Generation of Semantically Consistent Text and Its Evaluation

Rudali Huidrom

BEng, MEng

A dissertation submitted in fulfilment of the requirements for the award of Doctor of Philosophy (PhD)

to the

Dublin City University

School of Computing

Supervisors:

Prof Anya Belz

Dr Alessandra Mileo

August 2025

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of PhD is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Rudali Huidrom

ID No.: **21269841**

Date: Thursday 28th August, 2025

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Anya Belz. Your unwavering guidance, encouragement, and support, both academically and personally, have made this thesis possible. Thank you for being there through the smiles and tears, and for always making time to talk research and life. I'm forever grateful!

I am also sincerely thankful to my second supervisor, Dr. Alessandra Mileo, for your valuable insights, thoughtful discussions, and for helping shape this thesis in the best possible way.

To my parents, Aaum and Mama, thank you for standing by me through every high and low and for always believing in me. To my sister, Tina, your endless love and support have meant the world to me.

To my best friends, Alex, Bidya, Che Nao, Neha, and Oasis, thank you for always being a source of laughter, comfort, and strength. Your late-night conversations, unspeakable memes, and constant belief in me kept me going.

To my dearest friends, Talenza, Victoria, Yuan, and Zihui, thank you for being such an incredible support system. You add so much to my life!

To Dongyoung, thank you for being my constant source of motivation and delulu, and for inspiring me through your creative endeavours to grow into my best self.

To my Dublin family, Alla, Elaine, and Maeve, thank you for making this city feel like home and for the warmth and kindness that made all the difference.

I also extend my thanks to my colleagues and friends at DCU and ADAPT, especially those in the DCU-NLG research group, Craig, Massi, Michela, Mohammed, and Simon, and to Cynthia and Minkun, for their insightful discussions, generosity in sharing knowledge, and their help in making this research journey richer.

I am grateful to the Faculty of Engineering and Computing for awarding me the scholarship that made this PhD possible. To my collaborators, thank you for the many things I learned from working with you.

Finally, to everyone I may have missed by name, thank you. I am truly grateful for all the support, lessons, and encouragement I received throughout this journey.

Contents

List of Figures

List of Tables

Generation of Semantically Consistent Text and Its Evaluation

Rudali Huidrom

Abstract

Generating semantically consistent text remains a crucial challenge in Natural Language Processing (NLP), especially in input-controlled settings such as data-to-text generation. Despite advances in neural language generation, current models often produce outputs that contain semantic inconsistencies with the input, including omissions, hallucinations, and distortions. These issues stem from the black-box, non-deterministic nature of neural models, which limits their interpretability and controllability. This thesis investigates how semantic consistency, the property that all information conveyed by the output is entailed by the input, can be more reliably evaluated and ultimately improved in generated text. Focusing on data-to-text generation, the thesis proposes that advancing semantic controllability requires a combination of principled formalisation of semantic errors and robust evaluation methods. First, it synthesises prior work on error annotation to develop a task-agnostic consensus taxonomy for semantic errors, distinguishing omission, addition, and substitution errors. Second, it applies this taxonomy in human evaluation studies, including span-based semantic error annotation to better understand the nature and reliability of human judgements. Third, it assesses the capacity of automatic evaluation methods, including lexical, embedding-based, and prompt-based approaches, to reflect human ratings of semantic consistency. In this context, the thesis explores LLM-as-judge setups for scoring system outputs and uses LLMs to sanity-check repeated human evaluation experiments, investigating their generalisability and reliability. Together, these contributions provide new tools insights, approaches and results for the fine-grained assessment of semantic consistency and lay the groundwork for more interpretable, reliable, and scalable evaluation frameworks in neural language generation.

Chapter 1

Introduction

1.1 Thesis Topic

Generation of semantically accurate text is a challenging but crucial research problem for many natural language processing (NLP) downstream applications. Some examples where semantic accuracy is crucial are dialogue systems/chatbots, story generation, image captioning, open domain question answering, and data-to-text generation. Previously, predominant techniques for natural language generation (NLG) were based either on templates, rule-based systems, or probabilistic models such as n-gram or log-linear models. These methods lacked flexibility and expressivity. In contrast, recent studies introduce building end-to-end neural models, mitigating the mentioned drawbacks. Neural networks now achieve state-of-the-art performance within several core NLP generation tasks. However, these models perform in a black-box manner, and, due to their non-deterministic nature, are poor in controllability and interpretability, making it hard to generate text that is free of semantic inconsistencies with the input such as hallucinations, omissions, replacements, and factual inaccuracies.

High-quality natural language tends to reflect different linguistic variations across dimensions including semantic, syntactic, and pragmatic aspects of the language. For a neural model to attain human-like quality in the generated text, it should address an effective way of generating text that is semantically consistent with the

input as well as high quality in other respects. Semantic consistency is the property of having some or all elements from the input match the output, in different task-specific ways. In data-to-text scenarios, inputs are structured data which adds to the variedness of consistent output texts. In this thesis, we look at data-to-text generation which is the generation of natural language-like text from meaning representations that are abstract in nature. The scope of this thesis extends to foundations and development of methods for semantically consistent data-to-text generation. The thesis bridges the knowledge gap in semantic controllability for neural NLG in data-to-text (D2T) scenarios.

Although neural models have advanced considerably in producing fluent and coherent language, maintaining semantic consistency remains a persistent challenge for even the most advanced neural generators, whose outputs often contain semantic inconsistencies, such as omitted input content, unsupported additions, or distorted meanings. These issues significantly reduce the reliability of generated content in meaning-critical domains such as healthcare, finance, or document summarisation.

When semantic consistency is lacking, this thesis refers to the resulting issues as *semantic errors*, content-level deviations from the input. These are categorised into three main types: *omission*, where required input content is missing; *addition*, where the output introduces ungrounded content; and *substitution*, where input content is distorted in a meaning-altering way. These distinctions form the foundation for this thesis's approach to annotation and evaluation, enabling targeted and systematic assessments of generation quality. In particular, they are formalised as a consensus taxonomy in Chapter 3, operationalised in span-based human annotation in Chapter 4, and reflected in evaluation setups used in Chapter 5. We outline them in detail in Section 1.4.

The motivation for this aspect of the work arises from the observation that current methods and evaluation approaches fall short in identifying and analysing semantic consistency problems in language generation. Common automatic metrics often rely on surface-level similarity, such as token overlap (for e.g., ROUGE), or

embedding-based similarity (e.g., BERTScore) are not designed to detect specific inconsistencies between input and output. As a result, they do not offer actionable insight into whether generated content preserves the intended meaning. This limitation is especially critical in domains where semantic precision is essential.

Moreover, much of the prior evaluation work has focused on coarse, system-level comparisons or reference-based metrics, with limited attention paid to identifying the types or locations of semantic errors in generated text. This thesis addresses this methodological gap by proposing fine-grained, span-level approaches to semantic error identification and classification, as presented in Chapters 4 and 6.

The thesis's overarching hypothesis is that it is possible to advance our understanding of, and achieve an overall improvement in, the semantic controllability of automatically generated text by developing methods for fine-grained semantic error annotation methods, and utilising results and insights gained to develop improved methods for controlling the semantic correctness of generated texts. This breaks down into investigating the extent to which it is possible to:

- Improve the comparability and robustness of evaluating semantic inconsistencies in generated text by synthesising existing semantic error categorisation schemes and developing a consensus taxonomy for semantic errors, which is validated through its application in input-controlled generation settings (Chapter 3).

- Reliably and informatively detect and analyse semantic consistency problems in existing system outputs through human evaluation, by:

  - Conducting a survey of prior error annotation schemes for both human- and machine-generated text,

  - Applying span-based semantic error annotation using the top-level categories of the consensus taxonomy across outputs from multiple data-to-text systems, and

- Assess the reliability and generalisability of automatic methods for identifying

semantic consistency issues, by:

- Evaluating how well different types of similarity metrics (lexical, embedding-based, and distance-based) correlate with human judgments, and
- Developing and testing an LLM-as-judge setup that uses prompt-based large language models to predict human-like system-level quality scores across semantic consistency and other evaluation criteria, as well as their reliability and generalisability (Chapter 5).

These objectives frame the thesis's contribution to improving semantic controllability in neural text generation, particularly in data-to-text settings. Building on this foundation, the following section outlines the key research questions that guide the thesis and structure the investigations carried out across its core chapters.

1.2 Research Questions

This thesis addresses the underexplored challenge of evaluating and controlling semantic consistency in input-controlled NLG. Despite advances in generating fluent and coherent text, current evaluation methods often struggle to assess whether generated outputs meaningfully align with their inputs. This limitation is especially salient in tasks like data-to-text generation, where fidelity to structured input is crucial.

The core hypothesis of this thesis is that improving semantic controllability in NLG systems requires both principled formalisation of semantic errors and robust methods, both human and automatic, for identifying and evaluating those errors. The thesis is guided by the following research questions:

RQ1. How can semantic errors in generated text be systematically categorised and annotated in a way that enables task-agnostic evaluation of semantic consistency in input-controlled generation tasks? This research question aims to establish a task-agnostic and consistent approach for categoris-

ing and annotating semantic errors that occur in input-controlled generation. The underlying hypothesis is that synthesising existing error annotation schemes and taxonomies into a consensus taxonomy, focused on core error types of content/meaning such as *omission*, *addition*, and *substitution*, can enable more reliable and generalisable semantic consistency evaluations across diverse NLG tasks. This is addressed in Chapter 3.

RQ2. How can human annotation be used to investigate semantic consistency errors in system-generated text, and what can it reveal about the kinds of semantic errors present in existing language generation systems? This research question explores the role of human annotation in systematically identifying semantic consistency issues in NLG system outputs. The hypothesis is that span-based human annotations, using the consensus taxonomy, can provide fine-grained diagnostic insights into the performance of generation systems. Furthermore, reproducing the human evaluation of a semantic error detection system using annotator cohorts can offer insights into the importance of detailed annotation guidelines, annotator agreement, and the overall reliability and generalisability of semantic error annotation practices. Chapter 4 addresses this through an annotation study across multiple systems.

RQ3. To what extent can automatic evaluation methods, including lexical, embedding-based, and distance-based similarity metrics and prompt-based large language models, reliably reflect human judgements of semantic consistency and non-semantic quality dimensions (e.g., fluency, coherence) in NLG outputs, and how generalisable are these methods across systems and evaluation conditions? This research question investigates whether automatic methods can serve as effective and scalable alternatives to human evaluation for assessing semantic consistency and other quality dimensions in input-controlled NLG. It investigates both similarity-based metrics, including lexical, embedding-based, and distance-based measures, and large language model-

based approaches. The underlying hypothesis is that while traditional metrics may exhibit limited alignment with human judgments, large language models, when carefully prompted, can serve as reliable and generalisable evaluators, producing quality assessments that closely align with human ratings across different systems, tasks, and evaluation setups. This is explored in Chapter 5.

The following section, Section 1.3, introduces the formal terminology and notational conventions adopted in the subsequent chapters.

1.3 Formal Foundations

Terminology

To support the investigation outlined in the previous section, this chapter establishes the formal foundations that underpin the thesis. Since the central focus is on evaluating semantic consistency in input-controlled natural language generation, this requires precise definitions of key terms such as *semantic error*, *semantic consistency*, and their associated annotation and evaluation procedures. These definitions form the conceptual basis for both the human and automatic evaluation methods explored in later chapters.

Semantic error: Also referred to as content/meaning error, Huidrom and Belz (2023) define semantic error as "the highest level of error category subsuming all errors in outputs that relate to the content/meaning of the output rather than its form." We follow this definition.

