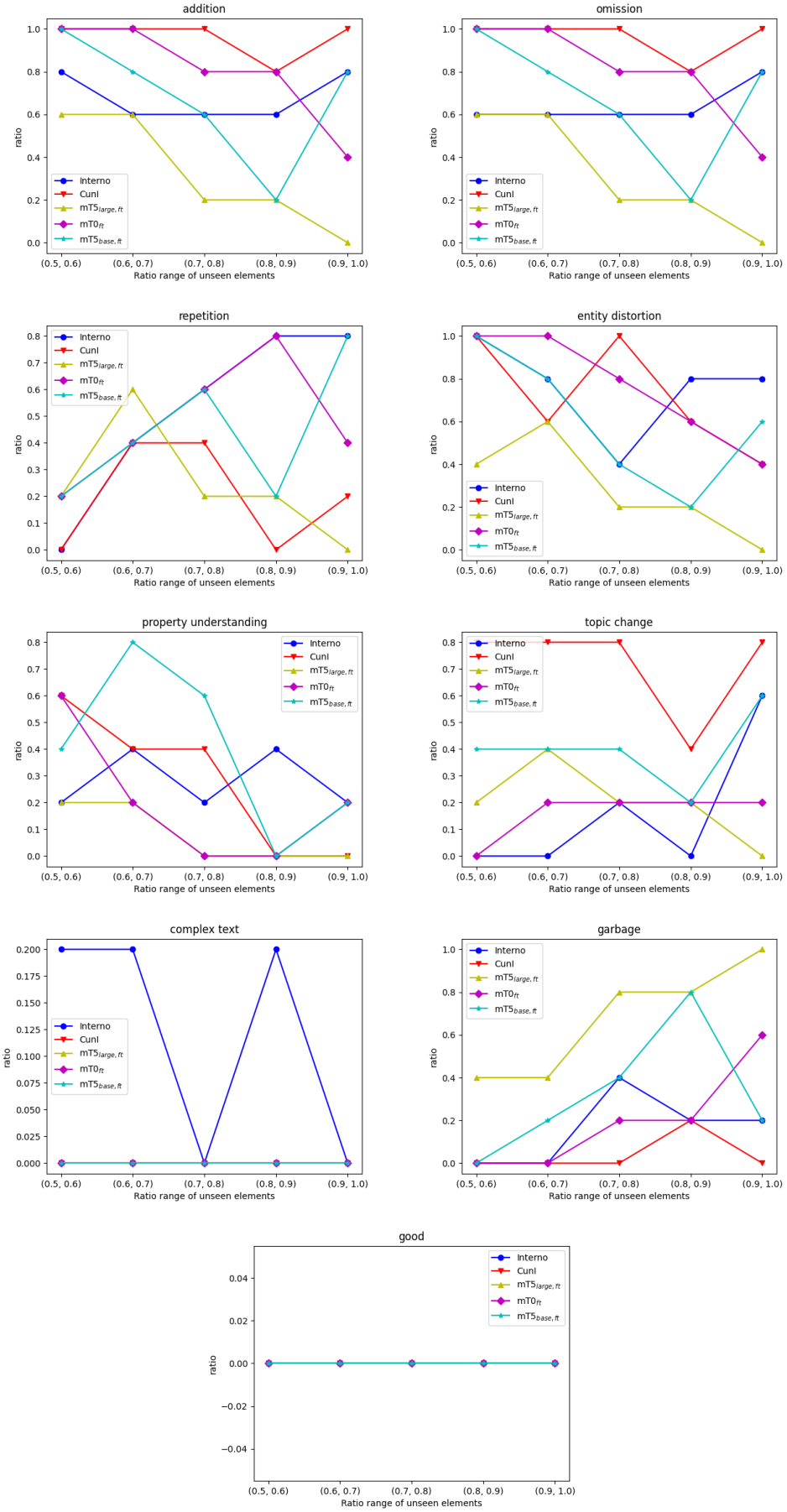


Figure A.3: Each error progression on KELM-E+P dataset for Russian.

B

Appendix for Chapter 3

B.1 Example Prompts

An example prompt for direct generation is provided in the listing B.1. Example prompts for teo steps of cumulative generation are provided on the listings B.2 and B.3. Finally, an example prompt with context information (RAG) is provided in listing B.4.

```
<|begin_of_text|><|start_header_id|>user<|end_header_id|>

{
    "labels": {
        "country": "государство",
        "United States": " США",
        "foundingDate": "дата основания, создания, возникновения"
    },
    "data": [
        {
            "id": 0,
            "subject": "AmeriGas",
            "property": "country",
            "object": "United States"
        },
        {
            "id": 1,
            "subject": "AmeriGas",
            "property": "foundingDate",
            "object": "1959-01-01"
        }
    ]
}

Generate a lexicalisation of all the 2 triples in Russian.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{
    "full-text": "AmeriGas была основана в Соединенных Штатах в 1959-01-01."
}
<|eot_id|><|start_header_id|>user<|end_header_id|>

{
    "labels": {
        "formationDate": "дата основания, создания, возникновения",
        "country": "государство",
        "United States": " США"
    }",
```

```
    "data": [
        {
            "id": 0,
            "subject": "Western Goals Foundation",
            "property": "formationDate",
            "object": "01 January 1979"
        },
        {
            "id": 1,
            "subject": "Western Goals Foundation",
            "property": "country",
            "object": "United States"
        }
    ]
}

Generate a lexicalisation of all the 2 triples in Russian.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Listing B.1: Example of Direct generation using Overlap One-Shot with labels information in Russian

```
<|begin_of_text|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "AmeriGas",
            "property": "foundingDate",
            "object": "1959-01-01"
        }
    ]
}

Generate a lexicalisation of all the 1 triples in Russian.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{
    "full-text": "AmeriGas была основана в 1959-01-01."
}
<|eot_id|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "Western Goals Foundation",
            "property": "formationDate",
            "object": "01 January 1979"
        }
    ]
}

Generate a lexicalisation of all the 1 triples in Russian.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Listing B.2: Example of a Cumulative generation (Step 1, getting a verbalisation for a single triple) for Russian, using Overlap One-Shot

```
<|begin_of_text|><|start_header_id|>user<|end_header_id|>

{
    "sentences": [
        "AmeriGas была основана в 1959-01-01.",
        "AmeriGas находится в США."
    ]
}

Generate a combined paragraph of all the 2 sentences in Russian.
Include all the information from the sentences regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{
    "full-text": "AmeriGas была основана в Соединенных Штатах в 1959-01-01."
}
<|eot_id|><|start_header_id|>user<|end_header_id|>

{
    "sentences": [
        "Western Goals Foundation была основана 1 января 1979.",
        "Western Goals Foundation находится в США."
    ]
}
```

```
Generate a combined paragraph of all the 2 sentences in Russian.
Include all the information from the sentences regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Listing B.3: Example of a Cumulative generation (Step 2, each fact verbalisation aggregation) using Overlap One-Shot Russian

```
<|begin_of_text|><|start_header_id|>user<|end_header_id|>

{
    "context": "Пилот: Алан Шепард. Шпеер, Альберт. 18 ноября 1923 – родился американский
астронавт Алан Бартлет Шепард.",
    "data": [
        {
            "id": 0,
            "subject": "Alan Shepard",
            "property": "occupation",
            "object": "Test pilot"
        },
        {
            "id": 1,
            "subject": "Alan Shepard",
            "property": "birthPlace",
            "object": "New Hampshire"
        },
        {
            "id": 2,
            "subject": "Alan Shepard",
            "property": "birthDate",
            "object": "1923-11-18"
        }
    ],
}

Generate a lexicalisation of all the 3 triples in Russian.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{
    "full-text": "Алан Шепард был летчиком-испытателем, который родился в Нью-Гэмпшире 18 ноября
1923 года."
}
<|eot_id|><|start_header_id|>user<|end_header_id|>

{
    "context": "Не Хайшэн () (13 октября 1964) – китайский космонавт (тайконавт). Военный
летчик.",
    "data": [
        {
            "id": 0,
            "subject": "Nie Haisheng",
            "property": "birthDate",
            "object": "1964-10-13"
        },
        {
            "id": 1,
            "subject": "Nie Haisheng",
            "property": "occupation",
            "object": "Fighter pilot"
        }
    ],

}

Generate a lexicalisation of all the 2 triples in Russian.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Listing B.4: Example of using context information (specifically, entities based RAG with retrieval of one sentence per triple) using Overlap One-Shot in Russian

B.2 Wikidata SparQL Queries

Figures B.1 and B.2 show example SparQL queries for retrieving Wikidata Russian labels of entities and properties.

```
SELECT ?item ?itemLabel WHERE {
    {
        ?item skos:altLabel "Vasilis Pliatsikas"@en .
    } UNION {
        ?item skos:prefLabel "Vasilis Pliatsikas"@en .
    }
    SERVICE wikibase:label {
        bd:serviceParam
        wikibase:language "ru"
    }
}
```

Figure B.1: SparQL query to get labels of entities

```
SELECT ?prop ?propLabel WHERE {
    ?prop rdfs:label "area"@en .
    ?prop wikibase:directClaim ?p .

    SERVICE wikibase:label { bd:serviceParam wikibase:language
       "ru". }
}
```

Figure B.2: SparQL query to get labels of properties

B.3 Automatic Evaluation Results

We provide complete automatic evaluation results for prompting models, all RAG variations present in the main body of the thesis, and some additional experiments that we ran but did not include in the main body of the thesis. The results are shown in tables B.1,B.2,B.3 and B.4.

The naming system of the prompting models is the same as the one in the main body of the thesis. As for RAG models, we first mention the base LLM, then we precise parameters of the RAG itself. The word 'para' stands for retrieval of the first paragraph based on the root entity. Types of query are the same as in the main body of the thesis. Numbers after the type of query are how many blocks are being retrieved with each query. The word 'multi' stands for graph-size chunks. The last part of the name of the model signifies type of few-shots used ('0shots' means no few-shots being used, 'P' stands for few-shots retrieved based on property, 'R' stands for random few-shots, 'S' stands for the few-shots retrieved by similar graph size).

Model	BLEU ↑	chrF++ ↑	TER ↓	BERT P ↑	BERT R ↑	BERT F1 ↑
WebNLG Seen						
PRU/+L/D	32.154	0.572	0.58	0.871	0.868	0.869
PRU/G/D	30.813	0.568	0.594	0.866	0.866	0.865
Llama+para+0shots	2.383	0.202	1.009	0.587	0.706	0.639
Llama+para+P	29.672	0.551	0.619	0.855	0.863	0.858
Mistral+para+P	23.877	0.458	0.715	0.812	0.843	0.825
Llama+para+R	18.092	0.441	0.756	0.809	0.821	0.814
Llama+rag-ents-1+P	30.424	0.566	0.598	0.863	0.865	0.863
Llama+rag-ents-3+R	18.84	0.452	0.732	0.825	0.825	0.824
Qwen+rag-ents-3+R	16.004	0.422	0.795	0.8	0.807	0.802
Mistral+rag-ents-3+R	16.924	0.439	0.764	0.813	0.817	0.814
Llama+rag-ents-3+P	31.033	0.568	0.598	0.864	0.866	0.864
Llama+rag-trps-3+R	17.559	0.442	0.756	0.817	0.818	0.817
Llama+rag-entsep-3+R	18.547	0.452	0.73	0.826	0.824	0.824
Llama+rag-trpsep-3+R	17.712	0.438	0.749	0.817	0.817	0.816
Llama+rag-1tblocks-entsep-3+R	17.635	0.45	0.75	0.82	0.823	0.821
Llama+rag-multi-entsep-3+R	18.421	0.446	0.741	0.821	0.822	0.82
Llama+rag-multi-entsep-7+R	18.328	0.44	0.741	0.819	0.822	0.819
Llama+rag-multi-unionentsep-7+R	18.6	0.448	0.732	0.824	0.824	0.823
Llama+rag-ents-3+0shots	2.534	0.217	1.03	0.59	0.701	0.639
Llama-rag-ents-3+S	19.35	0.459	0.722	0.828	0.828	0.827

Table B.1: **Results of LLMs with different prompting methods and RAG on the seen WebNLG test set**

Model	BLEU ↑	chrF++ ↑	TER ↓	BERT P ↑	BERT R ↑	BERT F1 ↑
WebNLG E						
PRU/+L/D	30.088	0.579	0.582	0.882	0.882	0.881
PRU/G/D	29.764	0.584	0.573	0.88	0.883	0.88
Llama+para+0shots	3.712	0.246	5.414	0.604	0.722	0.656
Llama+para+P	29.671	0.578	0.578	0.882	0.882	0.881
Mistral+para+P	21.614	0.509	1.077	0.838	0.859	0.847
Llama+para+R	25.953	0.549	0.612	0.875	0.872	0.872
Llama+rag-ents-1+P	27.54	0.574	0.6	0.875	0.878	0.875
Llama+rag-ents-3+R	28.819	0.567	0.593	0.876	0.873	0.873
Qwen+rag-ents-3+R	22.47	0.482	0.71	0.841	0.843	0.84
Mistral+rag-ents-3+R	24.447	0.518	0.647	0.858	0.857	0.857
Llama+rag-ents-3+P	31.947	0.579	0.557	0.883	0.886	0.884
Llama+rag-trps-3+R	26.488	0.54	0.615	0.875	0.869	0.87
Llama+rag-entsep-3+R	28.014	0.562	0.626	0.873	0.871	0.871
Llama+rag-trpsep-3+R	28.894	0.556	0.605	0.874	0.867	0.869
Llama+rag-1tblocks-entsep-3+R	28.859	0.571	0.603	0.878	0.874	0.875
Llama+rag-multi-entsep-3+R	26.543	0.559	0.636	0.871	0.873	0.871
Llama+rag-multi-entsep-7+R	27.346	0.553	0.607	0.874	0.872	0.872
Llama+rag-multi-unionentsep-7+R	28.914	0.567	0.579	0.877	0.873	0.874
Llama+rag-ents-3+0shots	3.754	0.292	4.674	0.617	0.729	0.666
Llama+rag-ents-3+S	28.44	0.557	0.592	0.874	0.872	0.872
WebNLG C						
PRU/+L/D	25.757	0.491	0.638	0.85	0.847	0.847
PRU/G/D	24.764	0.476	0.653	0.844	0.842	0.842
Llama+para+0shots	3.354	0.224	1.011	0.585	0.714	0.641
Llama+para+P	24.29	0.475	0.674	0.836	0.844	0.838
Mistral+para+P	20.913	0.429	0.746	0.807	0.83	0.816
Llama+para+R	22.849	0.469	0.693	0.831	0.841	0.834
Llama+rag-ents-1+P	24.577	0.477	0.651	0.844	0.842	0.842
Llama+rag-ents-3+R	24.631	0.478	0.657	0.844	0.841	0.841
Qwen+rag-ents-3+R	21.793	0.453	0.711	0.825	0.826	0.824
Mistral+rag-ents-3+R	22.811	0.464	0.678	0.834	0.833	0.832
Llama+rag-ents-3+P	24.568	0.479	0.655	0.844	0.842	0.841
Llama+rag-trps-3+R	21.778	0.453	0.698	0.83	0.832	0.83
Llama+rag-entsep-3+R	24.004	0.468	0.667	0.839	0.838	0.837
Llama+rag-trpsep-3+R	21.918	0.453	0.707	0.828	0.831	0.828
Llama+rag-1tblocks-entsep-3+R	23.356	0.465	0.675	0.837	0.839	0.837
Llama+rag-multi-entsep-3+R	23.823	0.47	0.67	0.84	0.841	0.839
Llama+rag-multi-entsep-7+R	23.668	0.472	0.67	0.84	0.841	0.839
Llama+rag-multi-unionentsep-7+R	23.292	0.463	0.67	0.837	0.836	0.836
Llama+rag-ents-3+0shots	3.535	0.239	1.012	0.59	0.701	0.638
Llama+rag-ents-3+S	23.41	0.47	0.665	0.841	0.839	0.839