Semantic error is further divided into the following error types:

1. Omission: Huidrom and Belz (2023) define omission as "some content that is present in the input and should be rendered in the output is not present in the output. Moreover there is no content in the output that is intended to render it, but does so wrongly. That is, this type of error can be fixed by adding something to the output."

2. Addition: Huidrom and Belz (2023) define addition as "some content that is not present in the input and should not be rendered in the output is present in the output. Moreover there is no content in the input that it is intended to render, but renders wrongly. I.e. this type of error can be fixed by removing something from the output."

3. Substitution: Huidrom and Belz (2023) define substitution as "some content in the output, that is intended to convey some content that is present in the input, does it wrongly. This definition means that a substitution cannot equally be construed as the combination of an omission and an addition. This type of error can be fixed by replacing something in the output."

More in-depth details of semantic error types can be found in Chapter 3.

In addition, although QCET (Belz et al., 2024) does not provide an explicit definition of a semantic error, it implies that a semantic error occurs when an output text fails to preserve the input's intended content under the "Correctness → Content" criterion, that is, when the generated output omits, distorts or adds information that contradicts the source data.

Semantic consistency: Semantic consistency is defined as the property whereby all information conveyed by the output fully aligns with, and does not go beyond, the information present in the input.

While this definition focuses on a grounded, input-aligned interpretation, other definitions exist in the literature. For example, for Novikova et al. (2025), semantic consistency can be understood in two complementary senses. In the logical/formal view, semantic consistency is derived from the idea of semantic equivalence: "The idea of semantic consistency is derived from the semantic equivalence property, represented as f(X) = f(Y) if X and Y mean the same." That is, if two inputs X and Y carry identical meaning, then any function f (e.g., a model's generation) should produce identical outputs for both. In the non-logical/informal view, semantic consistency is often equated with factual or faithful behavior namely, avoiding halluci-

nations or contradictions with the source: "Factual inconsistency is often referred to as hallucinations and/or faithfulness, i.e. models generating new information that contradicts the source document (e.g., Tam et al., 2022; Maynez et al., 2020)."

Under this lens, an output is semantically consistent if everything it conveys can be traced back to, and does not conflict with, the original input. This thesis adopts the former definition above.

Semantic error detection: We define semantic error detection as the process of detecting semantic errors, either manually or automatically, in a given input–output pair.

This includes any type of semantic error (*omission*, *addition*, or *substitution*) regardless of whether the error is assessed in relation to a specific input. Detection may occur in isolated outputs (for e.g., evaluating text alone) or in input–output pairs when available.

Semantic error annotation/labelling: We define semantic error annotation (or labelling) as the process of marking or tagging detected semantic errors in both the input (data/text) and the corresponding output (text) once those errors have been identified.

Semantic consistency testing: We define semantic consistency testing as the process, manual or automated, of verifying that everything conveyed by the output is fully aligned with, and does not exceed, the information present in the input.

Semantic consistency error: We define a semantic consistency error as any discrepancy or mismatch between the information in the input and the content produced by the output, indicating that the output has introduced, omitted or altered meaning relative to the source.

Please note that while a semantic error refers more broadly to any content-level error in the output, regardless of whether the input is explicitly considered,

a semantic consistency error is a subset of semantic errors that arises in input-controlled generation settings, where the output is expected to be directly and fully grounded in a given input.

Semantic consistency error identification: We define semantic consistency error identification as the process of pinpointing those specific semantic errors that reflect inconsistency between the input and output for a given pair.

This task requires explicitly comparing the two and involves pinpointing where the output fails to accurately reflect the input, through *omission*, *addition*, or *substitution*. In contrast to general semantic error detection, this definition is limited to input-controlled generation settings and assumes the availability of both input and output.

Notation and formal setup for LLM-based evaluation analysis

This section introduces the notation and formal setup used to analyse correlations between human and LLM-generated ratings for semantic consistency and non-semantic consistency quality criteria. The formalism supports empirical analyses in Chapter 5, particularly in Sections 5.4 and 5.5, where the LLM-as-judge approach is evaluated against human scores across a range of input–output tasks. The goal is to quantify the alignment between human judgments and LLM-predicted ratings at both the item and system levels.

Let I denote the set of all items in a given task T, with $|I|$ items. For each item $i \in I$, evaluation criterion c, and system $m \in M$ (where M is the set of generation systems under evaluation), let $H^{(c)}_{i,T,m}$ represent the human-assigned score for criterion c and system m, and let $J^{(c)}_{i,T,m}$ represent the LLM-generated score. Each LLM score is obtained by aggregating three independently sampled outputs $j^{(s,c)}_{i,T,m}$, where $s \in \{42, 1234, 1738\}$ denotes the random seed used at inference time.

The aggregated LLM score is computed either via mean or majority vote:

$$J_{i,T,m}^{(c)} = \begin{cases} \frac{1}{3}\sum_{s=1}^{3} j_{i,T,m}^{(s,c)} & \text{(mean aggregation)}, \\ \text{majority}\{j_{i,T,m}^{(42,c)},\, j_{i,T,m}^{(1234,c)},\, j_{i,T,m}^{(1738,c)}\} & \text{(majority vote)}. \end{cases}$$

The mean system-level scores across all items for criterion c are then:

$$\bar{H}_{T,m}^{(c)} = \frac{1}{|I|}\sum_{i\in I} H_{i,T,m}^{(c)}, \quad \bar{J}_{T,m}^{(c)} = \frac{1}{|I|}\sum_{i\in I} J_{i,T,m}^{(c)}$$

To evaluate alignment between human and LLM ratings, Pearson's correlation coefficient $r_{H,J}^{(c)}$ is computed over systems in task T as follows:

$$r_{H,J}^{(c)} = \frac{\sum_{m\in M}\left(H_{T,m}^{(c)} - \bar{H}_T^{(c)}\right)\left(J_{T,m}^{(c)} - \bar{J}_T^{(c)}\right)}{\sqrt{\sum_{m\in M}\left(H_{T,m}^{(c)} - \bar{H}_T^{(c)}\right)^2}\sqrt{\sum_{m\in M}\left(J_{T,m}^{(c)} - \bar{J}_T^{(c)}\right)^2}}$$

Here, $\bar{H}_T^{(c)}$ and $\bar{J}_T^{(c)}$ denote the mean human and LLM scores across systems in task T, respectively. This coefficient reflects the strength of the linear relationship between human and LLM assessments for each evaluation criterion.

1.4 Thesis Structure

This thesis is structured to investigate, develop, and evaluate methods for assessing and controlling semantic consistency in input-controlled natural language generation (NLG), with a focus on both human and automatic approaches. The work is organised across six chapters, each addressing a distinct aspect of the research problem, from foundational definitions to empirical evaluation.

1. Chapter 1 introduces the motivation, scope, and objectives of the thesis. It outlines the central research questions and positions semantic consistency as a critical but under-addressed dimension of quality in NLG, especially in input-controlled tasks. Additionally, this chapter also defines key terminology and

formal concepts used throughout the thesis. It specifies how semantic consistency and semantic errors are understood in this work, and introduces notational conventions that support later experimental setups, particularly for LLM-as-judge evaluations.

2. Chapter 2 reviews existing literature on semantic consistency evaluation, focusing on error annotation schemes, automatic evaluation methods and the limitations of current practices.

3. Chapter 3 presents a consensus taxonomy of semantic errors developed through synthesis of existing categorisation schemes. It defines three core types of semantic error: *omission*, *addition*, and *substitution*, and establishes a task-agnostic framework for evaluating semantic consistency in input-output generation settings.

4. Chapter 4 includes (i) a survey of prior error annotation schemes and taxonomies on both human- and machine-generated text, (ii) a span-based human annotation study applying the top-level categories from the consensus taxonomy to outputs from multiple data-to-text systems.

5. Chapter 5 assesses the extent to which automatic metrics, including lexical, embedding-based, and distance-based similarity measures, correlate with human judgements of semantic consistency. In addition, it introduces a prompt-based large language model evaluation framework (LLM-as-judge) to assess system outputs across semantic and non-semantic quality dimensions, focusing on alignment with human ratings. The chapter further explores the reliability and generalisability of the LLM-as-judge method across tasks, systems, and prompt formulations. Finally, it investigates the use of LLMs as sanity checkers for human evaluation experiments, examining whether their outputs correlate with human judgments across repeated evaluation setups, thereby assessing the generalisability and reliability of LLM-as-judge.

6. Chapter 6 brings together and summarises the main contributions and findings

of the thesis. It reflects on the implications of the work for evaluating semantic consistency in NLG systems and outlines limitations and future research directions.

1.5 Publications

This thesis has so far resulted in the following list of publications:

1. Rudali Huidrom and Anya Belz. 2022. **A Survey of Recent Error Annotation Schemes for Automatically Generated Text**. In Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pages 383–398, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

2. Rudali Huidrom, Ondřej Dušek, Zdeněk Kasner, Thiago Castro Ferreira, and Anya Belz. 2022. **Two Reproductions of a Human-Assessed Comparative Evaluation of a Semantic Error Detection System.** In Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 52–61, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.

3. Rudali Huidrom and Anya Belz. 2023. **Towards a Consensus Taxonomy for Annotating Errors in Automatically Generated Text.** In Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing, pages 527–540, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

4. Rudali Huidrom, Anya Belz, and Michela Lorandi. 2024. **Differences in Semantic Errors Made by Different Types of Data-to-text Systems.** In Proceedings of the 17th International Natural Language Generation Conference, pages 609–621, Tokyo, Japan. Association for Computational Linguistics.

5. Rudali Huidrom and Anya Belz. 2025. **Ask Me Like I'm Human: LLM-based Evaluation with For-Human Instructions Correlates Better with Human Evaluations than Human Judges.** In Proceedings of the 4th Workshop on Table Representation Learning, Vienna, Austria. Association for Computational Linguistics.

6. Rudali Huidrom and Anya Belz. 2025. **Using LLM Judgements for Sanity Checking Results and Reproducibility of Human Evaluations in NLP.** In Proceedings of the 4th Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), Vienna, Austria. Association for Computational Linguistics.

7. Rudali Huidrom, Michela Lorandi, Simon Mille, Craig Thomson and Anya Belz. 2025. **Assessing Semantic Consistency in Data-to-Text Generation: A Meta-Evaluation of Textual, Semantic and Model-Based Metrics.** In Proceedings of the 18th International Conference on Natural Language Generation: Generation Challenges, Hanoi, Vietnam. Association for Computational Linguistics.

Chapter 2

Related Work

This chapter establishes the conceptual and methodological ground for the thesis. We (i) align terminology so later chapters are unambiguous; (ii) critically synthesise error taxonomies and annotation protocols; (iii) summarise task-specific evidence about semantic errors; and (iv) review human and automatic evaluation approaches with a focus on semantic consistency. Each section explains why it is here and how it informs the methods presented in Chapters 3 to 5.

2.1 Terminological Alignment

Definitions in the literature often diverge or conflate meaning-level with form-level phenomena. Here we standardise terms for use throughout the thesis so that: (i) later analyses are comparable across tasks/datasets; (ii) our consensus taxonomy in Chapter 3 has a clear top layer; and (iii) evaluation setups in Chapters 3–5 map unambiguously to those terms.

Throughout this thesis, we adopt the terminology defined in Chapter 1, Section 1.3. We use *semantic error* as an umbrella term for any content/meaning error in a generated output (as distinct from error of form). We subdivide semantic errors into three types: omission, addition, and substitution. Together, these form the top level of the consensus taxonomy developed in Chapter 3.

We define *semantic consistency* as the property whereby all information conveyed

Term in Prior Work	Mapped to Thesis Definition
Hallucination (Maynez et al., 2020; Ji et al., 2023; Biten et al., 2022; Xia et al., 2024; Bao et al., 2024)	*addition* (typically unsupported or fabricated content)
Extrinsic hallucination (Maynez et al., 2020)	*addition* (unsupported content)
Intrinsic hallucination (Maynez et al., 2020)	*substitution* (internally inconsistent or misrepresented content)
Factual error (Huang et al., 2020; Fraile Navarro et al., 2025)	*omission*, *addition*, or *substitution* (context-dependent)
Unsupported content (Durmus et al., 2020; Huang et al., 2024; Liu et al., 2024)	*addition*
Missing facts/content (Durmus et al., 2020; Liu et al., 2024; Bao et al., 2024)	*omission*
Deletion (in MT) (Al Sharou and Specia, 2022)	*omission*
Mistranslation (Al Sharou and Specia, 2022; Popović and Ney, 2007; Du et al., 2024)	*substitution*
Incorrect coreference (Zhang et al., 2023)	*substitution* (if meaning is distorted)
Fabricated claims/facts (Huang et al., 2020; George and Stuhlmüller, 2023)	*addition*
Contradiction (Dušek and Kasner, 2020a; Wang et al., 2024)	*substitution*
Entailment violation (Scialom et al., 2021; Kryscinski et al., 2020)	*addition* or *substitution* (depending on inference direction)
Semantic inconsistency (Novikova et al., 2025; Pagnoni et al., 2021; Zhang et al., 2024a)	Umbrella term referring to *omission*, *addition*, or *substitution*
Factual inconsistency (Maynez et al., 2020; Fraile Navarro et al., 2025; Wang et al., 2024)	Often used interchangeably with semantic inconsistency; may include all semantic error types
Critical/major/minor error (Fraile Navarro et al., 2025)	Mapped based on severity; underlying type can be *omission*, *addition*, or *substitution*
Hallucination due to knowledge misuse or source errors (Paudel et al., 2025)	*addition*, *substitution*, or *omission* depending on context

Table 2.1: Alignment of prior terminology with thesis semantic error definitions

by the output fully aligns with, and does not exceed, the information present in the input. A semantically consistent output, therefore, contains no semantic errors.

While some works use terms like "hallucinations," "factual errors," or "inconsistencies," we reinterpret and classify such terms in alignment with our definitions. To compare across studies, we map common labels to our terms (e.g., extrinsic hallucination to addition; intrinsic hallucination to substitution; factual inconsistency to omission/addition/substitution depending on context). Table 2.1 provides the compact mapping used throughout this thesis; details and examples appear in Sections 2.2–2.5 and Chapter 3.

This normalisation serves three purposes. First, it enables synthesis across subfields in which different labels map to the same underlying types (for e.g., extrinsic hallucination corresponds to addition, intrinsic hallucination to substitution). Second, it separates semantic error type, Omission, Addition, and Substitution, from other dimensions (for e.g., severity, locality, discourse-level scope), thereby avoiding category drift. Third, it provides a stable foundation for the subsequent chapters: Chapter 3 formalises a consensus taxonomy with semantic errors at the top level, Chapter 4 operationalises these types in a span-based human evaluation protocol, and Chapter 5 evaluates automatic and LLM-as-judge methods, interpreting criteria such as Data Coverage, Relevance, and Correctness through the lens of semantic consistency.

Consequently, this conceptual framing guides Section 2.2's comparison of taxonomies and annotation protocols, structures the discussion of task-specific evidence in Section 2.3, and provides the lens for the evaluation methods surveyed in Sections 2.4 and 2.5.

2.2 Error Annotation Schemes and Semantic Error Taxonomies

In this section, we group prior work on error annotation schemes and taxonomies into four strands.

General taxonomies (MT/NLG-rooted) Early frameworks such as MQM (Lommel et al., 2014) and linguistically motivated MT schemes (Costa et al., 2015), followed by later extensions (Specia et al., 2017) and SCATE (Tezcan et al., 2017), provide hierarchical breadth for manual evaluation. However, at lower levels these schemes frequently mix meaning and form, include task-bound leaves that limit portability, and offer limited guidance on linking errors to source content. In this thesis, they serve as canonical references for label normalisation (Table 2.1) and motivate the separation of error type from other dimensions developed in Chapter 3. Building on these insights, Chapter 3 abstracts the high-level distinctions into an Omission/Addition/Substitution top layer semantic error type, and Chapter 4 requires explicit input–output span-level annotation of semantic errors to avoid form–meaning conflation.

Task-specific faithfulness/hallucination frameworks Datasets and protocols in summarisation, dialogue, and QA sharpen the focus on unsupported or contradictory content. Rotowire introduced distinctions among omitted, hallucinated, and incorrect facts (Wiseman et al., 2017); subsequent work formalised the extrinsic versus intrinsic divide in summarisation (Maynez et al., 2020) and operationalised QA-based checks (FEQA; Durmus et al. (2020)). These contributions inform our mapping from task-specific labels to semantic error types (Omission, Addition, Substitution) and motivate the adoption of input-grounded checks in later chapters. Nonetheless, definitions vary across papers, substitution is sometimes conflated with addition, omission is often under-specified, and portability beyond the originating task is limited. Accordingly, Chapter 4 borrows the QA notion of answerability while retaining explicit span labels so omissions are captured, and Chapter 5 evaluates whether LLM-as-judge reproduces human system-level trends without inheriting label drift.

Span/unit-based protocols with alignment Recent work emphasises span localisation and, in some cases, explicit input–output alignment (e.g., OTTAWA;

Huang et al. (2024)) or span-annotated detection resources (HalluciNot; Paudel et al. (2025)). Sentence-level hallucination detection has also been explored across many language directions (Benkirane et al., 2024). These approaches demonstrate the diagnostic value of localisation and alignment, which we adopt in our human study. However, overlap/stacking rules for annotation are often underspecified, empty spans for omission/addition are inconsistently handled. In response, Chapter 4 specifies span-boundary rules (including overlaps and empty spans) and, mandates both input and output error annotation.

Broader taxonomies beyond task-specific schemes Beyond the task-specific annotation schemes discussed above, there exists a wider body of error taxonomies spanning MT, learner corpora, and general-purpose NLG. Representative examples include linguistically motivated MT taxonomies (Costa et al., 2015), statistical /critical-error schemes for MT (Federico et al., 2014; Al Sharou and Specia, 2022; Caseli and Inácio, 2020), general NLG/PLM resources such as TGEA (He et al., 2021) and PolyTope (Huang et al., 2020), and learner-corpus/TEC-oriented schemes (Belkebir and Habash, 2021; Ng et al., 2014; van der Goot et al., 2018; Korre et al., 2021; Caines et al., 2020; Macklovitch, 1991; Costa et al., 2012; Dickinson and Herring, 2008; Dickinson, 2010; Nagata et al., 2018; Lin et al., 2022; Barbagli et al., 2016; Gayo et al., 2016; Di et al., 2019). We include these works because they provide established label inventories and design patterns that we (i) inform the consensus error taxonomy developed in Chapter 2, and (ii) provide label mappings for terminology alignment (Table 2.1). However, as a basis for semantic-consistency evaluation they exhibit recurrent shortfalls: lower-level categories often conflate meaning and form; leaf nodes are frequently domain or learner-specific, hierarchies tend to intermix error type with severity, and input grounding/linking is rarely required. Consequently, we reuse only what transfers (label inventories and illustrative examples; full structures are catalogued in Appendices A.2.1 and A.2.2) and, where necessary, diverge from these frameworks: Chapter 3 distils a consensus error taxonomy with omission/addition/substitution as its high level semantic error types, Chapter 4

mandates span-level, input-output annotation, and Chapter 5 evaluates methods against semantic error aware human judgements.

To make these contributions comparable, we analyse schemes along four axes: scope and portability (task-agnostic vs. task-specific); granularity and hierarchy (depth; whether annotators label leaves or internal nodes; separation of type from severity/locality); unit of analysis (sentence, clause, or span; whether input–output links are mandated, for e.g., alignment as in Huang et al. (2024)); and operationalisation and reliability (instructions, training, inter-annotator agreement. Two consistent observations emerge. First, despite divergent labels, categories in practice collapse to semantic errors at the top level, suggesting a shared core that can be made explicit across settings (for e.g., Wiseman et al. (2017); Durmus et al. (2020); Maynez et al. (2020)). Second, span-level annotation with both input-output pairs is critical for diagnosing systems and for comparing studies, but it requires precise guidance to avoid double-counting and to handle omissions.

Existing schemes offer valuable distinctions but are often task-bound, conflate error type with other facets, or are operationally under-specified. These observations motivate the thesis's design choices. Chapter 2 adopts omission, addition, and substitution as the task-agnostic top layer of a consensus taxonomy, separating other facets into distinct dimensions. Chapter 3 implements a span-based, input-linked annotation protocol with explicit rules for overlaps and empty spans, and evaluates reliability. The next sections present task-specific evidence and trends in semantic errors in NLG, and provide an overview of human and automatic (including LLM-based) evaluation of semantic consistency.

2.3 Semantic Errors in Text Generation: Task-Specific Evidence and Trends

Semantic errors have been widely reported across diverse NLG tasks. These errors often arise despite fluent output, highlighting a frequent trade-off between fluency

and fidelity. Ji et al. (2023) provide a comprehensive taxonomy of hallucinations in NLG, distinguishing between intrinsic hallucinations, logical inconsistencies or contradictions within the output, and extrinsic hallucinations, which refer to content unsupported by the input. According to the definitions adopted in this thesis, these correspond to *substitutions* and *additions*, respectively.

In *abstractive summarisation*, Maynez et al. (2020) demonstrate that pretrained models frequently produce summaries that are fluent but unfaithful. Hallucinations are categorised as *extrinsic* (unrelated to the input, corresponding to *additions*) or *intrinsic* (contradictory content, corresponding to *substitutions*). Frequent *omissions* of salient input content are also observed. Dreyer et al. (2021) report that the rate of hallucinated errors increases with greater abstractiveness in summarisation, prompting the development of new evaluation metrics. Zhang et al. (2023) find that even extractive systems exhibit semantic inconsistencies such as incorrect coreference and fact misattribution. These are labelled as "semantic inconsistencies" in the study and map to *substitutions* under terminology definitions used in this thesis.

Recent work continues to document the presence of semantic errors in summarisation and related tasks. Xia et al. (2024) introduce an active learning framework that detects a broad spectrum of semantic errors, including frame-level mismatches and discourse-level inconsistencies (corresponding to *substitutions*), as well as unverifiable content (corresponding to *additions*). In a comprehensive benchmark, Bao et al. (2024) categorise hallucination types in LLM-generated summaries into unsupported claims (*additions*), distorted input facts (*substitutions*), and missing information (*omissions*). In the context of scientific summarisation, George and Stuhlmüller (2023) achieve state-of-the-art performance in detecting subtle *substitution* errors, particularly in nuanced factual rewritings. For multi-document summarisation, Belem et al. (2024) report increased hallucination rates with larger numbers of input sources, with up to 75% of content exhibiting either *additions* or *substitutions*, especially in later summary segments.

QA-based metrics such as FEQA (Durmus et al., 2020) and QAGS (Wang et al.,

2020) are used to detect semantic errors by comparing answers generated from system outputs to answers grounded in the source. These methods are particularly effective for identifying *omissions* and *additions* (for e.g., when answers cannot be verified from the source), as well as *substitutions* through mismatched or incorrect answers.