Table B.2: **Results of LLMs with different prompting methods and RAG on WebNLG-based unseen test sets**

Model	BLEU ↑	chrF++ ↑	TER ↓	BERT P ↑	BERT R ↑	BERT F1 ↑
KELM-E						
PRU/+L/D	21.359	0.512	0.609	0.865	0.866	0.865
PRU/G/D	21.54	0.504	0.608	0.864	0.864	0.864
Llama+para+0shots	1.422	0.129	16.022	0.543	0.697	0.61
Llama+para+P	20.678	0.491	0.757	0.855	0.858	0.856
Mistral+para+P	15.053	0.391	1.925	0.804	0.836	0.818
Llama+para+R	19.943	0.491	0.81	0.85	0.858	0.853
Llama+rag-ents-1+P	19.634	0.485	0.67	0.855	0.857	0.856
Llama+rag-ents-3+R	21.408	0.505	0.632	0.859	0.86	0.859
Qwen+rag-ents-3+R	15.634	0.414	0.875	0.818	0.825	0.821
Mistral+rag-ents-3+R	16.64	0.456	0.78	0.833	0.841	0.837
Llama+rag-ents-3+P	19.502	0.482	0.648	0.858	0.858	0.857
Llama+rag-trps-3+R	18.788	0.481	0.865	0.843	0.851	0.846
Llama+rag-entsep-3+R	20.642	0.483	0.769	0.851	0.853	0.852
Llama+rag-trpsep-3+R	18.754	0.474	0.883	0.84	0.847	0.843
Llama+rag-1tblocks-entsep-3+R	20.441	0.484	0.74	0.848	0.852	0.85
Llama+rag-multi-entsep-3+R	20.186	0.491	0.725	0.85	0.853	0.851
Llama+rag-multi-entsep-7+R	20.987	0.49	0.739	0.853	0.854	0.853
Llama+rag-multi-unionentsep-7+R	20.3	0.489	0.709	0.854	0.854	0.854
Llama+rag-ents-3+0shots	1.488	0.135	14.628	0.544	0.682	0.604
Llama-rag-ents-3+S	20.442	0.496	0.632	0.858	0.857	0.857
KELM-E+P						
PRU/+L/D	16.193	0.46	0.867	0.816	0.835	0.825
PRU/G/D	16.509	0.455	0.823	0.823	0.833	0.828
Llama+para+0shots	1.916	0.184	10.076	0.557	0.679	0.611
Llama+para+P	16.038	0.438	1.062	0.812	0.83	0.82
Mistral+para+P	14.541	0.423	1.377	0.795	0.826	0.809
Llama+para+R	15.726	0.448	0.935	0.813	0.83	0.821
Llama+rag-ents-1+P	16.917	0.456	0.807	0.824	0.834	0.829
Llama+rag-ents-3+R	16.599	0.449	0.866	0.823	0.831	0.826
Qwen+rag-ents-3+R	13.587	0.424	0.981	0.797	0.814	0.805
Mistral+rag-ents-3+R	16.342	0.452	0.843	0.821	0.83	0.825
Llama+rag-ents-3+P	17.218	0.458	0.828	0.825	0.834	0.829
Llama+rag-trps-3+R	15.847	0.441	0.918	0.817	0.827	0.821
Llama+rag-entsep-3+R	16.702	0.455	0.843	0.822	0.831	0.826
Llama+rag-trpsep-3+R	15.653	0.446	0.882	0.817	0.828	0.822
Llama+rag-1tblocks-entsep-3+R	16.64	0.452	0.837	0.822	0.83	0.826
Llama+rag-multi-entsep-3+R	15.959	0.446	0.926	0.819	0.83	0.824
Llama+rag-multi-entsep-7+R	16.322	0.45	0.895	0.821	0.832	0.826
Llama+rag-multi-unionentsep-7+R	16.296	0.45	0.867	0.822	0.832	0.826
Llama+rag-ents-3+0shots	2.242	0.212	7.385	0.574	0.689	0.626
Llama+rag-ents-3+S	16.143	0.454	0.839	0.822	0.832	0.826

Table B.3: **Results of LLMs with different prompting methods and RAG on KELM-based test sets**

Model	BLEU ↑	chrF++ ↑	TER ↓	BERT P ↑	BERT R ↑	BERT F1 ↑
			$\mathbf{NG}_{500}$			
PRU/+L/D	20.543	0.437	0.723	0.825	0.82	0.822
PRU/G/D	21.51	0.441	0.711	0.824	0.822	0.822
Llama+para+0shots	6.635	0.275	4.865	0.603	0.72	0.655
Llama+para+P	19.641	0.426	1.012	0.802	0.814	0.807
Llama+para+R	18.508	0.414	0.865	0.806	0.81	0.807
Llama+rag-ents-3+R	19.554	0.425	0.746	0.819	0.813	0.815
Qwen+rag-ents-3+R	20.081	0.438	0.84	0.805	0.815	0.81
Mistral+rag-ents-3+R	16.46	0.407	0.806	0.807	0.805	0.805
Llama+rag-ents-3+P	20.543	0.433	0.753	0.817	0.817	0.816
Llama+rag-ents-3+S	19.928	0.421	0.747	0.818	0.813	0.815
Llama+rag-trps-3+R	17.434	0.398	0.792	0.81	0.803	0.806
Llama+rag-entsep-3+R	19.707	0.424	0.736	0.822	0.814	0.817
Llama+rag-trpsep-3+R	17.617	0.394	0.873	0.813	0.803	0.807
Llama+rag-1tblocks-entsep-3+R	19.336	0.421	0.766	0.819	0.811	0.814
Llama+rag-multi-entsep-3+R	20.035	0.428	0.782	0.819	0.815	0.817
Llama+rag-multi-entsep-7+R	19.409	0.418	0.782	0.818	0.814	0.815
Llama+rag-multi-unionentsep-7+R	18.716	0.409	0.813	0.815	0.81	0.812
Llama+rag-ents-3+0shots	5.848	0.264	4.778	0.598	0.709	0.648

Table B.4: **Results of LLMs with different prompting methods and RAG on $\mathbf{NG}_{500}$ test set**

B.4 Hyperparameters

For all LLMs we set `max_new_tokens` to 1200, `temperature` to 0.7 and `top_p` value equal to 0.7. We run all our experiments on a single GPU Nvidia A40 with 45GB of memory.

C

Appendix for Chapter 4

C.1 Comparing Different Prompting Strategies

Table C.1 shows the performance of different prompting strategies when evaluating on the Russian WebNLG test set.

			Metrics									
FS	**Sel.**	**Input**	**Prtg**	BLEU	ChrF	TER	BERT	BLRT	NMT-D	NMT-P	NMT-C	SBERT	Comet
Baseline													
Rand		Graph	Dir	21.47	45.16	70.09	93.06	70.65	47.33	71.59	54.92	90.16	65.72
Few-shot variants													
Ovlp		Graph	Dir	22.10	45.74	68.06	93.10	69.99	47.96	72.67	56.17	90.43	65.90
Size		Graph	Dir	20.76	44.61	70.14	92.95	69.93	46.87	70.80	54.34	90.09	65.57
Input variants													
Rand		Prop.	Dir	21.51	44.89	70.32	93.08	70.95	47.05	71.21	54.48	90.21	65.13
Rand		All	Dir	19.02	43.78	75.52	92.64	69.34	45.90	72.43	49.94	88.65	63.68
Rand		Labels	Dir	23.53	47.12	66.57	93.58	72.43	50.31	**75.95**	55.22	91.25	65.58
Generation variants													
Rand		Graph	Cml	19.50	43.82	77.64	92.35	65.80	46.36	70.20	50.63	85.47	60.86
Rand		Graph	CoT	20.67	42.38	70.64	92.76	67.41	47.19	71.43	54.71	89.46	65.85
Combined													
Ovlp		P+L	CoT	24.18	47.58	64.41	93.56	71.64	49.98	73.15	56.97	91.76	**66.72**
Ovlp		Labels	Dir	24.13	47.98	65.38	93.57	72.06	47.33	67.40	56.04	91.72	66.12
Ovlp		P+L	Dir	23.85	47.99	65.70	93.56	72.03	47.24	67.57	55.85	91.66	66.18
OvlpRU		Labels	Dir	**29.30**	**53.42**	**60.26**	**94.33**	**75.80**	**51.35**	70.61	**59.30**	**92.82**	66.21

Table C.1: Evaluation results on **Russian WebNLG Test** dataset with various metrics. Abbreviations: FS. = Few-shot, Prtg = Prompting Method, Rand = Random, Ovlp/Size = Property/Size Based Few Shot Selection, Prop. = Property description, All = Property and Entity descriptions, Dir = Direct generation, P+L = Property Description + Labels, Cml = Cumulative generation, CoT = Chain of Thought generation, MET. = METEOR, BERT = BertScore, BLRT = BLEURT, NMT-D = NMT Direct, NMT-P = NMT Pivot, NMT-C = NMT Cross, SBERT = Sentence-BERT, Comet = Comet-Kiwi.

C.2 Performance of the three KG-to-Text models on the 5L-GOLD-100 testset

Table C.2 shows the performance of the three KG-to-Text approaches for all metrics and all languages.

Model	BLEU	ChrF	TER	BS	Bleurt	direct	pivot	cross	SBert	Comet
English										
FTMT	38.28	61.90	53.39	94.92	74.05	48.69	56.26	50.24	93.80	40.19
NLG+MT	39.66	62.47	53.47	95.06	75.02	50.05	56.61	51.80	94.35	40.63
FEWSHOT	**45.62**	**72.83**	**48.41**	**95.63**	**78.54**	**55.25**	**61.92**	**56.41**	**96.52**	**41.07**
Chinese										
FTMT	33.88	**34.77**	61.97	92.71	60.83	50.44	61.41	**52.72**	**89.21**	52.64
NLG+MT	34.09	31.27	63.71	93.13	62.45	50.87	60.90	51.41	88.67	55.14
FEWSHOT	**34.10**	31.28	**61.72**	**93.22**	**65.96**	**51.50**	**62.45**	51.73	88.43	**56.51**
Spanish										
FTMT	29.50	54.42	60.62	93.20	61.67	43.88	57.24	44.34	90.15	49.43
NLG+MT	**34.70**	58.36	**59.33**	94.24	68.70	49.15	64.36	51.24	93.35	54.65
FEWSHOT	30.53	**60.13**	62.80	**94.54**	**69.99**	**50.03**	**67.30**	**51.40**	**94.50**	**60.15**
French										
FTMT	29.90	55.58	62.14	93.25	56.80	48.80	59.73	47.52	90.84	44.11
NLG+MT	**33.09**	56.14	59.96	93.70	62.77	53.11	65.00	53.06	92.64	49.85
FEWSHOT	32.41	**60.30**	**59.41**	**94.11**	**67.47**	**57.39**	**69.54**	**56.62**	**93.78**	**54.67**
Russian										
FTMT	22.43	45.42	64.93	92.84	63.26	40.14	55.12	46.27	90.66	53.73
NLG+MT	**23.44**	44.27	**65.32**	92.73	65.03	40.09	56.39	46.58	90.37	55.25
FEWSHOT	21.37	**45.91**	68.55	**93.03**	**69.19**	**41.55**	**60.37**	**50.00**	**90.96**	**59.43**

Table C.2: All results for all metrics for all languages. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BertScore, Comet = Cometkiwi, direct, pivot and cross stand accordingly for NMT direct, NMT pivot and NMT cross-likelihood metrics.

C.3 Impact of the input Graph Size on the Performance of the FewShot Model

Figure C.1 shows the impact of the Input Graph Size on the performance of the FewShot Model.

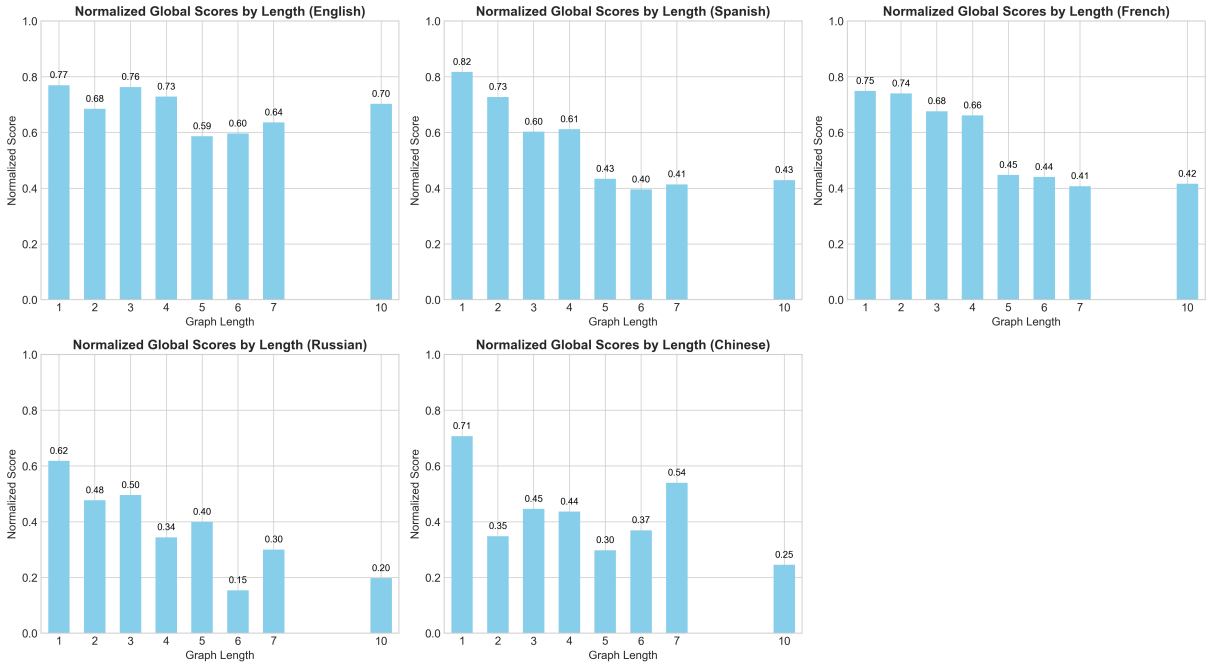


Figure C.1: Global Min-Max Normalized Scores of Subsets by Length.