In *data-to-text generation*, structured inputs allow for clearer identification of semantic errors. Lebret et al. (2016) report that approximately 30% of generated outputs in the WikiBio dataset contain hallucinated content (*additions*). Dziri et al. (2021) propose a detailed taxonomy that includes *omissions*, *additions*, and *substitutions*. In this taxonomy, hallucinations are defined as content not supported by the input, mismatched values are considered *substitutions*, and missing facts are defined as *omissions*. The study also highlights limitations in n-gram-based metrics for capturing these semantic errors. Metrics such as PARENT (Dhingra et al., 2019), which compute precision and recall against both source input and reference text, demonstrate improved correlation with human fidelity judgments.

Semantic accuracy remains a challenge in dialogue generation. Ji et al. (2023) again identify intrinsic hallucinations (*substitutions*) and extrinsic hallucinations (*additions*) in open-domain dialogue. Dziri et al. (2021) introduce a knowledge-grounded refinement model that validates generated content against external knowledge bases, effectively reducing *additions*. Yu et al. (2024) use causal analysis and counterfactual reasoning to trace the influence of noisy or misaligned input knowledge on semantic errors, particularly *additions*. A lightweight mitigation strategy is introduced to reduce hallucinated content while maintaining output fluency. Wan et al. (2024) propose sequence-level certainty metrics and Certainty-based Response Ranking (CRR), a decoding strategy that improves output fidelity by favouring responses less likely to contain *substitutions* or unsupported content (*additions*).

In *machine translation*, semantic errors often manifest as *omissions* (e.g., dropped phrases) or *substitutions* (e.g., mistranslations). Popović and Ney (2007) conduct a fine-grained analysis of Word Error Rate (WER), identifying dropped content as

omissions. While COMET (Rei et al., 2022) and BERTScore (Zhang et al., 2019) improve adequacy judgments through contextual embeddings, these methods do not explicitly categorise semantic errors. Zhang et al. (2024a) introduce a synthetic hallucination dataset for machine translation, improving detection of *substitution* errors. Du et al. (2024) propose a self-consistency scoring method for classifier training using unlabeled data to detect *omissions* and *substitutions*, enabling scalable evaluation without manual annotation.

In *vision-language generation*, such as image captioning, *additions* are frequently observed when models hallucinate objects not present in the input image. Rohrbach et al. (2018) introduce the CHAIR metric to measure object hallucinations and explicitly define these as *additions*. Biten et al. (2022) apply object-centric data augmentation to reduce hallucinations, also framing these errors as *additions*. However, Geigle et al. (2024) find that object grounding signals alone do not reliably prevent hallucinated mentions of non-existent objects, highlighting the persistent challenge of *additions* in vision-language generation.

Having explored the presence of semantic errors across diverse generation tasks, the following section discusses how semantic consistency is evaluated and how it is reflected in generated outputs.

2.4 Human Evaluation of Semantic Consistency

Human evaluation remains the gold standard for assessing semantic consistency in natural language generation due to its ability to incorporate contextual, domain-specific, and pragmatic understanding. However, the design of human evaluation protocols varies considerably, and most studies assess consistency at the sentence or system level, rather than identifying fine-grained error spans.

We structure the review this way to surface the main design choices and to motivate our own setup: we adopt rating-based, criterion-specific evaluations with trained (expert) annotators and evaluate LLM-as-judge configurations (Chapters. 4–5), while question-answering protocols are reviewed only for contrast and are not

used in our experiments.

Rating-Based Evaluations

Rating-based methods are among the most common approaches to human evaluation, typically relying on Likert scales or categorical judgments applied to sentence- or system-level outputs. In one of the most widely cited setups, Fabbri et al. (2021) apply a 5-point Likert scale to rate summaries across multiple dimensions, including fluency, coherence, and consistency. In this context, consistency is defined as factual alignment between the generated summary and the source document. Although specific error types are not annotated, this definition implicitly encompasses semantic errors such as unsupported content (*addition*), factual misstatements (*substitution*), and missing salient information (*omission*).

Building on the need for finer distinctions, Maynez et al. (2020) introduce a targeted annotation protocol for abstractive summarisation. This study classifies hallucinations into two categories: extrinsic hallucinations, which refer to content not inferable from the source (*addition*), and intrinsic hallucinations, which involve contradictions with the input (*substitution*). Although omissions are not systematically annotated, the absence of key information is recognised as a critical shortcoming in generated summaries.

Recognising the limitations of Likert scales in capturing subtle inconsistencies, Tang et al. (2022b) compare rating-based approaches to ranking-based alternatives, particularly Best-Worst Scaling (BWS). The study finds that BWS improves inter-annotator agreement and reliability, especially when multiple quality dimensions such as fluency and factuality must be disentangled. Although error types are not explicitly categorised, the results underscore the value of structured comparative formats in highlighting semantic inconsistencies.

QA-Based Consistency Evaluation

Question-answering (QA) protocols offer a complementary approach to rating methods by introducing a verification mechanism grounded in information retrieval. Instead of directly scoring generated outputs, these methods evaluate consistency by formulating questions based on the summary and verifying whether correct answers can be found in the source. QAGS (Wang et al., 2020), for example, generates cloze-style questions from named entities and noun phrases in the summary. When the source fails to support an answer, the output is considered hallucinated (*addition*); when the source yields a conflicting answer, the error is treated as a factual misrepresentation (*substitution*).

FRANK (Pagnoni et al., 2021) extends this paradigm by offering a unified benchmark that incorporates several QA-based and model-based evaluation techniques. The authors annotate specific error types such as entity hallucination (*addition*) and entity distortion (*substitution*), based on whether the generated content is unsupported or misaligned with the input. Although omission errors are not systematically annotated, qualitative analysis acknowledges the impact of missing content on perceived fidelity.

By grounding evaluation in answerability, QA-based protocols facilitate more targeted assessments of factual consistency than general-purpose rating scales. These methods are particularly effective when outputs are information-dense or contain multiple claims, and when direct reference comparisons are unavailable.

Expert vs. Crowd Evaluations

The reliability of human evaluation often depends on the background and expertise of annotators. Several studies show that domain expertise significantly enhances the ability to identify semantic errors, especially in specialised or high-risk settings. In the domain of clinical summarisation, Fraile Navarro et al. (2025) compare evaluations performed by crowdworkers and medical experts. The study introduces a four-level annotation scale: entailment, minor error, major error, and critical error,

to assess the fidelity of radiology report summaries. Critical and major errors often correspond to serious hallucinations (*addition*) or factual inaccuracies (*substitution*), while clinically significant omissions are also noted and flagged as a key challenge in automated summarisation.

Similarly, Lango et al. (2024a) examine how annotator background and task framing affect evaluation outcomes in dialogue summarisation. This study compares single-criterion (e.g., overall consistency) and multi-criterion (e.g., factuality, relevance, coherence) protocols and finds that both the level of expertise and the specificity of evaluation criteria influence the types of semantic inconsistencies detected. Although semantic errors are not explicitly annotated, the findings demonstrate that omissions and contradictions are differently interpreted depending on annotator familiarity and protocol design.

These findings collectively underscore the importance of evaluator expertise and protocol structure in achieving accurate and reliable semantic consistency assessments.

LLM-as-Judge Comparisons

With the advent of large language models (LLMs), recent work has explored the feasibility of using LLMs as automated evaluators of generation quality. Human evaluation is often used as a benchmark to assess the alignment of model-based judgments with expert annotations. Judge-Bench (Bavaresco et al., 2024) offers a systematic evaluation of LLM-as-judge capabilities across 20 tasks, including summarisation, question answering, and classification. The study compares LLM-based ratings to expert judgments, particularly on dimensions such as factual consistency.

Although the benchmark does not focus exclusively on semantic consistency, the results indicate substantial discrepancies between LLM and human evaluations in tasks requiring fine-grained factual accuracy. The errors that LLMs struggle to detect most consistently are those corresponding to unsupported content (*addition*) or factual distortions (*substitution*), particularly when domain knowledge or multi-

sentence reasoning is required.

2.5 Automatic Evaluation of Semantic Consistency

Automatic evaluation methods offer scalable and efficient alternatives to human judgments, though they vary significantly in their ability to capture semantic consistency. Broadly, these methods fall into four categories: entailment-based methods, QA-based approaches, uncertainty-based metrics, and semantic scoring/reranking models. Each targets specific types of semantic error, *omission*, *addition*, and *substitution*, with varying levels of granularity.

Entailment-Based Methods

Natural Language Inference (NLI) has become a foundational tool for identifying inconsistencies between source input and generated output. By evaluating whether output content is logically entailed by the input, NLI enables direct assessment of semantic alignment. For instance, Dušek and Kasner (2020a) apply NLI to meaning representations (MRs) and generated texts, measuring whether all output facts are derivable from the input. This approach supports detection of unsupported content (*additions*) as well as missed content from the MR (*omissions*).

Building on this, FactCC (Kryscinski et al., 2020) trains a supervised entailment model using summaries that have been synthetically perturbed to introduce semantic errors. The types of perturbations include entity mismatches, factual contradictions, and fabricated content. These categories map to *substitutions* and *additions* in the present taxonomy, making the method suitable for evaluating multiple forms of inconsistency.

To improve scalability and adaptability across domains, SummaCConv (Laban et al., 2022) leverages convolutional pooling to aggregate NLI scores over all source–summary sentence pairs. Although the framework does not explicitly tag individual error types, inconsistencies are flagged based on lack of entailment or

contradiction signals, thus implicitly encompassing *omissions*, *additions*, and *substitutions* as determined through model inference.

QA-Based Consistency Metrics

Question-answering (QA) frameworks provide an alternative lens on semantic evaluation by measuring how well the generated text preserves the factual answerability of the source content. Instead of directly comparing text segments, these approaches generate questions from one side (either the summary or the input) and verify whether the answers can be recovered from the other. In QuestEval (Scialom et al., 2021), questions generated from the summary are answered using the source text. When questions cannot be resolved from the source, the associated content is marked as hallucinated (*addition*). Conversely, if information expected from the source is absent in the summary, the corresponding failure is considered an *omission*. Discrepancies in fact rendering are treated as *substitutions*.

SummEQuAL (Liu et al., 2024) extends this model by introducing a reference-free QA framework based on large language models. Questions are generated from both the input and the output, and factual consistency is judged based on mutual answerability. When content in the input fails to be represented in the output, the error is marked as an *omission*. Conversely, if output-generated content cannot be traced back to the input, the inconsistency is categorised as an *addition*. Incorrectly rendered facts are mapped to *substitutions*. This approach increases coverage while minimising dependence on gold references.

Uncertainty and Confidence-Based Metrics

Confidence-based evaluation methods assess semantic consistency by estimating the certainty with which a model outputs factual statements. Wang et al. (2024) introduce an uncertainty-aware metric for radiology report generation that combines entailment scores with model confidence. The proposed method uses a clinical-domain NLI model to compute entailment probabilities between generated content

and the gold-standard report. Lower confidence scores are found to correlate with clinically relevant *omissions* and *substitutions*, enabling automatic flagging of semantically risky content. This approach is particularly suited to sensitive domains where factual precision is critical.

Semantic Scoring and Reranking Models

Beyond evaluation, some models actively incorporate semantic consistency into the generation process through scoring or reranking strategies. DataTuner (Harkous et al., 2020a), for instance, integrates a semantic fidelity score to rank alternative outputs in data-to-text generation. The scoring model is trained to favour outputs that closely align with structured input data, thereby indirectly minimising *omissions*, *additions*, and *substitutions*. Although this method does not label individual errors, its objective function encourages factual correctness at the output level.

LLM-as-judge approaches have also emerged as scalable evaluators of consistency and related qualities. Judge-Bench (Bavaresco et al., 2024) benchmarks a variety of LLMs on tasks such as summarisation and QA, comparing their judgments with those of human annotators. Results indicate that LLMs achieve moderate alignment with human ratings, but often fail to detect nuanced factual distortions, especially domain-specific *substitutions* and unsupported *additions*. The performance gap highlights the continued need for human oversight and domain calibration in high-risk generation contexts.

2.6 Chapter Summary

This chapter reviews existing work in NLG, focusing on error annotation schemes and error taxonomies, task-specific error patterns and methods for identifying and evaluating semantic consistency in text generation. Although semantic errors are reported across different tasks such as, summarisation, data-to-text generation, machine translation, and question answering, the terminology used to described these

errors vary significantly, often lacking standardisation and clarity.

While several taxonomies have been proposed, they differ in structure, granularity and task dependence, thereby limiting comparability and generalisability. Also existing evaluation protocols, both human and automatic, exhibit considerable variation in scope and reliability. Human evaluations remain the gold standard but are costly and inconsistent across studies. Automatic methods, although scalable, often lack sensitivity to subtle semantic discrepancies and are not grounded in consistent taxonomic frameworks.

In addition to evaluation, a growing body of work has emerged that focuses on ensuring semantic consistency during the generation process. These methods, ranging from knowledge-grounded refinement and constrained decoding to certainty-based reranking and synthetic error training, proactively reduce the likelihood of semantic errors. However, such strategies are often task-specific and are not informed by a unified evaluation or annotation framework, making cross-task comparisons and integration difficult.

Taken together, this chapter has provided a critical synthesis of prior work and established the definitions and design choices used throughout the thesis. The next chapters operationalise these decisions: Chapter 3 formalises a task-agnostic consensus taxonomy (Omission, Addition, Substitution); Chapter 4 applies it in a span-based human annotation study and reports agreement and reproduction results; Chapter 5 evaluates automatic metrics and LLM-as-judge setups against the same criteria.

Chapter 3

Mapping out the Evaluation Space for Automatically Generated Text

3.1 Background

Error analysis and error type annotation are widely considered important in the context of textual generation, with their aim to provide insights into system errors on different levels of granularity across various NLP tasks (Popović and Burchardt, 2011; Costa et al., 2015). NLP as a field has a long-standing practice of error analysis and error type annotation (Macklovitch, 1991; Costa et al., 2015; Rivera-Trigueros, 2021). These practices not only help in improving system performance but also in improving evaluation methods.

Errors of content, as opposed to errors of form or lexical choice, are becoming more prevalent, given the recent advent of neural methods which are prone to such error types than the traditional rule-based systems (Huidrom and Belz, 2023). Documenting and analysing errors made by different systems can help in improving the semantic correctness (Adequacy in machine translation (MT)) of the generated text. Moreover, identifying and annotating these errors helps us better understand NLP system performance. (van Miltenburg et al., 2021).

There are various existing error taxonomies available that are tailored to different

types of NLP tasks. However, a large variety of such error taxonomies and error annotation schemes are task and/or domain-specific in nature (Huidrom and Belz, 2022). This makes the comparison between the output annotations difficult, making it more difficult to use them to generally improve model performance.

In this chapter, we propose a standardised, task-agnostic error annotation taxonomy that will allow comparing different NLP system outputs as part of performance analysis. This taxonomy can be used as a primary, off the shelf error annotation scheme for human evaluation of any task. Additionally, it opens the door to developing automatic and/or semi-automatic metrics for various NLP tasks (van Miltenburg et al., 2021). We address developing automatic methods for our thesis in Chapter 5.

The content of this chapter has been previously reported in our following publications:

1. Huidrom and Belz (2022)

2. Huidrom and Belz (2023)

3.2 Chapter Overview

We explore to what extent a standard has evolved in current error annotation schemes. The primary objective is to look into whether or not enough consensus is present in current error annotation schemes and taxonomies to use them as the basis for a standardised task-agnostic consensus taxonomy for content/meaning errors. Our explorations have resulted in the following contributions:

1. A systematic survey of error annotation schemes comprising content/meaning error types (Section 3.3 and Table 3.1);

2. A collated list of all content/meaning error type definitions found in the papers in the survey (Appendix A.1);

3. The minimally merged taxonomy comprising all non-task and non-domain-specific error types from the above list (Section 3.4.3 and Figure 3.1);

4. A standardised and generalised taxonomy of content/meaning error types derived directly from the minimally merged taxonomy (Section 3.4.4 and Figure 3.2), which is applicable across different input-controlled language generation tasks[1] and application domains.

3.3 Systematic survey of error annotation schemes

This section reports the systematic survey of error annotation schemes that underpins the consensus taxonomy in Section 3.4. Unlike Section 2.2's high-level critique, here we specify the paper filtering and selection, provide paper summaries and results, then collate definitions used to derive the minimally merged taxonomy (shown in Fig. 3.1) and the maximally merged taxonomy (shown in Fig. 3.2) This section is therefore the evidence base for the taxonomy presented next. For complete details, please see Appendix A.1.

3.3.1 Paper filtering and selection

We aim to identify a set of papers that report error annotation schemes and taxonomies of any size and depth for content/meaning error types which can serve as a basis for deriving a consensus error taxonomy.

First, we proceed by selecting all relevant papers from our literature survey on error types in machine and human generated text described in Huidrom and Belz (2022) and from Section 4.3 of Chapter 4 that describe error annotation schemes and/or taxonomies for content/meaning errors. This resulted in seven papers.

Second, we further expand the paper selection by searching the ACL Anthology[2] for papers that contained both the terms "accuracy error"[3] and "taxonomy" (i.e.,

[1]Tasks where the output content is wholly or largely determined by the input, in contrast to free text generation tasks, where the output is guided (but not determined) by a prompt.

[2]`https://aclanthology.org`

[3]We use accuracy in the MQM sense (content/meaning errors; omission, addition, distortion, as

a conjunctive query) which yielded 15 papers. We included papers published up to the year 2022. We manually examined these 15 papers and selected the five that described error annotation schemes and/or taxonomies for content/meaning errors. Of these five papers, three used the same error taxonomy named SCATE (Tezcan et al., 2017); therefore, we included only one paper on SCATE taxonomy (Tezcan et al., 2017). In total, this step yielded three papers.

Third, we added one paper Specia et al. (2021) which was cited by Al Sharou and Specia (2022), one of the seven papers selected from our prior survey (Huidrom and Belz, 2022). We also added four relevant papers which we were already aware of Thomson and Reiter (2020); Tang et al. (2022a); Kasner and Dusek (2022); Popović (2020). Here, Popović (2020) is a rare paper in that it uses the top-level content/meaning error type in annotation (see Section 3.4; Adequacy, Accuracy).

3.3.2 Paper summaries

This section provides brief summaries of the selected 15 papers. We do not include the original MQM framework (Lommel et al., 2014) as a separate paper here; instead, MQM is represented via some of the MT/NLG papers listed below that adopt MQM framework for error annotation.

Costa et al. (2012) provide a corpus of 6,000 questions that have been manually translated into Portuguese. Error annotation addresses two types of errors that arose during the manual translation: semantic-level errors and structure-level errors.

Federico et al. (2014) propose a statistical framework to analyse the impact of different error types, employing linear mixed models for machine translation. The experiments are designed for English as the source language and languages that are distant from English as the target language. The paper uses a set of four error classes which partially overlap with those used by Vilar et al. (2006): reordering errors, lexicon errors, missing words, morphological errors.

defined in Section 4.3.2), and because terminology in the literature is heterogeneous (e.g., adequacy, faithfulness, factuality).

Costa et al. (2015) introduce a linguistically motivated taxonomy of errors in machine-translated text. The taxonomy has five high-level error categories: Orthography, Lexis, Grammar, Semantic, and Discourse.

Specia et al. (2017) present a large-scale machine translation (MT) dataset that combines various degrees of human annotation with automatically recorded productivity features. Errors are annotated using the Multidimensional Quality Metrics (MQM) error annotation framework (Lommel et al., 2014). The errors are broadly categorised into three main categories: Accuracy, Fluency and Terminology. Additionally, these errors are populated with detailed error categories from MQM.

Tezcan et al. (2017) introduce the SCATE (Smart Computer-aided Translation Environment) MT error taxonomy, which is hierarchical and categorises errors into Accuracy errors (detected by examining both source and target sentences), and Fluency errors (relating to the wellformedness of the target sentence, regardless of content or meaning).

Caseli and Inácio (2020) address error analysis of neural MT (NMT) system outputs for Brazilian Portuguese, comparing the errors made by the NMT system with those made by a phrase-based machine translation (PBSMT) system. The error analysis adopted by the paper extends the taxonomy put forward by Martins and Caseli (2015), which consists of four broad error categories: syntactic errors, lexical errors, n-gram, reordering errors.

Popović (2020) introduces a manual evaluation method for MT outputs which marks up errors in the translated text. The proposed method uses two quality criteria: Comprehensibility and Adequacy. Comprehensibility refers to the degree to which a translated text can be understood (as distinct from fluency). Adequacy refers to the degree to which the translation conveys the meaning of the original source text. These error types each subdivide into Major and Minor, which refers to the severity.

Huang et al. (2020) introduce PolyTope, a set of eight metrics for Accuracy and Fluency error types, designed to quantify primary errors for 10 representative models

for text summarisation. Accuracy-type errors occur when a target summarisation does not match or accurately reflect the source text, while Fluency-type errors relate to linguistic properties of the text that are independent of how source and target relate. These categories subdivide into three levels of severity: Critical, Minor and Major.

Thomson and Reiter (2020) propose a methodology for gold-standard accuracy evaluations in texts generated by data-to-text systems. There are six main categories: Incorrect Number, Incorrect Named Entity, Incorrect Word, Context Error, Not Checkable and Other Error.

Specia et al. (2021) report the WMT 2021 Shared Task on Quality Estimation, where the aim is to predict the quality of outputs of neural machine translation (MT) systems at the word and sentence levels. Three main categories of meaning deviation are involved: Mistranslation, Omission and Hallucination. For each meaning deviation category, there are five critical errors. Annotators are instructed to ignore minor grammatical or typographical errors.

Mahmud et al. (2021a) report a comprehensive qualitative and quantitative comparative analyses of recently proposed source code summarisation models. A taxonomy of different error types across various models is used, with seven top-level categories: Missing Context, Missing Information, Incorrect Semantic Information, Incorrect Construction, Consistent with Ground Truth, Extraneous/Unnecessary, and Over-generalisation.

Zou et al. (2022) explore the effect of translation briefs and search conditions on the quality of post-editing performed by participants with varying levels of translation expertise, using the error categorisation scheme adopted by the American Translators Association (ATA).[4] Mistranslations and addition/omission errors fall under a single Accuracy error type, while usage, grammar and others fall under Fluency. Each category has two levels of severity: Accuracy_Critical, Accuracy_Minor, Fluency_Critical and Fluency_Minor. Note that in this study, errors of omission and

[4] `https://www.atanet.org/certification/how-the-exam-is-graded/error-categories/`

addition are (unusually) treated as the same error type, rather than two different types.

Al Sharou and Specia (2022) adds two new categories of critical errors to that defined by Specia et al. (2021): deviation in instructions and other critical meaning deviation.

Tang et al. (2022a) investigate factual errors in summarisation system outputs, in the context of which they unify nine existing factual error annotation schemes into a single-level hierarchical typology. The latter distinguishes errors on a number of different dimensions, of which just two are used in the reported work: intrinsic (misrepresented words from the source text) vs. extrinsic (added words not in the source text) errors, involving a noun phrase vs. a predicate.