C.4 Evaluation of 6 MT models on the EN-SILVER-1K testset

C.4.1 High-Resource Languages

Table C.3 shows the scores of the 6 Machine Translation models for all 7 metrics and 4 languages when evaluated on the EN-SILVER-1K (graph, English text) test set using multilingual text similarity metrics.

Lang	Model	NMT_Direct	NMT_Pivot	NMT_Cross	SBert	BERTScore	Cometkiwi	Norm_Row_Avg
EN-ZH	**Helsinki-NLP**	0.634	0.754	0.590	0.8528	0.8984	0.618	0.420
	M2M100-418M	0.719	0.820	0.682	0.8896	0.9039	0.634	0.799
	NLLB-200-600M	0.583	0.712	0.493	0.7876	0.8886	0.591	0.000
	MBart-large-50	0.695	0.784	0.625	0.8705	0.8946	0.687	0.660
	M2M100-1.2B	0.749	0.844	0.695	0.8975	0.9067	0.679	**0.986**
	NLLB-200-3.3B	0.624	0.738	0.525	0.8023	0.8906	0.665	0.270
EN-FR	**Helsinki-NLP**	0.885	0.963	0.891	0.9714	0.9370	0.715	**0.996**
	M2M100-418M	0.848	0.943	0.875	0.9653	0.9372	0.638	0.679
	NLLB-200-600M	0.805	0.887	0.822	0.9501	0.9291	0.612	0.055
	MBart-large-50	0.833	0.915	0.848	0.9654	0.9348	0.667	0.532
	M2M100-1.2B	0.860	0.951	0.884	0.9645	0.9363	0.661	0.756
	NLLB-200-3.3B	0.807	0.879	0.819	0.9473	0.9284	0.631	0.035
EN-RU	**Helsinki-NLP**	0.618	0.748	0.649	0.8559	0.9074	0.582	0.000
	M2M100-418M	0.712	0.863	0.773	0.8978	0.9190	0.631	0.735
	NLLB-200-600M	0.670	0.798	0.696	0.8657	0.9094	0.601	0.259
	MBart-large-50	0.712	0.844	0.738	0.9087	0.9198	0.661	0.771
	M2M100-1.2B	0.756	0.892	0.802	0.9093	0.9216	0.677	**1.000**
	NLLB-200-3.3B	0.694	0.805	0.710	0.8712	0.9102	0.648	0.421
EN-ES	**Helsinki-NLP**	0.884	1.016	0.906	0.9725	0.9434	0.767	**1.000**
	M2M100-418M	0.838	0.956	0.881	0.9659	0.9422	0.682	0.539
	NLLB-200-600M	0.815	0.925	0.851	0.9530	0.9362	0.662	0.094
	MBart-large-50	0.794	0.912	0.837	0.9548	0.9398	0.671	0.113
	M2M100-1.2B	0.853	0.972	0.894	0.9683	0.9422	0.707	0.684
	NLLB-200-3.3B	0.851	0.957	0.882	0.9625	0.9389	0.723	0.527

Table C.3: Results on the 1K English test set for translation models across four target languages (Chinese, French, Russian, Spanish). 'Norm_Row_Avg' is the row-wise normalized average over the other metrics. We use multilingual evaluation metrics to compare the generated text to the English reference.

C.4.2 Low-Resource Languages

For low-resource languages, we adopt the three-stage model–selection protocol outlined below.

1. **Model & metric inventory.** We enumerate all publicly available MT systems and similarity metrics that support the target language l.

2. **Code-switching filter.** To curb the tendency of some systems to produce English instead of l, we run language identification (LID) on their outputs for the EN–GOLD-1K set. Systems whose mean LID confidence for l is below 80% are discarded.

3. **Score normalisation.** For the remaining candidates we compute normalised similarity scores exactly as in the high-resource setting (section C.4).

We then derive a synthetic selection score

$$S(\mathrm{T}, l) \;=\; \mathrm{LID}(\mathrm{T}, l) \times \mathrm{Sim}\big(\mathrm{Text}_{\mathrm{EN}}, \mathrm{T}\big), \tag{C.1}$$

where T is the model output, $\mathrm{LID}(\cdot) \in [0, 1]$ is the confidence from Step 2, and $\mathrm{Sim}(\cdot)$ is the similarity metric chosen in Step 1. For each language we retain the model with the highest $S(\mathrm{T}, l)$.

Language-specific analysis.

- **Breton (br).** Among the candidates only M2M100 provides Breton output, yet its LID precision is less than 60% or even almost zero (Table C.6), indicating severe code-switching. We therefore rely on HELSINKI-NLP, whose 89.52 % LID precision confirms that it produces Breton consistently.
- **Maltese (mt).** All surviving MT systems achieve excellent LID precision ($\geq$99%), but SBERT is the sole evaluation metric available for Maltese (Table C.4). Consequently, we rank MT systems solely by the SBERT similarity and select HELSINKI-NLP.
- **Irish (ga).** Both M2M100 variants score below the 80 % LID threshold (Table C.6) and are discarded. For the remaining three systems (HELSINKI-NLP, NLLB-200-600M, NLLB-200-3.3B) we compute the synthetic score S (Equation C.1) over six metrics. NLLB-200-600M attains the highest S (0.690; Table C.7) and is therefore selected.
- **Welsh (cy).** Both M2M100 and HELSINKI-NLP pass the code-switching filter (LID $\geq$80%). We rank the three viable models (HELSINKI-NLP, M2M100-418M, M2M100-1.2B) via the synthetic score; M2M100-1.2B is the clear winner with $S =$ 1.000 (Table C.7).

The final model selections are summarised in Table 4.1.

Metric	Breton (br)	Irish (ga)	Maltese (mt)	Welsh (cy)
NMTScore (small100)	✓	✓		✓
SBERT (LaBSE)		✓	✓	✓
BERTScore (XLM-R)	✓	✓		✓
COMET-Kiwi	✓	✓		✓

Table C.4: Similarity metric support for selected low-resource languages.

MT Model	Breton (br)	Irish (ga)	Maltese (mt)	Welsh (cy)
nllb-200		✓	✓	✓
mbart-large-50				
m2m100	✓	✓		✓

Table C.5: Support of many-to-many (M2M) MT models for selected low-resource languages.

MT Model	Breton (br)	Irish (ga)	Maltese (mt)	Welsh (cy)
m2m100-1.2B	56.77	65.11	–	98.41
m2m100-418M	0.05	68.07	–	96.27
nllb-200-600M	–	98.24	99.62	0.00
nllb-200-3.3B	–	98.03	99.89	0.00
Helsinki-NLP	89.52	98.19	99.23	88.81

Table C.6: Precision (%) of Language Identification (LID) for generated translations in low-resource languages.

Lang	Model	NMT_Direct	NMT_Pivot	NMT_Cross	SBERT	BERTScore	Cometkiwi	Norm_Syn_Score
EN-GA	**Helsinki-NLP**	0.2118	0.5535	0.2796	0.9169	0.8861	0.5353	0.490
	M2M100-418M	0.5240	0.5493	0.3153	0.3001	0.8096	-0.0174	0.157
	NLLB-200-600M	0.2403	0.6909	0.3265	0.8907	0.8868	0.5388	**0.690**
	MBart-large-50	-	-	-	-	-	-	-
	M2M100-1.2B	0.5989	0.6458	0.5394	0.4911	0.8357	0.1160	0.592
	NLLB-200-3.3B	0.2304	0.6502	0.3135	0.9018	0.8871	0.5804	0.644
EN-CY	**Helsinki-NLP**	0.067	0.146	0.073	0.4652	0.7295	0.067	0.334
	M2M100-418M	0.398	0.483	0.385	0.7713	0.8234	0.250	0.756
	NLLB-200-600M	0.000	0.000	0.000	0.0001	0.0001	0.000	0.000
	MBart-large-50	-	-	-	-	-	-	-
	M2M100-1.2B	0.541	0.673	0.564	0.8580	0.8643	0.454	**1.000**
	NLLB-200-3.3B	0.000	0.000	0.000	0.0001	0.0003	0.000	0.000
EN-MT	**Helsinki-NLP**	-	-	-	0.9308	-	-	-
	M2M100-418M	-	-	-	-	-	-	-
	NLLB-200-600M	-	-	-	0.9040	-	-	-
	MBart-large-50	-	-	-	-	-	-	-
	M2M100-1.2B	-	-	-	-	-	-	-
	NLLB-200-3.3B	-	-	-	0.9194	-	-	-

Table C.7: Synthetic Score Results on the EN-GOLD-1K English test set for translation models across three target languages (Irish (GA), Welsh (CY)), Maltese (MT). 'Norm_Syn_Score' is the row-wise normalized synthetic selection score over the other metrics. We use multilingual evaluation metrics to compare the generated text to the English reference.

C.5 Evaluation Results

Table C.17, Table C.21, Table C.18, Table C.20 and Table C.19 show the results of the three KG-to-Text models on each of the three subsets of the 5L-GOLD-100 testset.

C.5.1 English Subsets Results

Model	Subset	BLEU	ChrF	TER	BS	BLRT	Direct	Pivot	Cross	SBert	Comet
FTMT	kelm	44.03	**70.79**	45.81	**95.52**	55.15	**56.01**	61.73	**59.55**	**95.34**	37.05
	wikidata	30.55	59.51	61.35	94.51	41.01	43.56	49.91	43.77	93.06	40.10
	webnlg	**48.00**	67.77	**45.04**	95.16	**63.54**	51.61	**63.49**	53.85	93.76	**43.48**
	Length 1	**62.23**	**81.13**	**23.34**	**96.89**	**70.68**	**71.40**	**76.41**	**74.27**	**95.76**	32.69
	Length 2	37.77	63.68	46.78	95.15	65.79	53.97	58.11	56.39	95.07	48.08
	Length 3	37.09	64.99	53.50	94.71	57.97	51.23	61.33	54.53	93.69	41.92
	Length 4	36.14	64.89	59.89	95.36	52.27	48.08	56.37	49.57	95.15	**44.73**
	Length 5	40.70	61.13	52.88	94.53	43.23	43.74	52.32	46.20	94.10	34.07
	Length 6	32.99	60.55	62.72	94.22	35.28	41.39	51.62	40.59	91.34	40.16
	Length 7	32.70	61.05	61.36	94.18	41.21	41.80	45.63	41.88	89.91	40.13
	Length 10	25.22	58.02	65.21	94.18	27.88	37.46	44.21	35.18	92.90	39.57
NLG+MT	kelm	41.16	**71.82**	48.48	**95.91**	78.48	**56.26**	60.57	**58.66**	**96.43**	37.06
	wikidata	34.22	60.39	60.53	94.36	71.20	45.56	51.79	46.24	92.40	39.31
	webnlg	**49.05**	69.70	**44.34**	95.60	**79.22**	52.81	**62.30**	56.05	96.18	**46.83**
	Length 1	**60.63**	**82.76**	**29.94**	**97.57**	**83.64**	**70.61**	**72.57**	**71.84**	97.08	37.97
	Length 2	42.35	66.07	41.57	96.06	79.92	56.77	59.89	58.50	**97.19**	**50.48**
	Length 3	37.72	69.31	49.88	95.55	80.25	54.29	60.74	55.93	96.26	46.70
	Length 4	37.68	66.85	54.69	95.61	77.78	52.58	61.53	55.12	95.41	44.51
	Length 5	36.63	62.51	62.23	94.16	71.40	41.89	53.27	45.46	93.80	35.58
	Length 6	34.52	56.33	59.67	93.46	70.15	39.74	49.89	41.54	91.20	35.70
	Length 7	36.44	57.15	63.69	94.20	69.20	46.02	47.85	45.72	92.00	37.61
	Length 10	35.24	58.50	64.28	93.25	62.24	36.71	39.49	36.08	88.88	30.96
FewShot	kelm	38.84	71.27	48.00	**95.79**	**79.16**	**57.34**	**61.80**	**59.54**	**97.18**	41.08
	wikidata	**49.82**	**74.41**	48.82	95.67	78.12	55.74	62.04	55.78	96.36	36.11
	webnlg	44.02	66.80	**47.99**	95.38	78.79	52.15	61.79	54.52	96.18	**50.97**
	Length 1	43.49	72.61	**39.00**	**96.30**	80.51	**62.66**	**66.47**	**64.49**	95.95	35.90
	Length 2	37.94	64.57	47.75	95.43	79.17	54.97	58.78	57.64	97.07	**55.40**
	Length 3	46.85	73.93	42.56	95.79	**81.68**	59.83	63.69	61.38	97.22	41.72
	Length 4	45.84	72.38	46.99	96.15	80.15	56.36	63.12	57.67	**97.50**	42.12
	Length 5	42.75	69.59	54.70	94.96	75.92	48.01	60.33	50.59	96.24	37.01
	Length 6	46.03	71.89	55.09	95.36	75.99	50.77	57.62	51.47	94.23	37.56
	Length 7	51.05	74.19	58.57	95.23	77.33	53.87	59.47	53.35	95.10	39.16
	Length 10	**53.36**	**74.57**	46.76	95.64	75.16	55.53	63.38	52.72	97.00	40.50

Table C.8: Subset results for English categorized by dataset source and graph length, with all metrics. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, Direct = NMT direct, Pivot = NMT pivot, Cross = NMT cross, Comet = Cometkiwi. All values are multiplied by 100 for better readability.