Kasner and Dusek (2022) present a zero-shot alternative for data-to-text generation using ordering, aggregation, and paragraph compression. A manual error analysis is performed using five error types: Hallucination, Incorrect Fact Merging, Omissions, Redundancy, Grammar Error, and Disfluency.

3.3.3 Results

Paper and Taxonomy Name (where named)	Language Generation Task	# Error types	# Leaf nodes	Depth
Costa et al. (2012)	Machine Translation [MT]	11	9	2
Federico et al. (2014)	Machine Translation [MT]	5	4	1
Costa et al. (2015)	Machine Translation [MT]	36	25	4
Specia et al. (2017)	Machine Translation [MT]	21	15	4
Tezcan et al. (2017), SCATE	Machine Translation [MT]	45	33	4
Caseli and Inácio (2020)	Machine Translation [MT]	17	12	2
Popović (2020)	Machine Translation [MT]	7	4	2
Huang et al. (2020), PolyTope	Text Summarisation [TS]	11	8	2
Thomson and Reiter (2020)	Data-to-Text Generation [D2T]	7	6	1
Specia et al. (2021)	Machine Translation [MT]	19	15	2
Mahmud et al. (2021a)	Textual Summarisation of source code [TS(SC)]	39	31	2
Zou et al. (2022)	Machine Translation [MT]	5	4	1
Al Sharou and Specia (2022)	Machine Translation [MT]	25	21	2
Tang et al. (2022a)	Text Summarisation [TS]	19	8	5
Kasner and Dusek (2022)	Data-to-Text Generation [D2T]	6	5	1
Minimally merged error taxonomy	Task-agnostic	40	30	4
Maximally merged consensus error taxonomy	Task-agnostic	15	11	3

Table 3.1: Overview of properties of the error annotation schemes that form the basis of the merged taxonomies presented in this chapter (last two rows).

Table 3.1 presents an overview of the final set of selected 15 papers, ordered by year of publication, and providing information about authors, language generation task,[5] number of error types, number of leaf nodes and depth of the taxonomy. The **number of error types** is the number of nodes in the tree including the root. For example, the (complete) error annotation scheme used by Popović (2020) is (error → (comprehensibility → (major, minor)), → (adequacy (major, minor))), and we count that as 7 different error types.

The **number of leaf nodes** is simply the number of terminal nodes in a taxonomy, 4 in the above example. Note that in some cases, both internal and leaf nodes are used in annotation, in other cases just leaf nodes. The **depth of the tree** is the longest path from the root to a leaf. In the above example, the depth is 2. If there is no underlying hierarchical structure, then depth=1 (as we always assume a default top-level root error category, even if an explicit one is not included).

Building on our systematic survey, we propose a task-agnostic consensus error taxonomy of content/meaning error type which we discuss in detail in the following section. We derive a single hierarchy of error types by merging only those error categories that are identical in scope across papers while retaining task-specific leaf nodes only when they capture genuinely distinct phenomena (Fig. 3.1), and, where needed for comparability, consolidating error categories that that denote the same phenomenon (Fig. 3.2). We present both minimally merged (fine-grained) and maximally merged (coarse) variants of the taxonomy, along with the orthogonal error type schemes.

3.4 Consensus Taxonomy of Semantic Error Types

3.4.1 General error concepts

Our proposed consensus error taxonomy is intended for input-controlled text generation rather than free text generation. In the case of the former, only con-

[5]Note that our taxonomy is task-agnostic.

tent/meaning from the input must be present in the output, and all content in the input must be present in the output, except in contexts where only task-relevant parts of the input are required (e.g. in Summarisation and arguably also in Paraphrasing). What constitutes an error is therefore relatively clear in input-controlled text generation. If we think of the output as rendering the input, errors in input-controlled text generation are mismatches between input and output, where the input (1) is missing something (often referred to as an error of *Omission*), (2) adds something it shouldn't (error of *Addition*), or (3) renders something from the input wrongly (error of *Substitution*). Definitions of these and other error types are provided in the following section.

It is much less clear what constitutes an error in free text generation. Factual incorrectness and faulty common-sense reasoning are at the clearer end of the spectrum, but deviation from an intended reference continuation and relevance to the prompt are less clear to judge or measure. The term 'hallucination' is often used as something of a coverall term for anything that is undesirable in the output in free text generation.

In contrast, in input-controlled text generation, factual incorrectness or common-sense faults have no relevance; what matters is whether what is in the output can be justified by (a) overlap between input and output content, and (b) whether the given NLP task requires all content in the input to be rendered, or just part of it.

In other fields such as psychology, the term 'hallucination' is defined e.g. as "a percept, experienced by a waking individual, in the absence of an appropriate stimulus from the extracorporeal world" Blom (2010). Because of its association with mental health conditions, using the term for errors made by a computational system is controversial, and we prefer to use the more sober 'addition error' or just 'addition.' Recent LLM work also recommends reframing such phenomena as 'confabulation' (Sui et al., 2024). We stick with the task-grounded term 'addition' here, while noting this alternative terminology.

Omission errors are also a recognised phenomenon in neuroscience, defined e.g. as

"infrequent errors consisting in missing responses to the target stimuli" (Perri et al., 2017) which is fairly close to how the concept is used in NLP error assessment.

Section 3.4.2 provides an overview of the consensus taxonomy. Sections 3.4.3 and 3.4.4 detail the two construction steps of minimally and maximally merged taxonomies. Section 3.4.5 outlines how the final consensus taxonomy can be used in practice.

3.4.2 Overview of taxonomy

We present a consensus taxonomy of errors of meaning and content for use in error annotation and analysis that is based on a representative sample of existing taxonomies and is agnostic with respect to NLP task and domain. We proceed towards this goal in two steps: (1) directly deriving a single hierarchy of error types from our sample of existing taxonomies, minimally merging only those categories that are identical in scope (even if a different category name is used); (2) merging further error categories that are very similar (but not necessarily identical) in scope, yielding what we call a maximally merged taxonomy which standardises over, and encodes the consensus among, the original error type schemes.

The error taxonomies that form the starting point for our process of consensus identification often address both errors of content/meaning, and errors of form. We only use the former, although the orthogonal error types below (Section 3.4.4) can in principle apply to errors of either form or content/meaning.

We draw the line between the two as follows. Errors of content/meaning (in input-controlled NLG) refer to cases where the information conveyed by the output differs from the information conveyed by the input. They are defined relative to the input, hence can only be identified with reference to the input. Errors of form in NLG in general refer to flaws or mistakes in how the word sequence in the output is put together (rather than what it means), e.g. grammatical errors, disfluencies, or inappropriate style.

In this thesis, we only consider errors of content/meaning. We use content/meaning

error type as an umbrella for meaning-level errors across the literature, subsuming MQM Accuracy and terms such as adequacy, faithfulness, factuality, and fidelity, because many of these labels are overloaded or used inconsistently (e.g., "accuracy" as a performance metric). To avoid ambiguity, we standardise on the high-level content/meaning error type, mapping to our task-agnostic error types, Omission, Addition, and Substitution.

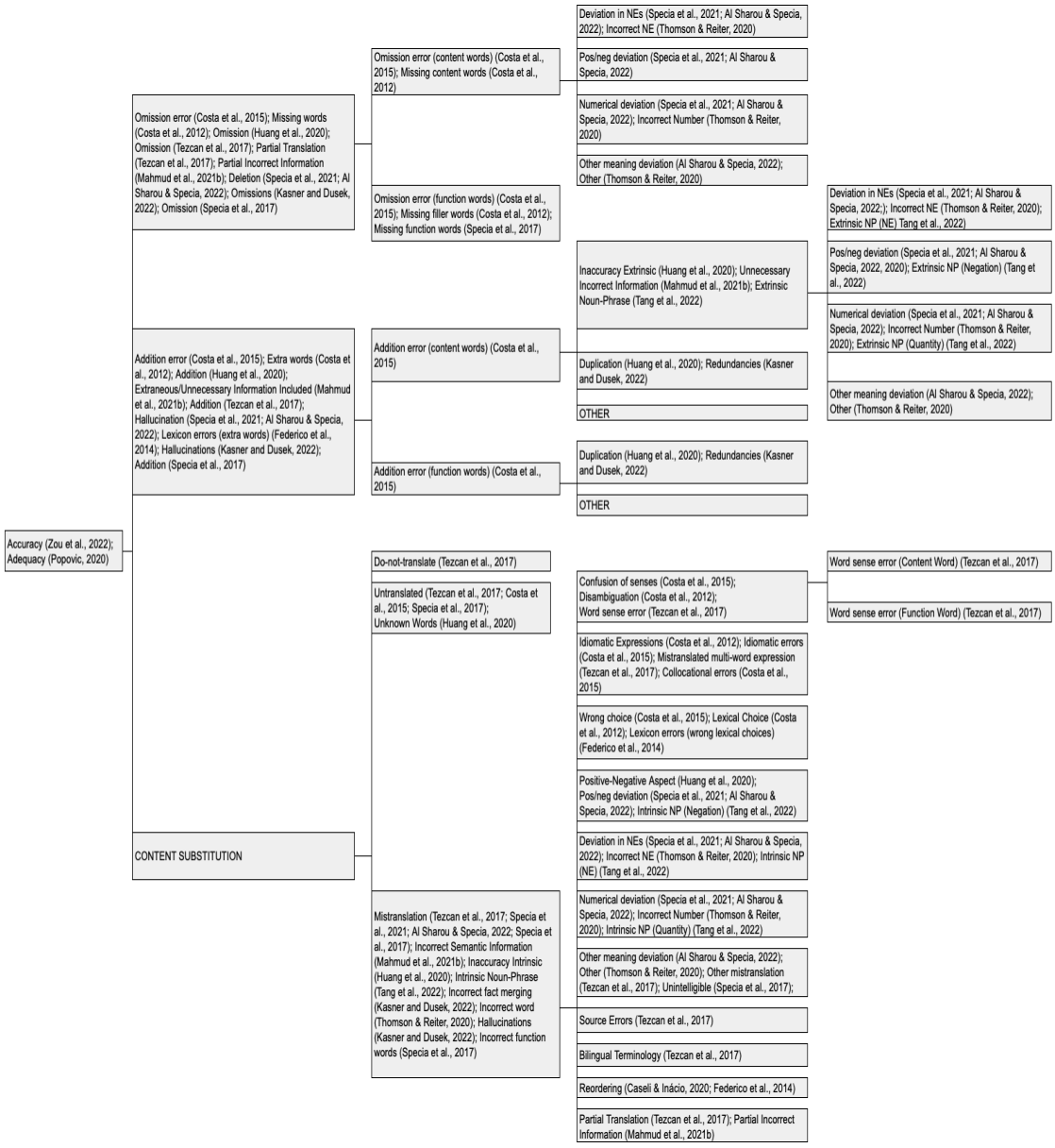

Figure 3.1: The minimally merged taxonomy of categories of errors in data-to-text and text-to-text generation (see Appendix A.1 for definitions of error categories). Note that we have left off some subclasses (see in text for details).

3.4.3 Minimally merged error taxonomy

As our starting point we collated all error categories along with their definitions where available from all of our 15 papers (see Appendix A.1). We removed those categories that relate to errors in the form, rather than the content, of outputs. Furthermore, we removed highly task or domain-specific categories, e.g. Missing Programming Language Information in code-to-summary generation Mahmud et al. (2021b), and Toxicity-introducing Error in catastrophic error detection Al Sharou et al. (2021); Specia et al. (2021).

For the remaining error categories we then grouped those together that we took to refer to the same error phenomenon, and arranged the resulting groups in superset/subset relations. This gave us what we refer to as our minimally merged taxonomy, shown in Figure 3.1. Each node in the hierarchy in Figure 3.1 shows the original names of the error categories and the papers we extracted them from. For the definitions provided in the original paper for each of these error categories, see Appendix A.1. We added two error categories (Content Substitution and Other) to ensure completeness and balance in the taxonomy.

For space reasons, in the diagram we are not showing subcategories that refer purely to (i) whether the error relates to a single word vs. multiple words Caseli and Inácio (2020); (ii) whether the error was major/critical vs. minor; (iii) which syntactic category the error related to (e.g. part of speech); and (iv) whether the error concerns function word(s) or content word(s). We return to these four sets of subcategories in the next section.

As can be seen from Figure 3.1, there is considerable consensus about the higher up categories, where we found up to ten papers using the same error category, albeit often under different names. In the next section, we develop the consensus further, generalising and creating single labels for sets of error names, to create a maximally merged version of the taxonomy.

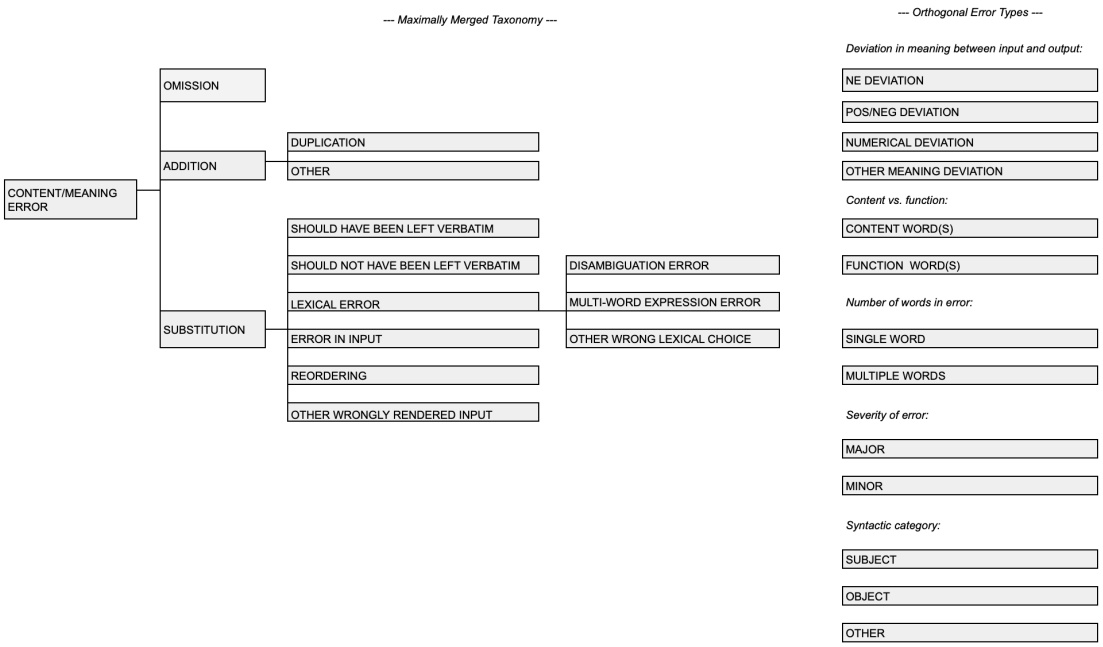

Figure 3.2: The maximally merged consensus taxonomy of categories of errors in data-to-text and text-to-text generation, with orthogonal error types.

3.4.4 Maximally merged error taxonomy

Building on the process of alignment and consensus identification in the previous section, the next step toward our overall goal was to create a single generic error annotation taxonomy that would work across task construals and application domains. Importantly, this is an error annotation taxonomy, not an automatic edit-distance metric. Unlike TER, which measures token-level edits against a reference, our consensus taxonomy aims to classify meaning/content deviations relative to the input, and is task-agnostic. More specifically, our objectives were as follows:

1. To normalise the different names used in the source papers for the same error type using single error category names;
2. To ensure that names and definitions are general enough to work for text generated under both data-to-text and text-to-text tasks, the latter including at least summarisation, paraphrasing and machine translation; and
3. To extract the orthogonal error type dimensions and incorporate them separately, rather than duplicating them across different parts of the taxonomy as previously in Figure 3.1, e.g. for the meaning deviation subtypes towards the

top right of the diagram (NEs, Pos/neg, Numerical, Other), since including them in the main hierarchy would necessarily duplicate categories.

The extraction criterion for orthogonal error type dimensions was that any of the primary error categories can additionally be annotated with them.We identified the following:

1. Type of deviation in meaning between input and output Sharou and Specia (2022); Thomson and Reiter (2020); Tang et al. (2022a) resulting from one of the primary error types (listed at the end of this subsection):
   (a) NE Deviation: Deviation in named entities.
   (b) Pos/Neg Deviation: Deviation in negation, polarity or positive/negative sentiment.
   (c) Numerical Deviation: Deviation in numerical content.
   (d) Other Meaning Deviation.
2. Number of words involved in a given error Caseli and Inácio (2020): Single Word and Multiple Words.
3. Severity of the error: Major and Minor Zou (2022); Popović (2020); Specia et al. (2017, 2021).
4. Degree to which words in the error contribute to the content/meaning of the output: Content Word(s) vs. Function Word(s) Costa et al. (2012, 2015); Specia et al. (2017).

Note that our aim was to extract all error categories that met the extraction criterion precisely because, if systematically applied, they cause unnecessary duplication in the hierarchy. Conversely, the remaining error categories do not cause such duplication. In other words, this is a fundamental difference between, on the one hand, the error categories in the taxonomy which are in natural subsumption relationships with each other, and, on the other, the orthogonal error types which are not, and can apply to any categories at any level of the hierarchy. We believe it is therefore right to account for them differently.

After taking out the orthogonal error types, the remaining error categories in the taxonomy are as shown on the left of Figure 3.2. The corresponding definitions are the following:

1. **Content/Meaning Error**: The highest level error category subsuming all errors in outputs that relate to the content/meaning of the output rather than its form (see also start of Section 3.4.4 on content vs. form).

2. **Omission**: Some content that is present in the input and should be rendered in the output is not present in the output. Moreover there is no content in the output that is intended to render it, but does so wrongly. That is, this type of error can be fixed by adding something to the output.

3. **Addition**: Some content that is not present in the input and should not be rendered in the output is present in the output. Moreover there is no content in the input that it is intended to render, but renders wrongly. I.e. this type of error can be fixed by removing something from the output. Addition errors can be further classified into:

   (a) Duplication: Some content is repeated verbatim in the output, but there is no corresponding repetition in the input.

   (b) Other.

4. **Substitution**: Some content in the output, that is intended to convey some content that is present in the input, does it wrongly. This definition means that a substitution cannot equally be construed as the combination of an omission and an addition. This type of error can be fixed by replacing something in the output. Substitution errors can be further classified into:

   (a) Should Not Be Verbatim: Some part of the input has been copied verbatim to the output, but should have been rendered differently.

   (b) Should Be Verbatim: Some part of the input should have been copied verbatim to the output, but has been rendered differently.

(c) Lexical Error: An error that can be fixed by replacing one lexical item in the output with another.

(d) Error In Input: An error that is caused by an error in the input.

(e) Reordering: An error that can be fixed by reordering parts of the output.

(f) Other Wrongly Rendered Output.

3.4.5 Example usage

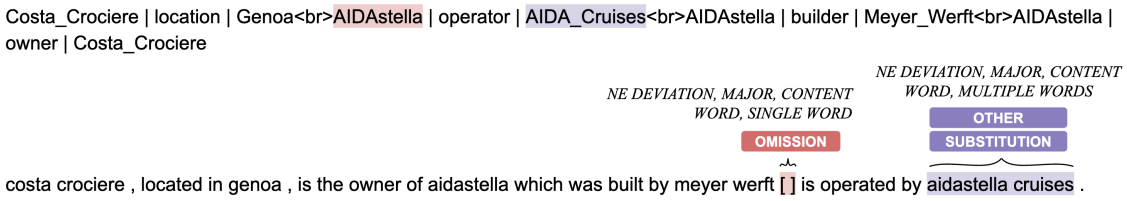

Figure 3.3: Input/output pair from WebNLG dataset: input 'triples' at the top, verbalisation beneath, both with linked annotations for two errors, using maximally merged consensus taxonomy.

Figure 3.3 shows an input/output pair from the WebNLG Shared Task data (Gardent et al., 2017a) annotated with the (maximally merged) consensus taxonomy, including annotations for the orthogonal error types. The input meaning representation (known as a set of triples in WebNLG terminology, where each triple consists of the elements: Subject, Predicate, and Object) is shown at the top, with a verbalisation for it produced by one of the participating systems. We later use this example method to annotate semantic errors on various data-to-text system in Chapter 4.

The steps in annotating the output text for errors are as follows (shown here for manual annotation by marking up and labelling character spans; alternatively labels can be attached to default spans, such as sentences or whole inputs/outputs):

1. Compare input and output, identifying and marking up word spans in the output text that contain some error, and the corresponding span in the input; in the case of Omission errors, the span in the output will be an empty string in the approximate place where the verbalisation of the omitted content would

be, had it been rendered, and in the case of Addition errors, conversely an empty string is annotated span in the input;

2. For each linked annotation, a label is attached from the top level in the taxonomy (Omission, Addition, Substitution), then from the second level, until leaf nodes are reached;

3. Finally, the orthogonal error type labels are attached, one from each type.

Note that this is intended as an illustration of how the consensus taxonomy would be used for manual annotation. The annotation as shown in Figure 3.3 was done by the thesis author. See following section re expanding the taxonomy with further error categories, and using it for automatic error annotation.

3.5 Discussion

Error analysis identifying different types of errors plays an important role in NLP system development, providing information about specific strengths and weaknesses frequencies of error occurrence, for different approaches, rather than a global quality assessment. For this, whether manually or automatically carried out, error categories need to be defined, at multiple levels of granularity.

The current situation is that many different sets of error categories are in use, certainly for different application tasks (MT, Paraphrasing, data-to-text, etc.), but very much also the same tasks, as can be seen from the ten different sets of error types for MT we have included in this chapter (see Table 3.1). Creating a consensus taxonomy incorporating and standardising existing taxonomies means both being able to create annotations and counts that are directly comparable across different research efforts, and, through maximising consensus, increase the taxonomy's acceptability.

The consensus taxonomy as presented incorporates only error categories as used in previous work. The taxonomy can be expanded in various ways at the leaf nodes to increase granularity, notably in the Substitution category, and particularly to

reflect domain and task-specific distinctions. In principle, the taxonomy can be used for both manual and automatic error annotation.

In standardising the error categories we have tried to make them applicable across all input-controlled forms of text generation. However, the judgment in particular of whether there is an Omission is different in tasks where not all of the input needs to be rendered in the output, such as Summarisation. The task we will use the taxonomy for is word-span error annotation across various data-to-text systems as reported in Chapter 4.

Unlike Chapter 2, which set the background and terminology, this chapter presented our work where we took 15 papers with error annotation schemes and derived a consensus taxonomy from them in two stages. The first was directly forming a taxonomy from error categories and hierarchical relations between them in their original forms; the second stage was maximally standardising and merging error categories and identifying and treating separately what we called orthogonal error categories that are not in any subsumption relations with other categories.