C.5.2 Chinese Subsets Results

Model	Subset	BLEU	ChrF	TER	BS	BLRT	Direct	Pivot	Cross	SBert	Comet
FTMT	kelm	**47.37**	**48.29**	**48.11**	**95.08**	**73.51**	**67.11**	**75.28**	**69.88**	**91.18**	52.65
	wikidata	23.94	24.56	72.58	91.18	51.63	38.32	50.92	41.14	87.90	50.50
	webnlg	40.25	41.67	54.59	93.40	66.54	58.00	68.52	58.71	89.89	**56.93**
	Length 1	**47.12**	**53.26**	**38.29**	**95.00**	**78.78**	**75.20**	**79.89**	**76.37**	**93.20**	56.18
	Length 2	42.90	45.60	51.55	94.87	71.90	63.66	71.34	64.67	88.51	**61.81**
	Length 3	36.19	35.35	62.79	93.08	64.94	50.61	66.14	56.02	89.71	53.78
	Length 4	41.90	40.55	52.52	94.23	68.04	58.31	68.10	60.76	91.93	57.61
	Length 5	29.57	28.91	68.39	91.92	54.25	46.13	57.24	47.39	89.01	49.09
	Length 6	22.14	19.36	79.43	89.61	51.50	36.10	47.16	38.08	84.79	46.61
	Length 7	14.68	24.87	78.23	90.49	46.62	31.88	43.46	32.83	85.35	44.70
	Length 10	25.50	24.80	73.04	90.52	42.09	32.07	45.71	33.91	87.15	46.50
NLG+MT	kelm	**43.41**	**41.14**	**56.72**	**94.59**	**71.36**	**62.21**	**68.93**	**60.34**	89.47	54.95
	wikidata	26.73	26.14	69.88	92.33	55.54	42.30	53.73	44.20	87.74	53.32
	webnlg	39.49	40.72	58.37	93.28	67.35	56.67	67.20	56.88	**89.73**	**58.99**
	Length 1	49.77	45.14	**41.34**	95.09	**78.21**	**73.53**	**76.87**	**71.62**	89.90	53.68
	Length 2	**54.14**	**47.70**	46.46	**95.71**	71.79	66.97	69.21	64.07	90.70	**60.94**
	Length 3	32.73	33.98	66.39	92.94	66.90	51.27	65.52	53.19	**91.27**	60.32
	Length 4	36.10	35.81	58.15	93.96	66.26	54.09	64.69	55.22	91.20	57.88
	Length 5	25.56	26.71	77.20	91.96	58.47	47.05	60.24	47.71	88.24	52.72
	Length 6	23.95	20.85	75.23	92.04	50.42	35.17	44.56	35.82	85.87	53.11
	Length 7	26.96	31.25	67.48	92.05	55.26	40.66	50.47	44.50	88.34	51.55
	Length 10	26.38	27.11	71.98	91.31	44.99	33.20	44.03	33.12	79.42	45.43
FewShot	kelm	31.91	29.70	67.79	93.11	67.74	52.82	**63.18**	51.89	84.61	53.73
	wikidata	34.09	**33.42**	**57.27**	**93.64**	63.51	49.56	61.97	**52.96**	**91.50**	56.34
	webnlg	**36.32**	28.59	64.54	92.49	**69.09**	**54.06**	62.67	49.10	86.12	**59.64**
	Length 1	**46.26**	**41.83**	**46.45**	**94.84**	**80.73**	**73.07**	**81.05**	**63.97**	84.15	55.15
	Length 2	28.90	25.45	66.09	92.62	65.07	54.92	62.23	50.96	81.96	**65.10**
	Length 3	35.69	32.67	62.72	93.52	69.53	49.27	61.09	52.93	91.05	55.31
	Length 4	35.66	28.41	61.09	93.30	65.53	52.22	64.07	53.85	88.52	59.50
	Length 5	31.75	27.02	68.32	92.74	64.37	43.86	53.43	44.06	89.65	52.96
	Length 6	28.77	37.69	60.13	93.07	60.79	48.85	62.64	51.07	85.89	54.95
	Length 7	36.53	38.67	58.50	94.59	65.27	53.28	66.92	57.03	**93.56**	58.03
	Length 10	28.07	25.91	63.87	91.40	54.01	43.69	56.11	44.35	90.04	52.52

Table C.9: Results for Chinese subsets, categorized by dataset source and graph length, with all metrics. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = BleuRT, Direct = NMT direct, Pivot = NMT pivot, Cross = NMT cross, Comet = Cometkiwi. All values are multiplied by 100 for better readability.

C.5.3 Spanish Subsets Results

Model	Subset	BLEU	ChrF	TER	BS	BLRT	Direct	Pivot	Cross	SBert	Comet
FTMT	kelm	36.14	63.62	48.30	**95.17**	**75.40**	**58.11**	67.66	**63.57**	**95.38**	**57.38**
	wikidata	19.85	44.41	73.22	91.48	48.80	31.45	46.18	28.16	85.77	41.67
	webnlg	**42.18**	**65.22**	**47.73**	94.67	73.69	54.51	**68.92**	57.47	93.67	57.00
	Length 1	**52.82**	**78.16**	**31.53**	**96.80**	**82.12**	**66.65**	**70.44**	**66.41**	93.26	44.47
	Length 2	27.76	53.90	51.96	94.36	79.12	57.48	61.00	58.84	**94.54**	**73.88**
	Length 3	30.95	60.24	59.84	94.01	68.00	48.61	58.89	53.19	93.39	55.50
	Length 4	30.02	55.01	58.08	93.77	64.26	46.88	64.55	49.36	92.19	54.53
	Length 5	26.27	50.80	65.35	92.50	57.50	39.05	58.72	41.20	91.46	47.16
	Length 6	19.47	40.38	74.98	90.12	49.64	27.02	38.34	23.29	77.69	37.06
	Length 7	27.33	46.80	73.98	92.01	45.65	28.55	42.61	22.49	81.60	38.31
	Length 10	19.98	43.48	73.61	90.64	37.68	28.01	48.27	22.79	87.58	32.70
NLG+MT	kelm	34.91	64.47	53.94	**95.50**	**74.47**	**58.28**	66.89	**61.19**	95.36	54.95
	wikidata	31.71	56.25	66.06	93.44	63.18	42.06	60.39	42.15	91.32	51.09
	webnlg	**40.46**	**64.97**	**51.26**	94.57	73.96	54.20	**69.78**	59.47	**95.39**	**61.46**
	Length 1	**53.83**	**75.80**	**38.84**	**96.93**	**84.41**	**70.12**	**73.28**	**70.82**	**96.41**	49.90
	Length 2	38.11	63.46	44.78	95.59	77.48	61.51	68.26	66.53	96.31	**68.07**
	Length 3	32.92	62.01	57.43	94.35	74.84	52.73	65.85	58.40	95.33	64.81
	Length 4	31.07	59.81	60.49	94.47	70.53	50.48	67.42	52.14	93.62	59.28
	Length 5	28.81	56.73	69.34	93.36	60.98	41.97	63.66	44.96	92.44	50.68
	Length 6	34.21	55.69	64.75	93.46	64.53	38.85	54.35	35.35	92.27	51.00
	Length 7	35.72	58.44	65.72	94.00	65.92	42.12	64.44	39.59	91.39	47.40
	Length 10	32.05	52.87	68.23	91.97	49.29	33.76	52.54	35.17	87.31	35.22
FewShot	kelm	26.40	60.65	61.63	**95.03**	74.35	54.94	66.10	**60.39**	**94.66**	**63.11**
	wikidata	26.94	57.21	68.41	94.09	64.40	44.30	65.80	43.72	94.36	57.20
	webnlg	**41.83**	**65.45**	**52.76**	94.95	**76.82**	**56.59**	**71.52**	57.79	94.60	63.09
	Length 1	**43.68**	**71.28**	**43.89**	**96.14**	**80.77**	**67.31**	71.95	**68.26**	94.14	51.02
	Length 2	31.74	56.78	49.99	95.55	79.83	58.16	64.51	60.94	94.95	**77.84**
	Length 3	26.93	60.86	59.01	94.35	76.36	52.46	64.47	56.19	94.89	63.95
	Length 4	29.71	62.24	62.50	94.95	72.96	51.84	69.18	54.50	94.48	65.19
	Length 5	27.11	55.68	74.63	93.73	64.33	44.31	68.43	46.79	93.33	56.13
	Length 6	28.82	58.36	71.05	93.96	63.58	39.30	57.34	36.27	94.78	60.67
	Length 7	34.83	59.56	69.34	94.32	59.34	41.11	63.33	36.92	94.53	52.36
	Length 10	28.21	57.09	68.79	93.58	56.41	43.03	**76.30**	42.81	**95.56**	48.73

Table C.10: Results for Spanish subsets, categorized by dataset source and graph length, with all metrics. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, Direct = NMT direct, Pivot = NMT pivot, Cross = NMT cross, Comet = Cometkiwi. All values are multiplied by 100 for better readability.

C.5.4 French Subsets Results

Model	Subset	BLEU	ChrF	TER	BS	BLRT	Direct	Pivot	Cross	SBert	Comet
FTMT	kelm	34.90	62.92	52.96	**94.99**	**71.75**	**61.49**	71.27	**64.99**	**93.54**	**52.53**
	wikidata	20.93	47.69	75.48	91.80	43.18	38.75	47.70	33.99	88.50	36.15
	webnlg	**42.81**	**64.02**	**44.62**	94.43	69.09	56.19	**72.26**	57.12	92.83	51.60
	Length 1	**40.83**	64.47	47.95	94.28	73.47	58.18	63.73	59.55	89.78	45.76
	Length 2	34.87	**65.17**	**45.29**	**95.86**	**80.92**	**68.32**	**78.69**	**67.36**	**96.91**	**71.63**
	Length 3	27.40	57.06	64.25	93.43	61.79	53.44	65.11	54.74	92.75	52.84
	Length 4	36.52	58.87	56.19	94.36	61.95	52.14	60.88	50.64	91.78	41.86
	Length 5	29.53	53.43	63.75	93.15	54.28	46.72	63.31	46.31	92.00	41.06
	Length 6	19.44	45.96	78.05	91.44	37.03	33.66	45.50	30.09	85.26	34.65
	Length 7	22.33	46.75	77.35	90.99	37.15	36.79	51.33	31.01	82.02	29.98
	Length 10	21.63	47.15	72.26	90.76	33.81	30.95	36.72	26.40	90.17	27.65
NLG+MT	kelm	34.21	62.32	56.40	**94.40**	**70.84**	**60.53**	68.43	**61.84**	**94.57**	51.05
	wikidata	28.11	54.51	66.75	93.09	55.29	47.91	59.27	46.73	91.65	46.57
	webnlg	**41.95**	**63.15**	**49.95**	94.22	69.69	56.09	**73.01**	56.96	92.71	**55.21**
	Length 1	**51.85**	**70.27**	41.51	94.81	78.92	64.49	**73.05**	**66.59**	91.05	47.73
	Length 2	40.11	65.57	**41.05**	**95.62**	**78.99**	**66.99**	72.72	62.22	**96.84**	**67.18**
	Length 3	29.65	57.79	61.06	93.71	67.42	57.64	69.88	57.41	94.41	54.73
	Length 4	33.50	62.24	60.55	94.56	68.18	57.63	67.16	59.26	93.85	55.90
	Length 5	28.35	55.86	65.25	93.00	55.55	48.20	69.23	49.12	93.23	41.48
	Length 6	26.75	51.22	65.83	92.76	52.61	41.50	56.30	41.73	91.44	47.03
	Length 7	27.80	52.70	74.12	93.14	55.03	46.39	57.14	46.53	90.78	46.56
	Length 10	30.62	50.63	68.78	91.57	39.69	35.05	42.17	32.80	86.11	34.95
FewShot	kelm	30.35	62.51	60.29	94.23	72.68	60.66	72.72	**62.81**	93.54	55.45
	wikidata	29.61	56.79	62.30	93.90	62.25	53.53	64.30	51.04	**94.13**	52.79
	webnlg	**40.07**	**65.11**	**52.75**	**94.42**	**72.71**	**61.86**	**76.85**	61.58	93.33	**57.65**
	Length 1	**42.95**	**72.42**	**45.81**	94.44	78.10	**69.12**	**74.64**	**66.42**	91.84	44.99
	Length 2	36.31	61.34	54.53	94.51	**79.77**	64.76	70.67	61.58	95.30	**70.90**
	Length 3	30.00	60.40	55.35	94.48	72.10	61.65	74.55	63.59	94.67	61.39
	Length 4	33.39	62.16	58.90	**94.72**	71.39	60.56	72.26	61.29	**95.86**	56.51
	Length 5	27.53	56.55	67.46	93.60	61.12	50.48	71.15	51.58	91.32	48.12
	Length 6	27.10	54.30	60.90	93.65	58.18	49.90	67.68	44.80	92.12	56.07
	Length 7	28.63	55.48	70.74	93.51	58.60	50.78	59.20	47.69	93.41	53.31
	Length 10	36.62	58.98	61.36	93.40	55.17	48.68	56.31	46.58	94.92	44.48

Table C.11: Results for French subsets, categorized by dataset source and graph length, with all metrics. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, Direct = NMT direct, Pivot = NMT pivot, Cross = NMT cross, Comet = Cometkiwi. All values are multiplied by 100 for better readability.

C.5.5 Russian Subsets Results

Model	Subset	BLEU	ChrF	TER	BS	BLRT	Direct	Pivot	Cross	SBert	Comet
FTMT	kelm	**31.79**	**56.89**	**50.75**	**94.52**	**71.70**	**52.60**	**69.60**	**63.83**	**91.14**	**60.20**
	wikidata	14.22	36.34	75.55	91.63	55.96	30.96	43.96	34.47	90.94	49.50
	webnlg	29.48	52.08	57.88	93.58	69.41	46.06	62.94	52.28	89.64	55.72
	Length 1	**36.44**	**58.41**	**40.78**	**94.47**	**79.52**	**57.26**	**70.61**	**66.42**	89.18	57.03
	Length 2	23.36	48.39	58.46	94.06	69.31	45.39	59.58	51.77	86.92	**70.09**
	Length 3	24.07	46.16	63.71	93.05	67.43	43.97	56.32	51.52	91.15	58.14
	Length 4	23.36	49.12	66.53	93.79	66.21	44.04	61.76	51.88	91.78	57.15
	Length 5	21.51	46.14	67.47	92.86	60.95	39.82	58.23	46.21	**92.52**	48.11
	Length 6	17.43	36.36	73.45	91.53	54.21	30.70	38.99	33.29	90.26	48.46
	Length 7	15.21	35.28	73.53	91.28	52.28	27.34	44.57	28.56	88.89	46.53
	Length 10	14.28	35.51	76.65	90.19	48.28	22.26	36.76	25.69	91.42	40.06
NLG+MT	kelm	26.07	**52.96**	60.11	**94.03**	**69.12**	**46.79**	61.66	**57.03**	**91.27**	**58.18**
	wikidata	19.27	41.18	71.50	92.19	61.98	34.93	49.86	39.39	90.54	52.80
	webnlg	**29.14**	51.26	**58.17**	92.50	67.05	43.70	**64.18**	50.50	89.13	57.23
	Length 1	**40.31**	**62.24**	**47.19**	**94.41**	**79.71**	**51.91**	64.08	**61.10**	87.31	55.01
	Length 2	22.29	47.51	57.47	93.48	67.88	46.83	57.71	49.81	88.48	62.86
	Length 3	19.50	47.04	68.65	92.93	67.39	43.19	57.98	50.43	91.23	**62.92**
	Length 4	24.14	47.58	65.39	93.05	65.62	43.84	55.80	49.07	91.67	57.37
	Length 5	20.32	44.00	68.87	92.46	61.94	37.43	**64.79**	49.10	**92.41**	51.30
	Length 6	21.57	41.43	69.56	91.75	59.07	31.34	44.49	32.52	90.49	51.99
	Length 7	26.08	42.79	67.47	92.54	60.56	35.64	55.37	41.04	90.91	50.38
	Length 10	18.51	39.77	73.69	90.74	56.11	24.55	40.78	28.53	87.24	43.92
FewShot	kelm	21.00	46.67	66.34	**93.57**	71.17	44.46	61.88	54.03	90.43	**62.99**
	wikidata	18.71	42.91	72.89	92.60	65.62	37.29	55.68	44.51	90.73	56.57
	webnlg	**27.05**	**51.16**	**62.09**	93.33	**74.36**	**47.17**	**68.23**	**56.96**	**91.95**	61.60
	Length 1	**40.83**	**57.92**	**51.29**	**94.28**	**82.86**	**52.02**	62.14	**60.99**	88.38	55.35
	Length 2	15.34	44.85	68.56	93.40	71.35	47.71	63.15	56.93	92.39	**75.01**
	Length 3	20.56	48.84	63.53	93.68	72.55	46.22	63.78	55.72	**92.93**	65.14
	Length 4	16.98	42.96	74.06	93.00	68.52	40.26	58.16	49.17	90.74	61.01
	Length 5	21.90	46.68	67.36	93.20	66.74	39.76	**67.56**	49.46	92.08	53.91
	Length 6	17.94	38.58	74.70	91.11	60.08	34.24	48.61	37.69	86.96	55.94
	Length 7	21.38	43.21	72.76	92.67	63.92	35.86	60.41	45.51	89.59	54.97
	Length 10	19.07	41.66	78.81	91.78	64.46	32.48	50.06	37.69	91.25	51.14

Table C.12: Results for Russian subsets, categorized by dataset source and graph length, with all metrics. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, Direct = NMT direct, Pivot = NMT pivot, Cross = NMT cross, Comet = Cometkiwi. All values are multiplied by 100 for better readability.

C.6 Annotator recruitment

We use Prolific together with our own website to find the participants for annotations. All participants are paid in accordance with the hourly rate equal to €11.54. The test task is paid estimating it would take 15 minutes, and the full task (100 instances) is paid estimating it would take 90 minutes. In cases when for some languages the median time spent on the test task was more than 15 minutes, we adjusted the payment accordingly to the hourly rate.

In this section we also present the instruction given to annotators. The following instructions were in a separate document and participants had access to it throughout both test and main tasks.

Summary of what to do

1. Read this document carefully.
2. Return back to the annotation website.
3. Find inconsistencies between data and text in the given set of data/text pairs.
4. Answer questions about the presence of those inconsistencies and fluency of the text.
5. Repeat steps 3 and 4 until you reach the end of a survey.
6. Copy the code from the end of the survey.
7. Paste the code into Prolific to receive compensation.

Test/main task information

If you are taking the test task, you will be presented with 15 data/text pairs. The main task will only be available to you if your annotation of these 15 data/text pairs is sufficient and shows you have read and understood the instructions and understood them.

If you are doing the main task, you will be presented with 100 data/text pairs and you will be able to take a similar task with a different number of data/text pairs later.

Introduction

In this task, you will be shown a number of data/text pairs. We ask you to assess the fluency of the texts, and, separately, the similarities and differences between the data and the text.

The following are a few examples of data/text pairs:

Example #	Text	Data
1	adirondack regional airport serves both lake placid and saranac lake. its 's runway length is 2003.0.	Subject: Adirondack Regional Airport / Predicate: city served / Object: Lake Placid, New York Subject: Adirondack Regional Airport / Predicate: runway length / Object: 2003.0 Subject: Adirondack Regional Airport / Predicate: city served / Object: Saranac Lake, New York
2	adirondack regional airport serves lake placid. its 's runway length is 2003.0.	Subject: Adirondack Regional Airport / Predicate: city served / Object: Lake Placid, New York Subject: Adirondack Regional Airport / Predicate: runway length / Object: 2003.0 Subject: Adirondack Regional Airport / Predicate: city served / Object: Saranac Lake, New York
3	adirondack regional airport, which serves lake placid, serves lake placid. its 's runway length is 2003.0.	Subject: Adirondack Regional Airport / Predicate: city served / Object: Lake Placid, New York

Data Expressions

The column labeled 'data' contains a table where each row expresses a fact, a statement with a single meaning, in Subject Predicate Object form, where subject and object signify entities (things, concepts, etc.) and the predicate signifies the relationship between those entities. Using the above data as an example, the interpretation could be as follows:

Fact: **Subject** Adirondack Regional Airport, **predicate** city served, **object** Lake Placid, New York

Text: Adirondack Regional Airport serves the city of Lake Placid, New York

Fact: **Subject** Adirondack Regional Airport, **predicate** runway length, **object** 2003.0

Text: The runway length of Adirondack Regional Airport is 2003.0 [units]

Fact: **Subject** Adirondack Regional Airport, **predicate** city served, **object** Saranac Lake, New York,

Text: Adirondack Regional Airport serves the city of Saranac Lake, New York

Note that predicate and arguments are not strictly speaking words, but concepts. E.g. "Lake Placid, New York" is a single unique identifier for this particular city, and its content is therefore the city, not the city and the state (New York) both.

Texts

The text should express the content (meaning) of the data, but in standard, fluent language. While texts often use different words than can be seen in elements of the data, such words should nevertheless have the same meaning as some part of the data. The fluency of the texts you will receive varies.

Evaluation Process

Assessment of similarities and differences between data and text

It is necessary to identify omissions, additions, and/or repetitions and correct them. Taking the example above we provide a possible thought process to pinpoint those problems in the text. Note that data or text may occasionally contain factual inaccuracies. In these cases the data should be represented to the best degree possible within the text. We do not consider such inaccuracies (if they are present in the data) as errors. If an element is expressed in the text but is distorted we do not consider it to be the same element. For example, if the data talk about New York, but in the text we see Nyew Yeork, we consider it a separate entity. For languages other than English, verify that the element is expressed in the text in a way it is commonly expressed in your language. I.e. for languages with non-Latin scripts make sure the elements are transliterated understandably and in the way which is common for the language.

Checking for Omissions (in the text)

Looking at ***each element of the Data*** *in turn, does the Text express each of these elements in full (allow synonyms and aggregation)?*

Our task is to look at each row of the table, and each of its three components, i.e. subject, predicate, and object, in turn, considering whether they are fully expressed by the text.

- Example #1. There are no omissions. The answer to the question is Yes. Explanation:

  Subject Adirondack Regional Airport, predicate city served, object Lake Placid, New York: All three components are expressed by "adirondack regional airport serves ... lake placid.." Recall that Lake Placid, New York identifies only the city, so expressing the state is optional.

Subject Adirondack Regional Airport, predicate runway length, object 2003.0: All three components are covered by "its 's runway length is 2003.0." Here, the possessive pronoun "its" is a valid way of expressing Adirondack Regional Airport, because it has already been mentioned.

Subject Adirondack Regional Airport, predicate city served, object Saranac Lake, New York: All three components are expressed by "adirondack regional airport serves ... saranac lake." As explained above, Saranac Lake, New York is a single identifier for the city of Saranac Lake, therefore expressing the state is optional.

Note that the "both ... and" construction is a valid way of avoiding repetition of "adirondack regional airport serves". This is a form of aggregation.

- Example #2. There is an omission. The answer to the question is No. Explanation:

  One can see that the third row of the data table is not expressed via text, as Saranac Lake is not mentioned in any way.

- Example #3. There are no omissions. The answer to the question is Yes. Explanation:

  There is only one row in the Data table and it is fully expressed by the first sentence of the text.

Checking for Additions (in the text)

Looking at the Text, is all of its content expressed in the Data ? (Allow duplication of content.)

Here our task is to look at each part of the text in turn, considering whether each is fully expressed by the data.

- Example #1. There are no additions. The answer to the question is Yes. Explanation:

  adirondack regional airport serves: This part of the text is expressed by two occurrences with a subject Adirondack Regional Airport and a predicate city served. both lake placid and saranac lake: This part of the text is expressed by two objects Lake Placid, New York and Saranac Lake, New York its 's runway length is 2003.0.: This last part of the text is fully expressed by the second row in Data.

- Example #2. There are no additions. The answer to the question is Yes. Explanation:

  The first sentence of the text is expressed in the first row of the Data. The second sentence of the text is expressed in the second row of the Data.

- Example #3. There is an addition. The answer to the question is No. Explanation:

  The second sentence of the text is not expressed by the Data. The runway length is never mentioned in the Data.

Checking for repetitions (in the text)

Is any content in the Text unnecessarily repeated?

- Example #1. No repetitions.
- Example #2. No repetitions.
- Example #3. Yes. The fact the airport serves lake placid is mentioned two times without any need for it. We ask you to modify the text so that it would not.

Assessment of fluency

Please rate the text shown in terms of fluency on a scale of 1 to 5 where 5 is the highest (best) score. Highly fluent text 'flows well' and is well connected and free from disfluencies.

Let's look at a few examples of fluency evaluation.

- adirondack regional airport serves both lake placid and saranac lake. its 's runway length is 2003.0.

  Overall the meaning is clear, there is however one grammar mistake ("its 's" should be just its). We can rate the fluency as 4 out of 5.

- Nord (Year of No Light album) was a 2006 09 06 player, Year of No Light. He was signed to Crucial Blast and was also known as E Vinyl. He has been also been also known as 58.41.

  The text is hard to understand, it's ambiguous what the pronoun "he" refers to. Since the fluency is poor, we can rate it as 1 out of 5.

C.7 Annotation Website

We provide a screenshot of how the annotation website looks in Figure C.2.

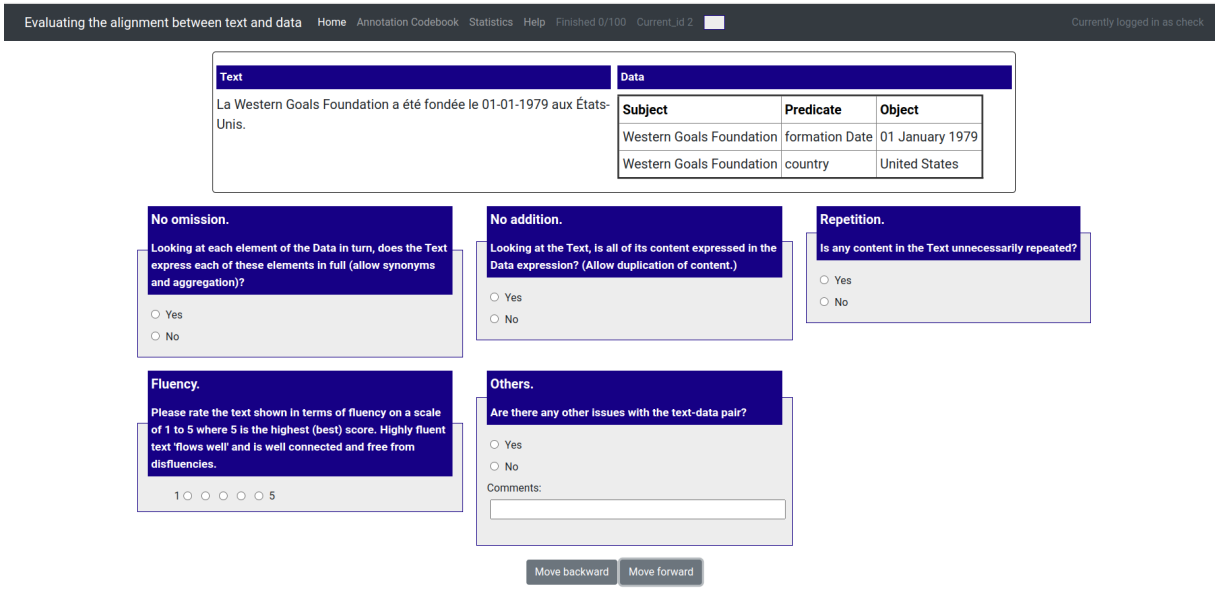


Figure C.2: A screenshot of the website with an annotation example

C.8 Full Human Evaluation results

In this section, we provide human evaluation results for each language on each part of the 100-test set divided by source.

Language	No A ↑	No O ↑	Rep ↓	Fluency ↑
English	0.86	**0.93**	0.17	4.12
French	**0.94**	0.88	0.11	4.19
Spanish	0.92	0.89	0.11	**4.53**
Russian	0.89	0.76	**0.07**	4.30
Chinese	0.91	0.84	0.09	4.05

(a) Scores for human evaluation for each language on the full test set. 'No A' stands for 'No addition', 'No O' stands for 'No omission' and 'Rep' stands for 'repetition'

Language	No A	No O	Rep	Fluency
English	0.34	0.33	0.30	0.07
French	0.61	0.81	0.30	0.05
Spanish	0.43	0.32	0.24	0.24
Russian	0.22	0.68	0.20	0.12
Chinese	0.47	0.78	0.59	-0.05

(b) Fleiss' kappa for inter-annotator agreement for ealanguage and each annotation question for the full test set.

Table C.13: Full test set.

Language	No A ↑	No O ↑	Rep ↓	Fluency ↑
English	0.88	0.95	0.09	4.15
French	0.96	0.93	0.15	4.24
Spanish	0.92	0.93	0.13	4.57
Russian	0.93	0.91	0.07	4.24
Chinese	0.93	0.87	0.04	4.19

(a) Scores for human evaluation for each language on WebNLG part of the test set. 'No A' stands for 'No addition', 'No O' stands for 'No omission' and 'Rep' stands for 'repetition'.

Language	No A	No O	Rep	Fluency
English	0.24	0.21	0.52	0.20
French	0.31	0.57	0.15	0.10
Spanish	0.46	0.14	0.19	0.21
Russian	-0.07	0.37	0.14	0.33
Chinese	-0.07	0.65	0.31	0.00

(b) Fleiss' kappa for inter-annotator agreement for each language and each annotation question for WebNLG part of the data.

Table C.14: WebNLG

Language	No A ↑	No O ↑	Rep ↓	Fluency ↑
English	0.93	0.97	0.27	4.05
French	0.93	0.92	0.21	4.09
Spanish	0.95	0.95	0.15	4.40
Russian	0.91	0.73	0.13	4.16
Chinese	0.96	0.97	0.19	4.01

(a) Scores for human evaluation for each language on KELM part of the test set. 'No A' stands for 'No addition', 'No O' stands for 'No omission' and 'Rep' stands for 'repetition'.

Language	No A	No O	Rep	Fluency
English	-0.07	-0.03	0.32	0.11
French	0.57	0.28	0.36	0.17
Spanish	0.21	0.21	0.36	0.11
Russian	0.21	0.59	0.19	0.15
Chinese	0.31	-0.02	0.74	0.01

(b) Fleiss' kappa for inter-annotator agreement for each language and each annotation question for KELM part of the data.

Table C.15: Kelm

Language	No A ↑	No O ↑	Rep ↓	Fluency ↑
English	0.81	0.90	0.15	4.13
French	0.93	0.84	0.05	4.21
Spanish	0.91	0.84	0.09	4.57
Russian	0.86	0.71	0.03	4.41
Chinese	0.87	0.75	0.07	3.99

(a) Scores for human evaluation for each language on Wikikdata part of the test set. 'No A' stands for 'No addition', 'No O' stands for 'No omission' and 'Rep' stands for 'repetition'.

Language	No A	No O	Rep	Fluency
English	0.43	0.41	0.17	-0.01
French	0.71	1.00	0.25	-0.05
Spanish	0.49	0.36	0.16	0.32
Russian	0.28	0.77	0.17	-0.04
Chinese	0.64	0.86	0.46	-0.11

(b) Fleiss' kappa for inter-annotator agreement for each language and each annotation question for Wikidata part of the data.

Table C.16: Wikidata

C.9 Prompting Options

This section illustrates the various components of the prompting strategies we tried. The three ways of selecting few shots (random, size based and overlap) are shown in Listing C.5, Listing C.6 and Listing C.7 using the Direct Generation approach. The Chain-of-Thought prompting is illustrated in Listing C.8, and Cumulative Generation in Listing C.9, Listing C.10 and Listing C.11 all with Overlap few-shots. Enriching the input with labels was shown in the main body of the thesis (Table 4.4).

```
<|begin_of_text|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "Mermaid_(Train_song)",
            "property": "producer",
            "object": "Espionage_(production_team)"
        },
        {
            "id": 1,
            "subject": "Mermaid_(Train_song)",
            "property": "followedBy",
            "object": "Imagine_(John_Lennon_song)"
        },
        {
            "id": 2,
            "subject": "Imagine_(John_Lennon_song)",
            "property": "musicalArtist",
            "object": "John_Lennon"
        },
        {
            "id": 3,
            "subject": "Imagine_(John_Lennon_song)",
            "property": "certification",
            "object": "FIMI"
        }
    ]
}

Generate a lexicalisation of all the 4 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{
    "full-text": "The Train song Mermaid, produced by the Espionage production team,
    was followed by John Lennon's song Imagine. Musician John Lennon wrote the song
    Imagine which is certified by FIMI."
}
<|eot_id|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "Western Goals Foundation",
            "property": "formationDate",
            "object": "01 January 1979"
        },
        {
            "id": 1,
            "subject": "Western Goals Foundation",
            "property": "country",
            "object": "United States"
```

```
        }
    ]
}

Generate a lexicalisation of all the 2 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Listing C.5: Example of Direct generation using Random One-Shot in English

```
<|begin_of_text|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "Buzz_Aldrin",
            "property": "birthDate",
            "object": "1930-01-20"
        },
        {
            "id": 1,
            "subject": "Buzz_Aldrin",
            "property": "mission",
            "object": "Apollo_11"
        }
    ]
}

Generate a lexicalisation of all the 2 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{
    "full-text": "Buzz Aldrin (born on 1930-01-20) was in the Apollo 11 mission."
}
<|eot_id|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "Western Goals Foundation",
            "property": "formationDate",
            "object": "01 January 1979"
        },
        {
            "id": 1,
            "subject": "Western Goals Foundation",
            "property": "country",
            "object": "United States"
        }
    ]
}

Generate a lexicalisation of all the 2 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Listing C.6: Example of Direct generation using Size One-Shot in English

```
<|begin_of_text|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "AmeriGas",
            "property": "country",
            "object": "United States"
        },
        {
            "id": 1,
            "subject": "AmeriGas",
            "property": "foundingDate",
            "object": "1959-01-01"
        }
    ]
}

Generate a lexicalisation of all the 2 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{
    "full-text": "AmeriGas was founded in the United States on 1959-01-01."
}
<|eot_id|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "Western Goals Foundation",
            "property": "formationDate",
            "object": "01 January 1979"
        },
        {
            "id": 1,
            "subject": "Western Goals Foundation",
            "property": "country",
            "object": "United States"
        }
    ]
}

Generate a lexicalisation of all the 2 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Listing C.7: Example of Direct generation using Overlap One-Shot English

```
<|begin_of_text|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "AmeriGas",
            "property": "country",
            "object": "United States"
        },
        {
            "id": 1,
            "subject": "AmeriGas",
            "property": "foundingDate",
            "object": "1959-01-01"
        }
    ]
}

Generate a lexicalisation of all the 2 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{
    "individual-lexicalisations": [
        {
            "id": 0,
            "lex": "AmeriGas is in the United States."
        },
        {
            "id": 1,
            "lex": "AmeriGas was founded on 1959-01-01."
        },
    ]
    "full-text": "AmeriGas was founded in the United States on 1959-01-01."
}
<|eot_id|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "Western Goals Foundation",
            "property": "formationDate",
            "object": "01 January 1979"
        },
        {
            "id": 1,
            "subject": "Western Goals Foundation",
            "property": "country",
            "object": "United States"
        }
    ]
}

Generate a lexicalisation of all the 2 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Listing C.8: Example of CoT generation using Overlap One-Shot English

```
<|begin_of_text|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "AmeriGas",
            "property": "foundingDate",
            "object": "1959-01-01"
        }
    ]
}

Generate a lexicalisation of all the 1 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{
    "full-text": "AmeriGas was founded on 1959-01-01."
}
<|eot_id|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "Western Goals Foundation",
            "property": "formationDate",
            "object": "01 January 1979"
        }
    ]
}

Generate a lexicalisation of all the 1 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Listing C.9: Example of Cumulative generation (Step 1a) using Overlap One-Shot English

```
<|begin_of_text|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
            "id": 0,
            "subject": "AmeriGas",
            "property": "country",
            "object": "United States"
        }
    ]
}

Generate a lexicalisation of all the 1 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{
    "full-text": "AmeriGas is in the United States."
}
<|eot_id|><|start_header_id|>user<|end_header_id|>

{
    "data": [
        {
```

```
            "id": 0,
            "subject": "Western Goals Foundation",
            "property": "country",
            "object": "United States"
        }
    ]
}

Generate a lexicalisation of all the 1 triples in English.
When necessary, generate text from other languages in their original script.
Include all the information from the triples regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Listing C.10: Example of Cumulative generation (Step 1b) using Overlap One-Shot English

```
<|begin_of_text|><|start_header_id|>user<|end_header_id|>

{
    "sentences": [
        "AmeriGas was founded on 1959-01-01.",
        "AmeriGas is in the United States."
    ]
}

Generate a combined paragraph of all the 2 sentences in English.
Include all the information from the sentences regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{
    "full-text": "AmeriGas was founded in the United States on 1959-01-01."
}
<|eot_id|><|start_header_id|>user<|end_header_id|>

{
    "sentences": [
        "Western Goals Foundation was formed on 01 January 1979.",
        "Western Goals Foundation is in the United States."
    ]
}

Generate a combined paragraph of all the 2 sentences in English.
Include all the information from the sentences regardless of their type or relevance.
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Listing C.11: Example of Cumulative generation (Step 2) using Overlap One-Shot English

C.10 Extended Evaluation Results

C.10.1 English Results

Model	LLM	Dataset	BLEU	ChrF	BS	BLRT	NMT	SBERT
FTMT	–	WebNLG	46.07	67.61	95.48	76.82	53.33	95.16
	–	5L-GOLD-OOD	24.40	53.99	93.20	66.84	39.31	88.93
NLG+MT	–	WebNLG	53.80	71.13	96.14	80.06	59.23	96.86
	–	5L-GOLD-OOD	34.97	62.34	95.10	74.95	50.90	94.10
FewShot	LLaMA3-8B	WebNLG	50.06	70.18	95.99	80.01	57.93	96.92
	LLaMA3-8B	5L-GOLD-OOD	50.34	72.68	96.70	81.67	63.93	97.11
	LLaMA3-70B	WebNLG	50.21	70.86	96.07	80.72	59.23	97.16
	LLaMA3-70B	5L-GOLD-OOD	59.45	79.18	97.27	84.56	70.44	97.58
	Qwen2.5-7B	WebNLG	49.27	70.86	95.90	79.34	56.52	96.49
	Qwen2.5-7B	5L-GOLD-OOD	51.37	73.68	96.58	80.78	62.16	95.95
	Qwen2.5-72B	WebNLG	49.09	71.59	96.01	80.24	58.53	96.78
	Qwen2.5-72B	5L-GOLD-OOD	52.46	75.39	96.80	82.24	64.75	96.62

Table C.17: Few-shot and baseline results on English WebNLG and 5L-GOLD-OOD. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, NMT = NMTScore-Direct. All values are multiplied by 100 for better readability, and all LLMs are instruction-tuned versions.

C.10.2 Chinese Results

Model	LLM	Dataset	BLEU	ChrF	BS	BLRT	NMT	SBERT
FTMT	–	5L-GOLD-OOD	17.61	21.25	90.20	48.70	33.17	78.09
NLG+MT	–	5L-GOLD-OOD	29.03	29.41	92.20	60.30	44.28	86.97
FewShot	LLaMA3-8B	5L-GOLD-OOD	7.86	7.24	30.32	18.86	14.55	26.96
	LLaMA3-70B	5L-GOLD-OOD	22.49	23.33	76.64	49.73	38.40	72.12
	Qwen2.5-7B	5L-GOLD-OOD	38.12	34.02	93.53	69.30	51.88	89.77
	Qwen2.5-72B	5L-GOLD-OOD	48.07	41.32	95.07	74.25	61.65	93.48

Table C.18: Few-shot and baseline results for Chinese on 5L-GOLD-OOD. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, NMT = NMTScore-Direct. All values are multiplied by 100 for better readability, and all LLMs are instruction-tuned versions.

C.10.3 Spanish Subsets Results

Model	LLM	Dataset	BLEU	ChrF	BS	BLRT	NMT	SBERT
FTMT	–	WebNLG	34.42	57.03	93.80	66.55	45.01	89.32
	–	5L-GOLD-OOD	13.42	37.48	90.16	45.85	28.71	75.76
NLG+MT	–	WebNLG	43.52	63.70	95.04	75.71	56.97	95.58
	–	5L-GOLD-OOD	32.83	58.65	94.42	69.56	49.36	92.86
FewShot	LLaMA3-8B	WebNLG	42.20	63.97	95.01	76.31	55.47	95.10
	LLaMA3-8B	5L-GOLD-OOD	33.74	58.89	94.79	72.41	52.91	93.96
	LLaMA3-70B	WebNLG	44.01	65.91	95.22	77.59	57.21	95.81
	LLaMA3-70B	5L-GOLD-OOD	33.85	61.26	94.98	73.60	55.23	94.85
	Qwen2.5-7B	WebNLG	40.40	63.44	94.92	76.47	55.00	95.15
	Qwen2.5-7B	5L-GOLD-OOD	31.74	57.16	94.59	71.70	53.37	94.17
	Qwen2.5-72B	WebNLG	40.88	65.14	95.03	77.88	56.71	95.69
	Qwen2.5-72B	5L-GOLD-OOD	33.07	60.43	94.78	73.94	54.34	94.72

Table C.19: Few-shot and baseline results on Spanish WebNLG and 5L-GOLD-OOD. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, NMT = NMTScore-Direct. All values are multiplied by 100 for better readability, and all LLMs are instruction-tuned versions.

C.10.4 French Results

Model	LLM	Dataset	BLEU	ChrF	BS	BLRT	NMT	SBERT
FTMT	–	5L-GOLD-OOD	14.26	40.71	90.47	37.07	35.16	79.06
NLG+MT	–	5L-GOLD-OOD	30.33	57.75	93.86	61.18	54.42	92.47
FewShot	LLaMA3-8B	5L-GOLD-OOD	31.69	57.63	94.39	67.62	58.05	93.49
	LLaMA3-70B	5L-GOLD-OOD	32.91	60.09	94.61	69.47	60.54	94.42
	Qwen2.5-7B	5L-GOLD-OOD	29.89	56.00	94.21	66.36	58.09	93.62
	Qwen2.5-72B	5L-GOLD-OOD	33.19	60.81	94.66	70.83	61.22	94.48

Table C.20: Few-shot and baseline results for French on 5L-GOLD-OOD. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, NMT = NMTScore-Direct. All values are multiplied by 100 for better readability, and all LLMs are instruction-tuned versions.

C.10.5 Russian Results

Model	LLM	Dataset	BLEU	ChrF	BS	BLRT	NMT	SBERT
FTMT	–	WebNLG	25.43	48.89	93.64	69.38	48.76	89.86
	–	5L-GOLD-OOD	11.18	31.61	90.63	51.46	29.31	81.05
NLG+MT	–	WebNLG	23.45	46.74	93.17	67.25	45.94	89.21
	–	5L-GOLD-OOD	19.51	41.83	92.61	63.55	39.97	90.30
FewShot	LLaMA3-8B	WebNLG	25.64	51.03	93.82	72.41	48.62	91.34
	LLaMA3-8B	5L-GOLD-OOD	23.11	45.07	93.55	71.87	46.88	92.06
	LLaMA3-70B	WebNLG	27.30	52.88	94.00	73.76	51.14	92.05
	LLaMA3-70B	5L-GOLD-OOD	26.64	50.74	94.18	74.77	51.30	93.36
	Qwen2.5-7B	WebNLG	24.52	50.68	93.54	70.63	47.54	90.88
	Qwen2.5-7B	5L-GOLD-OOD	24.37	47.53	93.51	70.38	47.74	92.07
	Qwen2.5-72B	WebNLG	24.88	52.69	93.68	91.29	49.21	91.29
	Qwen2.5-72B	5L-GOLD-OOD	27.84	52.34	94.21	75.05	51.99	93.61

Table C.21: Few-shot and baseline results on Russian WebNLG and 5L-GOLD-OOD. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, NMT = NMTScore-Direct. All values are multiplied by 100 for better readability, and all LLMs are instruction-tuned versions.

C.10.6 Breton Results (Br)

Model	LLM	Dataset	BLEU	ChrF	BS	BLRT	NMT	SBERT
FTMT	–	WebNLG	14.59	41.53	88.50	45.44	28.00	79.39
NLG+MT	–	WebNLG	9.12	31.47	84.66	17.30	25.32	77.25
FewShot	LLaMA3-8B	WebNLG	16.62	45.09	89.05	47.44	29.96	83.53
	LLaMA3-70B	WebNLG	17.68	46.88	89.27	48.67	30.41	83.95
	Qwen2.5-7B	WebNLG	12.62	40.87	87.64	44.77	26.19	81.45
	Qwen2.5-72B	WebNLG	14.18	43.87	88.50	47.30	28.58	83.10

Table C.22: Few-shot and baseline results on Breton WebNLG. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, NMT = NMTScore-Direct. All values are multiplied by 100 for better readability, and all LLMs are instruction-tuned versions..

C.10.7 Irish Results (Ga)

Model	LLM	Dataset	BLEU	ChrF	BS	BLRT	NMT	SBERT
FTMT	–	WebNLG	14.70	42.09	89.32	52.85	21.38	75.84
NLG+MT	–	WebNLG	15.78	43.09	89.79	54.65	21.75	78.45
FewShot	LLaMA3-8B	WebNLG	16.45	46.20	89.80	55.73	20.44	80.32
	LLaMA3-70B	WebNLG	17.27	48.46	90.23	59.59	21.28	81.27
	Qwen2.5-7B	WebNLG	13.66	43.38	89.07	49.47	17.46	78.76
	Qwen2.5-72B	WebNLG	16.14	47.93	89.90	57.09	20.01	81.27

Table C.23: Few-shot and baseline results on Irish WebNLG. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, NMT = NMTScore-Direct. All values are multiplied by 100 for better readability, and all LLMs are instruction-tuned versions..

C.10.8 Maltese Results (Mt)

Model	LLM	Dataset	BLEU	ChrF	BS	BLRT	NMT	SBERT
FTMT	–	WebNLG	10.93	39.42	87.97	21.71	-	76.50
NLG+MT	–	WebNLG	14.03	46.70	89.26	33.39	-	83.23
FewShot	LLaMA3-8B	WebNLG	14.36	46.31	89.03	29.25	-	84.93
	LLaMA3-70B	WebNLG	15.92	50.07	89.72	34.41	-	86.99
	Qwen2.5-7B	WebNLG	12.42	43.35	88.20	24.42	-	83.93
	Qwen2.5-72B	WebNLG	13.49	47.22	89.02	23.98	-	85.69

Table C.24: Few-shot and baseline results on Maltese WebNLG. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, NMT = NMTScore-Direct. All values are multiplied by 100 for better readability, and all LLMs are instruction-tuned versions..

C.10.9 Welsh Results (Cy)

Model	LLM	Dataset	BLEU	ChrF	BS	BLRT	NMT	SBERT
FTMT	–	WebNLG	12.94	38.50	88.78	27.40	19.22	76.04
NLG+MT	–	WebNLG	16.35	43.53	89.93	32.29	20.79	81.98
FewShot	LLaMA3-8B	WebNLG	17.08	47.34	90.43	38.06	20.59	83.45
	LLaMA3-70B	WebNLG	17.20	49.25	90.80	41.26	21.73	84.20
	Qwen2.5-7B	WebNLG	12.13	42.43	88.97	25.39	16.81	80.36
	Qwen2.5-72B	WebNLG	13.73	46.24	89.89	32.68	18.76	83.12

Table C.25: Few-shot and baseline results on Welsh WebNLG. Abbreviations: CP = Control Prefix (refers to NLG-MT), BS = BERTScore, BLRT = Bleurt, NMT = NMTScore-Direct. All values are multiplied by 100 for better readability, and all LLMs are instruction-tuned versions..

C.11 Hyper Parameters

Table C.26 shows the hyperparameters of the different generation approaches.

HyperParameter	FTMT(Helsinki)	FTMT (M2M)	NLG+MT	Prompting
General				
Hardware	1 32GB V100	1 32GB V100	1 32GB V100	1 32GB V100
Parameters	75M	1.2B	770M + (75M or 1.2B)	8 B
Training				
Instances	∼ 30K	∼ 30K	∼ 30k	N/A
Epochs	4	4	40	N/A
Time	∼ 1h	∼ 5h	∼ 40h	N/A
Optimizer	Adafactor	Adafactor	Adafactor	N/A
Learning Rate	2e-5	2e-5	7e-5	N/A
Inference				
Num. Beam	4	5	5	N/A
Temperature	N/A	N/A	N/A	0.7
Top P	N/A	N/A	N/A	0.7

Table C.26: Hyper parameters of the different models.

Bibliography

Ankush Agarwal, Raj Gite, Shreya Laddha, Pushpak Bhattacharyya, Satyanarayan Kar, Asif Ekbal, Prabhjit Thind, Rajesh Zele, and Ravi Shankar. 2022. Knowledge graph - deep learning: A case study in question answering in aviation safety domain. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6260–6270, Marseille, France. European Language Resources Association.

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.

Oshin Agarwal, Mihir Kale, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2020. Machine translation aided bilingual data-to-text generation and semantic parsing. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 125–130, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Hiba Ahsan, Daivat Bhatt, Kaivan Shah, and Nikita Bhalla. 2021. Multi-modal image captioning for the visually impaired. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 53–60, Online. Association for Computational Linguistics.

Ojas Ahuja, Jiacheng Xu, Akshay Gupta, Kevin Horecka, and Greg Durrett. 2022. ASPECTNEWS: Aspect-oriented summarization of news documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6494–6506, Dublin, Ireland. Association for Computational Linguistics.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, and 17 others. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Suha S Al-Thanyyan and Aqil M Azmi. 2021. Automated text simplification: a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

Mariana Arroyo Chavez, Molly Feanny, Matthew Seita, Bernard Thompson, Keith Delk, Skyler Officer, Abraham Glasser, Raja Kushalnagar, and Christian Vogler. 2024. How users experience closed captions on live television: quality metrics remain a challenge. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–16.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg. Springer-Verlag.

Agnes Axelsson and Gabriel Skantze. 2023. Using large language models for zero-shot natural language generation from knowledge graphs. In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 39–54, Prague, Czech Republic. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, 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.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, and 2 others. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara

Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2025. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255, Vienna, Austria. Association for Computational Linguistics.

Nadine Beks van Raaij, Daan Kolkman, and Ksenia Podoynitsyna. 2024. Clearer governmental communication: Text simplification with ChatGPT evaluated by quantitative and qualitative research. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024*, pages 152–178, Torino, Italia. ELRA and ICCL.

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455.

Anya Belz, Craig Thomson, Ehud Reiter, Gavin Abercrombie, Jose M. Alonso-Moral, Mohammad Arvan, Anouck Braggaar, Mark Cieliebak, Elizabeth Clark, Kees van Deemter, Tanvi Dinkar, Ondřej Dušek, Steffen Eger, Qixiang Fang, Mingqi Gao, Albert Gatt, Dimitra Gkatzia, Javier González-Corbelle, Dirk Hovy, and 23 others. 2023. Missing information, unresponsive authors, experimental flaws: The impossibility of assessing the reproducibility of previous human evaluations in NLP. In *Proceedings of the Fourth Workshop on Insights from Negative Results in NLP*, pages 1–10, Dubrovnik, Croatia. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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, and 12 others. 2020b. Language models are few-shot learners. *CoRR*, abs/2005.14165.

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, and 12 others. 2020c. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Sheila Castilho, Stephen Doherty, Federico Gaspari, and Joss Moorkens. 2018. Approaches to human and machine translation quality assessment. In *Translation quality assessment: From principles to practice*, pages 9–38. Springer.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020a. 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, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina, editors. 2020b. *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. Association for Computational Linguistics, Dublin, Ireland (Virtual).

Arun Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Melbourne, Australia. Association for Computational Linguistics.

Harrison Chase. 2022. LangChain.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

David L Chen and Raymond J Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th international conference on Machine learning*, pages 128–135.

Meiqi Chen, Fandong Meng, Yingxue Zhang, Yan Zhang, and Jie Zhou. 2024. Crat: A multi-agent framework for causality-enhanced reflective and retrieval-augmented translation with large language models. *arXiv preprint arXiv:2410.21067*.

Mingda Chen, Sam Wiseman, and Kevin Gimpel. 2021. WikiTableT: A large-scale data-to-text dataset for generating Wikipedia article sections. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 193–209, Online. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Aleksandr Chuklin, Justin Zhao, and Mihir Kale. 2022. CLSE: Corpus of linguistically significant entities. In *Proceedings of the Second Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 78–96, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Jordan Clive, Kris Cao, and Marek Rei. 2022. Control prefixes for parameter-efficient text generation. In *Proceedings of the Second Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 363–382, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Arman Cohan and Nazli Goharian. 2015. Scientific article summarization using citation-context and article's discourse structure. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 390–400, Lisbon, Portugal. Association for Computational Linguistics.

Emilie Colin, Claire Gardent, Yassine M'rabet, Shashi Narayan, and Laura Perez-Beltrachini. 2016. The WebNLG challenge: Generating text from DBPedia data. In *Proceedings of the 9th International Natural Language Generation conference*, pages 163–167, Edinburgh, UK. Association for Computational Linguistics.

Simone Conia, Daniel Lee, Min Li, Umar Farooq Minhas, and Yunyao Li. 2024. Enhancing machine translation experiences with multilingual knowledge graphs. In *AAAI*, pages 23781–23783.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Liam Cripwell, Anya Belz, Claire Gardent, Albert Gatt, Claudia Borg, Marthese Borg, John Judge, Michela Lorandi, Anna Nikiforovskaya, William Soto-Martinez, and Craig Thomson. 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.

Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, and Serge Belongie. 2018. Learning to evaluate image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5804–5812.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Nina Dethlefs and Heriberto Cuayáhuitl. 2011. Hierarchical reinforcement learning and hidden Markov models for task-oriented natural language generation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 654–659, Portland, Oregon, 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.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Daniel Duma and Ewan Klein. 2013. Generating natural language from linked data: Unsupervised template extraction. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 83–94, Potsdam, Germany. Association for Computational Linguistics.

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 421–426, Tokyo, Japan. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Björn Engelmann, Fabian Haak, Christin Katharina Kreutz, Narjes Nikzad-Khasmakhi, and Philipp Schaer. 2023. Text simplification of scientific texts for non-expert readers. *CoRR*, abs/2307.03569.

Shai Erera, Michal Shmueli-Scheuer, Guy Feigenblat, Ora Peled Nakash, Odellia Boni, Haggai Roitman, Doron Cohen, Bar Weiner, Yosi Mass, Or Rivlin, Guy Lev, Achiya Jerbi, Jonathan Herzig, Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, Francesca Bonin, and David Konopnicki. 2019. A summarization system for scientific documents. 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 211–216, Hong Kong, China. Association for Computational Linguistics.

Liana Ermakova, Patrice Bellot, Jaap Kamps, Diana Nurbakova, Irina Ovchinnikova, Eric SanJuan, Elise Mathurin, Sílvia Araújo, Radia Hannachi, Stéphane Huet, and 1 others. 2022. Automatic simplification of scientific texts: Simpletext lab at clef-2022. In *44th European Conference on Information Retrieval (ECIR)*, volume 13186, pages 364–373. Springer International Publishing.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *Preprint*, arXiv:2010.11125.

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, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. 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 4186–4196, Hong Kong, China. Association for Computational Linguistics.

Artem Fedorchenko and Tanel Alumäe. 2025. Optimizing Estonian TV subtitles with semi-supervised learning and LLMs. In *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pages 136–141, Tallinn, Estonia. University of Tartu Library.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

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, Vancouver, Canada. 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.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Asma Ghandeharioun, Judy Hanwen Shen, Natasha Jaques, Craig Ferguson, Noah Jones, Agata Lapedriza, and Rosalind Picard. 2019. Approximating interactive human evaluation with self-play for open-domain dialog systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mozhdeh Gheini, Xiang Ren, and Jonathan May. 2021. Cross-attention is all you need: Adapting pretrained Transformers for machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1765, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dimitra Gkatzia. 2016. Content selection in data-to-text systems: A survey. *arXiv preprint arXiv:1610.08375*.

Eli Goldberg, Norbert Driedger, and Richard I Kittredge. 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019. Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). 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 3143–3152, Hong Kong, China. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ananya Gupta, Jae Takeuchi, and Bart Knijnenburg. 2023. On the real-world performance of machine translation: Exploring social media post-authors' perspectives. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 302–310, Toronto, Canada. Association for Computational Linguistics.

Kelvin Han and Claire Gardent. 2023. 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 (Volume 1: Long Papers)*, pages 285–304, Nusa Dua, Bali. Association for Computational Linguistics.

Jie He, Yijun Yang, Wanqiu Long, Deyi Xiong, Victor Gutierrez Basulto, and Jeff Z. Pan. 2025. Evaluating and improving graph to text generation with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10219–10244, Albuquerque, New Mexico. Association for Computational Linguistics.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Cuong Hoang, Devendra Sachan, Prashant Mathur, Brian Thompson, and Marcello Federico. 2023. Improving retrieval augmented neural machine translation by controlling source and fuzzy-match interactions. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 289–295, Dubrovnik, Croatia. Association for Computational Linguistics.

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

MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2019. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36.

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.

F Jelinek. 1976. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64(4):532–556.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Qiao Jin, Won Kim, Qingyu Chen, Donald C. Comeau, Lana Yeganova, W. John Wilbur, and Zhiyong Lu. 2023. Biocpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval. *CoRR*, abs/2307.00589.

Zhijing Jin, Qipeng Guo, Xipeng Qiu, and Zheng Zhang. 2020. GenWiki: A dataset of 1.3 million content-sharing text and graphs for unsupervised graph-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2398–2409, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mayank Jobanputra and Vera Demberg. 2024. TeamSaarLST at the GEM'24 data-to-text task: Revisiting symbolic retrieval in the LLM-age. In *Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges*, pages 92–99, Tokyo, Japan. Association for Computational Linguistics.

Anirudh Joshi, Namit Katariya, Xavier Amatriain, and Anitha Kannan. 2020. Dr. summarize: Global summarization of medical dialogue by exploiting local structures. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3755–3763, Online. Association for Computational Linguistics.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume*

1: Long Papers), pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using Mechanical Turk to evaluate open-ended text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1265–1285, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ivan Kartac, Mateusz Lango, and Ondrej Dusek. 2025. OpeNLGauge: An explainable metric for NLG evaluation with open-weights LLMs. In *Proceedings of the 18th International Natural Language Generation Conference*, pages 292–337, Hanoi, Vietnam. Association for Computational Linguistics.

Zdeněk Kasner, Vilém Zouhar, Patrícia Schmidtová, Ivan Kartáč, Kristýna Onderková, Ondřej Plátek, Dimitra Gkatzia, Saad Mahamood, Ondřej Dušek, and Simone Balloccu. 2025. Llms as span annotators: A comparative study of llms and humans. *Preprint*, arXiv:2504.08697.

Maxim Kazakov, Julia Preobrazhenskaya, Ivan Bulychev, and Aleksandr Shain. 2023. WebNLG-interno: Utilizing FRED-t5 to address the RDF-to-text problem (WebNLG 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 67–72, Prague, Czech Republic. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. 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 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada. Association for Computational Linguistics.

Aomi Koyama, Tomoshige Kiyuna, Kenji Kobayashi, Mio Arai, and Mamoru Komachi. 2020. Construction of an evaluation corpus for grammatical error correction for learners of Japanese as a second language. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 204–211, Marseille, France. European Language Resources Association.

Nalin Kumar, Saad Obaid Ul Islam, and Ondrej Dusek. 2023. Better translation + split and generate for multilingual RDF-to-text (WebNLG 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 73–79, Prague, Czech Republic. Association for Computational Linguistics.

Pranjal Kumar. 2024. Large language models (llms): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10):260.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Ora Lassila and Ralph R. Swick. 1999. Resource description framework (rdf) model and syntax.

Teven Le Scao and Claire Gardent. 2023. Joint representations of text and knowledge graphs for retrieval and evaluation. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 110–122, Nusa Dua, Bali. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Yebin Lee, Imseong Park, and Myungjoo Kang. 2024. FLEUR: An explainable reference-free evaluation metric for image captioning using a large multimodal model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3732–3746, Bangkok, Thailand. Association for Computational Linguistics.

Leo Leppänen, Myriam Munezero, Stefanie Sirén-Heikel, Mark Granroth-Wilding, and Hannu Toivonen. 2017. Finding and expressing news from structured data. In *Proceedings of the 21st International Academic Mindtrek Conference*, pages 174–183.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Junyi Li, 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.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2025. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In *Forty-second International Conference on Machine Learning*.

Percy Liang, Michael Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99, Suntec, Singapore. 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.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2024. RA-DIT: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Tianrui Liu, Qi Cai, Changxin Xu, Bo Hong, Jize Xiong, Yuxin Qiao, and Tsungwei Yang. 2024b. Image captioning in news report scenario. *arXiv preprint arXiv:2403.16209*.

Tianyu Liu, Fuli Luo, Qiaolin Xia, Shuming Ma, Baobao Chang, and Zhifang Sui. 2019. Hierarchical encoder with auxiliary supervision for neural table-to-text generation: Learning better representation for tables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6786–6793.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Tradumàtica*, (12):0455–463.

Do Xuan Long, Ngoc-Hai Nguyen, Tiviatis Sim, Hieu Dao, Shafiq Joty, Kenji Kawaguchi, Nancy F. Chen, and Min-Yen Kan. 2025. LLMs are biased towards output formats! systematically evaluating and mitigating output format bias of LLMs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 299–330, Albuquerque, New Mexico. Association for Computational Linguistics.

Zola Mahlaza, Tadiwa Magwenzi, C. Maria Keet, and Langa Khumalo. 2024. Automatically generating IsiZulu words from Indo-Arabic numerals. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 254–271, Tokyo, Japan. Association for Computational Linguistics.

Lloyd May, Keita Ohshiro, Khang Dang, Sripathi Sridhar, Jhanvi Pai, Magdalena Fuentes, Sooyeon Lee, and Mark Cartwright. 2024. Unspoken sound: identifying trends in non-speech audio captioning on youtube. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–19.

Susan W McRoy, Songsak Channarukul, and Syed S Ali. 2003. An augmented template-based approach to text realization. *Natural Language Engineering*, 9(4):381–420.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (SR'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12, Melbourne, Australia. Association for Computational Linguistics.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, and Leo Wanner. 2019. The second multilingual surface realisation shared task (SR'19): Overview and evaluation results. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 1–17, Hong Kong, China. Association for Computational Linguistics.

Simon Mille, Kaustubh Dhole, Saad Mahamood, Laura Perez-Beltrachini, Varun Gangal, Mihir Kale, Emiel van Miltenburg, and Sebastian Gehrmann. 2021. Automatic construction of evaluation suites for natural language generation datasets. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Simon Mille, Massimiliano Pronesti, Craig Thomson, Michela Lorandi, Sophie Fitzpatrick, Rudali Huidrom, Mohammed Sabry, Amy O'Riordan, and Anya Belz. 2024a. Filling gaps in Wikipedia: Leveraging data-to-text generation to improve encyclopedic coverage of underrepresented groups. In *Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations*, pages 16–19, Tokyo, Japan. Association for Computational Linguistics.

Simon Mille, Mohammed Sabry, and Anya Belz. 2024b. DCU-NLG-small at the GEM'24 data-to-text task: Rule-based generation and post-processing with t5-base. In *Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges*, pages 84–91, Tokyo, Japan. Association for Computational Linguistics.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Johanna Axelsson, Miruna Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Nyunya Obonyo, and Lining Zhang. 2024c. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges*, pages 17–38, Tokyo, Japan. Association for Computational Linguistics.

Simon Mille, Elaine Uí Dhonnchadha, Lauren Cassidy, Brian Davis, Stamatia Dasiopoulou, and Anya Belz. 2023. Generating Irish text with a flexible plug-and-play architecture. In *Proceedings of the 2nd Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning*, pages 25–42, Singapore. 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, and 1 others. 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.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, and 5 others. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.

Mariana Neves, Cristian Grozea, Philippe Thomas, Roland Roller, Rachel Bawden, Aurélie Névéol, Steffen Castle, Vanessa Bonato, Giorgio Maria Di Nunzio, Federica Vezzani, Maika Vicente Navarro, Lana Yeganova, and Antonio Jimeno Yepes. 2024. Findings of the WMT 2024 biomedical translation shared task: Test sets on abstract level. In *Proceedings of the Ninth Conference on Machine Translation*, pages 124–138, Miami, Florida, USA. Association for Computational Linguistics.

Anna Nikiforovskaya and Claire Gardent. 2024. Evaluating RDF-to-text generation models for English and Russian on out of domain data. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 134–144, Tokyo, Japan. Association for Computational Linguistics.

Anna Nikiforovskaya, William Eduardo Soto Martinez, Evan Parker Kelly Chapple, and Claire Gardent. 2025. Fine-tuning, prompting and RAG for knowledge graph-to-Russian text generation. how do these methods generalise to out-of-distribution

data? In *Proceedings of the 18th International Natural Language Generation Conference*, pages 419–448, Hanoi, Vietnam. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

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, and 20 others. 2022. No language left behind: Scaling human-centered machine translation.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2018. RankME: Reliable human ratings for natural language 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 72–78, New Orleans, Louisiana. Association for Computational Linguistics.

Michael Oliverio, Pier Felice Balestrucci, Alessandro Mazzei, and Valerio Basile. 2025. WebNLG-IT: Construction of an aligned RDF-Italian corpus through machine translation techniques. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12073–12083, Vienna, Austria. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002a. 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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002b. 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.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation

dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. 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 1470–1480, Beijing, China. Association for Computational Linguistics.

Jiaxin Pei, Aparna Ananthasubramaniam, Xingyao Wang, Naitian Zhou, Apostolos Dedeloudis, Jackson Sargent, and David Jurgens. 2022. POTATO: The portable text annotation tool. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 327–337, Abu Dhabi, UAE. Association for Computational Linguistics.

Laura Perez-Beltrachini and Mirella Lapata. 2018. Bootstrapping generators from noisy data. 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 1516–1527, New Orleans, Louisiana. Association for Computational Linguistics.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.

Jeff Pitman. 2021. Google translate: One billion installs, one billion stories. *The Keyword*.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Nima Pourdamghani, Kevin Knight, and Ulf Hermjakob. 2016. Generating English from Abstract Meaning Representations. In *Proceedings of the 9th International Natural Language Generation conference*, pages 21–25, Edinburgh, UK. Association for Computational Linguistics.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on*

Empirical Methods in Natural Language Processing, pages 751–762, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Abdallah Abu Qub'a, Mohammed Nour Abu Guba, and Shehdeh Fareh. 2024. Exploring the use of grammarly in assessing english academic writing. *Heliyon*, 10(15).

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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.

Virginia Ramón-Ferrer, Carlos Badenes-Olmedo, and Oscar Corcho. 2025. Spanish triple-to-text benchmark on low-resource large language models.

Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *Journal of machine learning research*, 11(4).

Amir Hossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. 2025. Benchmarking prompt sensitivity in large language models. *CoRR*, abs/2502.06065.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ehud Reiter. 2018. A structured review of the validity of bleu. *Computational Linguistics*, 44(3):393–401.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural language engineering*, 3(1):57–87.

Ehud Reiter, Chris Mellish, and John Levine. 1995. Automatic generation of technical documentation. *Applied Artificial Intelligence an International Journal*, 9(3):259–287.

Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing AMR-to-text generation with dual graph representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3183–3194, Hong Kong, China. Association for Computational Linguistics.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182.

Geoffrey Sampson and Anna Babarczy. 2008. Definitional and human constraints on structural annotation of english. *Natural Language Engineering*, 14(4):471–494.

Martin Schmitt, Leonardo F. R. Ribeiro, Philipp Dufter, Iryna Gurevych, and Hinrich Schütze. 2021. Modeling graph structure via relative position for text generation from knowledge graphs. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 10–21, Mexico City, Mexico. Association for Computational Linguistics.

Martin Schmitt, Sahand Sharifzadeh, Volker Tresp, and Hinrich Schütze. 2020. An unsupervised joint system for text generation from knowledge graphs and semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7117–7130, Online. Association for Computational Linguistics.

Phillip Schneider, Manuel Klettner, Elena Simperl, and Florian Matthes. 2024. A comparative analysis of conversational large language models in knowledge-based text generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–367, St. Julian's, Malta. 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.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2018. Order-planning neural text generation from structured data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Matthew Shardlow and Raheel Nawaz. 2019. Neural text simplification of clinical letters with a domain specific phrase table. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 380–389, Florence, Italy. Association for Computational Linguistics.

Anastasia Shimorina, Claire Gardent, Shashi Narayan, and Laura Perez-Beltrachini. 2018. WebNLG Challenge: Human Evaluation Results. Technical report, Loria & Inria Grand Est.

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.

Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. 2022. Legal case document summarization: Extractive and abstractive methods and their evaluation. 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 1048–1064, Online only. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Yifei Song and Claire Gardent. 2025. MuCAL: Contrastive alignment for preference-driven KG-to-text generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14238–14281, Suzhou, China. Association for Computational Linguistics.

Yifei Song, William Soto Martinez, Anna Nikiforovskaya, Evan Parker Kelly Chapple, and Claire Gardent. 2025. Multilingual verbalisation of knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1111–1162, Suzhou, China. Association for Computational Linguistics.

William Soto Martinez, Yannick Parmentier, and Claire Gardent. 2024. Generating from AMRs into high and low-resource languages using phylogenetic knowledge and hierarchical QLoRA training (HQL). In *Proceedings of the 17th International Natural Language Generation Conference*, pages 70–81, Tokyo, Japan. Association for Computational Linguistics.

Somayajulu G Sripada, Ehud Reiter, Ian Davy, and Kristian Nilssen. 2004. Lessons from deploying nlg technology for marine weather forecast text generation. *WEATHER*, 5(7).

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *CoRR*, abs/2405.01724.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1017–1024.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Craig Thomson, Ehud Reiter, and Somayajulu Sripada. 2020. SportSett:basketball - a robust and maintainable data-set for natural language generation. In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*, pages 32–40, Santiago de Compostela, Spain. Association for Computational Lingustics.

Yuanhe Tian, Fei Xia, and Yan Song. 2024. Dialogue summarization with mixture of experts based on large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7143–7155, Bangkok, Thailand. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Nasim Tohidi and Chitra Dadkhah. 2022. A short review of abstract meaning representation applications. *Modeling and Simulation in Electrical and Electronics Engineering*, 2(3):1–9.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jannis Vamvas and Rico Sennrich. 2022. NMTScore: A multilingual analysis of translation-based text similarity measures. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 198–213, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Haifeng Wang, Hua Wu, Zhongjun He, Liang Huang, and Kenneth Ward Church. 2022a. Progress in machine translation. *Engineering*, 18:143–153.

Jiaan Wang, Fandong Meng, Yingxue Zhang, and Jie Zhou. 2024a. Retrieval-augmented machine translation with unstructured knowledge. *arXiv preprint arXiv:2412.04342*.

Pancheng Wang, Shasha Li, Kunyuan Pang, Liangliang He, Dong Li, Jintao Tang, and Ting Wang. 2022b. Multi-document scientific summarization from a knowledge graph-centric view. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6222–6233, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024b. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, and Chun Jason Xue. 2024. Retrieval-augmented generation for natural language processing: A survey. *CoRR*, abs/2407.13193.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6233–6251, Bangkok, Thailand. Association for Computational Linguistics.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024. Pride and prejudice: LLM amplifies self-bias in self-refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15474–15492, Bangkok, Thailand. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zixiaofan Yang, Arash Einolghozati, Hakan Inan, Keith Diedrick, Angela Fan, Pinar Donmez, and Sonal Gupta. 2020. Improving text-to-text pre-trained models for the graph-to-text task. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 107–116, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Fan Yin, Jesse Vig, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2023. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3063–3079, Toronto, Canada. Association for Computational Linguistics.

Xunjian Yin and Xiaojun Wan. 2022. How do Seq2Seq models perform on end-to-end data-to-text generation? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7701–7710, Dublin, Ireland. Association for Computational Linguistics.

Han Yu, Jiashuo Liu, Xingxuan Zhang, Jiayun Wu, and Peng Cui. 2024. A survey on evaluation of out-of-distribution generalization. *CoRR*, abs/2403.01874.

Shuzhou Yuan and Michael Faerber. 2023. Evaluating generative models for graph-to-text generation. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 1256–1264, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Haopeng Zhang, Philip S. Yu, and Jiawei Zhang. 2025. A systematic survey of text summarization: From statistical methods to large language models. *ACM Comput. Surv.*, 57(11).

Kun Zhang, Oana Balalau, and Ioana Manolescu. 2023. 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.

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*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Seq2SQL: Generating structured queries from natural language using reinforcement learning.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 4623–4629. AAAI Press.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023a. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023b. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2025. JudgeLM: Fine-tuned large language models are scalable judges. In *The Thirteenth International Conference on Learning Representations*.

Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. ProSA: Assessing and understanding the prompt sensitivity of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1950–1976, Miami, Florida, USA. Association for Computational Linguistics.

Dmitry Zmitrovich, Aleksandr Abramov, Andrey Kalmykov, Vitaly Kadulin, Maria Tikhonova, Ekaterina Taktasheva, Danil Astafurov, Mark Baushenko, Artem Snegirev, Tatiana Shavrina, Sergei S. Markov, Vladislav Mikhailov, and Alena Fenogenova. 2024. A family of pretrained transformer language models for Russian. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 507–524, Torino, Italia. ELRA and ICCL.