This chapter established a task-agnostic consensus taxonomy for content/meaning errors. Chapter 4 presents a systematic review of error annotation schemes and taxonomies for human and machine-generated text. From this review, we then select a subset of papers that is used in the systematic survey reported in this chapter and and apply the consensus error taxonomy in a span-based error annotation study on triple input(s) and data-to-text system outputs, in order to investigate the nature of semantic errors across systems.

Chapter 4

Exploring Semantic Consistency Issues in Natural Language Generation Systems

4.1 Background

With the recent advances in pretrained language models scaled to ever larger numbers of parameters and the ever larger datasets that these models are trained on, the quality of generated text has improved dramatically, especially in terms of fluency and coherence (Brown et al., 2020; Zhang et al., 2022; Hoffmann et al., 2022). However, these advances have come at the expense of semantic controllability, with the modern generative systems being commonly prone to errors of meaning/content such as *additions*, *omissions*, *repetitions* and *substitutions* (Bang et al., 2023). In controlled text generation, whether converting structured data into text or producing free-form outputs guided by specific attributes, the output must faithfully reflect some or all of the input. However, maintaining this semantic consistency remains a persistent challenge for even the most advanced neural generators (Corbelle et al., 2022; Raj et al., 2023; Yang et al., 2024). Addressing semantic consistency issues requires identifying whether the generated output omits, adds or alters meaning

relative to the input, and pinpointing specific semantic error types to accurately reflect the inconsistency. This can be achieved through semantic error annotation or labelling using the different error types provided by a chosen annotation scheme or taxonomy (e.g. our consensus semantic error taxonomy described in Chapter 3).

In this chapter, we conducted a systematic literature survey of 22 papers. The scope of this survey is to map out work on annotating errors in human and machine generated text, with a particular focus on error taxonomies. The purpose of this survey is to draw similarities and differences in the error annotation schemes reported in these papers for semantic errors. Additionally, we investigate how different semantic, or content-related, errors made by different types of data-to-text systems differ in terms of number and type.

The content of this chapter is previously reported in our following publications:

1. Huidrom and Belz (2022)

2. Huidrom et al. (2024)

4.2 Chapter Overview

We present a systematic review of semantic errors in different error annotation schemes and taxonomies for human and machine generated text. We looked at a total of 22 papers, highlighting similarities and differences among these error annotation schemes based on factors such as target text type, error types captured, NLP task involved, purpose of annotation and annotation scheme properties.

Additionally, we conducted a comprehensive text annotation experiment yielding word span annotations of semantic errors made by a range of different data-to-text systems. The result is a dataset of system outputs with manually annotated semantic errors, providing a basis for valuable insights regarding semantic errors made by different systems. We provide an in-depth analysis of the annotated data to identify patterns and correlations between different types of errors.

Our explorations have resulted in:

1. A structured survey of work on error type annotation schemes (with a focus on error taxonomies) in human and machine generated text, covering both content/meaning and form errors, as reported in papers from the ACL Anthology (Section 4.3).

2. In-depth analysis of properties of the error annotations schemes surveyed (Sections 4.3.2 and 4.3.3).

3. A comprehensive text annotation experiment by obtaining word-span error annotations of semantic errors made by a range of different data-to-text systems using brat annotation tool (Section 4.4).

4. Creation and publication of a dataset of system outputs with manually annotated semantic errors, providing a basis for valuable insights regarding semantic errors made by different systems (Section 4.4.2).

5. In-depth analysis of the annotated data to identify patterns and correlations between different types of errors (Section 4.4.4).

6. Detailed discussion of our key observations and findings for each respective study (Sections 4.3.3, 4.4.5).

4.3 Systematic Review of Error Annotation Schemes and Error Taxonomies

4.3.1 Paper selection and filtering

We searched the ACL Anthology[1] for our paper selection with the query terms "error taxonomy" and "NLP," and "error type annotation" and "NLP" which yielded 84 results. We included papers published up to year 2022. We removed non-paper

[1] `https://aclanthology.org`

results and duplicates,[2] which left 27 papers. We manually examined the remaining papers keeping only those that actually reported an error taxonomy or error annotation scheme including semantic errors, which left 18 papers. We added four relevant papers from the related work sections of three of the 18 papers. Our survey reviews the resulting 22 papers.

4.3.2 Properties of Error Annotation Schemes

Addressing semantic consistency issues requires identifying whether the output omits, adds or alters meaning relative to the input, and correctly categorising these inconsistencies into specific semantic error types. This process relies on semantic error annotation or labelling using the different error types provided by a chosen annotation scheme or taxonomy. It is essential to distinguish clearly between semantic error annotation schemes (or taxonomies) and semantic error assessment. Semantic error assessment is the process of verifying, manually or automatically, that everything conveyed by the output is fully aligned with, and does not exceed, what is present in the input. In contrast, semantic error annotation schemes (or taxonomies) define various types of semantic errors that can be used to annotate or label inconsistencies in an input–output pair.

In order to enable a systematic comparison of the various error annotation schemes and draw conclusions about their similarities and differences, we labelled each scheme in terms of (i) whether it was designed for machine or human generated text; (ii) whether it contained error types related to semantic accuracy, fluency or both; (iii) NLP system task; (iv) purpose of the annotation that was carried out; and (v) details of how many error labels and how many hierarchical levels there are in the scheme.

[2]Search results included 39 author profiles, and 18 paper duplicates, where papers are repeated in two places, e.g. when the same paper is found both individually and as a part of proceedings in the search.

Error Types	Definition(s)
Hallucination (h)	An example definition in the context of NLP is "generated content that is nonsensical or unfaithful to the source content." This is a widely accepted term (Ji et al., 2022) to refer to content in the output that does not have corresponding content in the input. The term comes from the field of psychology where e.g. Blom (2010) defines hallucination as "a percept, experienced by waking individual, in the absence of an appropriate stimulus from extracorporeal world." Some other terms used for errors very similar to hallucination are *addition, insertion, extra words, unnecessary information.*
Omission (o)	Used so commonly in MT that a definition is not usually given, this term refers to content in the input that should be rendered in the output not having corresponding content in the output (Weng et al., 2020). Some alternative terms in use are *deletion, absent word/n-gram, missing context/information.*
Replacement (re)	We use this term to refer to a range of error phenomena (given a variety of names in the literature) where some content in the output is clearly intended to convey some part of the input, but does so incorrectly (Subramaniam et al., 2009; Gouws et al., 2011; Han and Baldwin, 2011; Al Sharou and Specia, 2022). We can also look at replacement errors from the perspective of a combination of omission and addition, in the special case where what is added is the incorrect version of what is omitted. Some alternative terms also in use are *substitution, mistranslation, transposition.*
Repetition (r)	An example definition is "occurrence of the same words several times or syntactically similar units unintentionally or on purpose" (Al Sharou et al., 2021). Some alternative terms in use are *duplication, redundancy.*

Figure 4.1: Definitions of high-level semantic error types found in the literature (Col 1: semantic error types; Col 2: definitions). Please note that we generally prefer the term 'arbitrary content addition' or just 'content addition' but use the original term used in the literature ('hallucination') to avoid confusion.

4.3.2.1 Text type and error type:

We categorise each paper in terms of the (1) *text type*, and (2) *error type* addressed. Regarding the former, we group the error annotation schemes in our survey into those developed for machine-generated text (MGT) and those developed for human-generated text (HGT), according to the following definitions:

- **Machine generated texts (MGT)** i.e. synthetic texts generated by a system or a model based on pre-defined rules or algorithms, including MT, text summarisation, story generation etc. Errors like mistranslation, omission or arbitrary content addition, etc. are observed frequently in annotation schemes for this text type. Figure A.5 in the Appendix A.2 provides examples of typical errors. This is the text type of interest for this thesis.

- **Human generated texts (HGT)** include reference texts for evaluating systems, and training corpora for various downstream NLP tasks. The nature of the text depends on its intended purpose and oftentimes, they are used for evaluation of the models we build. Compared to MGT, HGTs are less prone to semantic errors. However, it cannot be generalised that all human generated texts are of good quality, and semantic errors do occur. Figure A.14 in the Appendix provides examples of errors in human-generated texts.

We further categorise error annotation schemes in terms of the broad error type(s) addressed. Here we use 'accuracy' and 'fluency[3]' as shorthand for content type errors as per Figure 4.1, and non-content type errors, respectively. These two terms are used frequently in the MT literature, e.g. in the Multidimensional Quality Metrics (MQM) framework[4] (Lommel et al., 2014) where they are defined as follows:

- **Accuracy**: Errors where the target sentence does not correspond to the source text due to omission, distortion or addition to the text. Error types include mistranslation, over-translation, under-translation, untranslated, omission, and addition.
- **Fluency**: Errors related to grammar and style. Examples include errors relating to spelling, punctuation, grammatical rules, inconsistent style, unidiomatic style etc.

4.3.2.2 Structure of annotation scheme:

We also categorised error annotation schemes in terms of two structural properties:

1. The **number of different error types** included in an annotation scheme, for which we use a standardised definition as the number of nodes in the tree including the root;

[3]Please note that for this thesis, we are only interested in content/meaning errors and not fluency error type.

[4]MQM is not listed separately in Section 3.3, which surveys papers that apply MQM rather than framework specifications.

Sl.no	Work	Type	NLP Task	Accuracy	Fluency	# Error types	Depth	Purpose
1.	Costa et al. (2015)	MGT	MT	✓	✓	36	4	EAn+EA+EC
2.	Al Sharou and Specia (2022)	MGT	MT	✓		8	1	EAn+E(S)
3.	Caseli and Inácio (2020)	MGT	MT	✓	✓	17	2	EAn+EA+E(S)
4.	Federico et al. (2014)	MGT	MT	✓	✓	5	1	EAn+EA
5.	He et al. (2021)	MGT	EA(D)	✓		32	2	EAn+EA+E(S)
6.	Belkebir and Habash (2021)	MGT	EA(S)		✓	34	2	EAn+EA
7.	Huang et al. (2020)	MGT	TS	✓	✓	11	2	EAn+E(S)
8.	Di et al. (2019)	MGT	MI		✓	5	2	EAn+EA+E(S)
9.	Mahmud et al. (2021a)	MGT	TS(SC)	✓		39	2	EAn+E(S)
10.	Costa et al. (2012)	MGT	CE	✓	✓	11	2	EAn+EA+E(C)
11.	Macklovitch (1991)	MGT	MT	✓	✓	23	2	EAn+E(S)
12.	Lin et al. (2022)	HGT	EC(T)		✓	7	2	EAn+E(C)+E(S)
13.	van der Goot et al. (2018)	HGT	TN		✓	20	3	EAn+E(S)
14.	Ng et al. (2014)	HGT	EC(G)		✓	29	1	EAn+E(S)
15.	Dickinson and Herring (2008)	HGT	MDC		✓	11	2	ED+EA
16.	Dickinson (2010)	HGT	EG		✓	14	3	EAn+E(C)
17.	Nagata et al. (2018)	HGT	EC(S)	✓	✓	14	1	EAn+E(C)+EC
18.	Gayo et al. (2016)	HGT	CE	✓	✓	4	1	EAn+E(C)
19.	Barbagli et al. (2016)	HGT	N(CL)	✓	✓	35	3	EAn+E(C)
20.	Himoro and Pareja-Lora (2020)	HGT	EC(S)		✓	39	7	EAn+E(C)+EC
21.	Caines et al. (2020)	HGT	D		✓	25	1	EAn+E(C)
22.	Korre et al. (2021)	HGT	ETC		✓	17	1	EAn+E(C)+EC

Table 4.1: Overview table of properties of the error annotations schemes surveyed (for explanation of abbreviations see Table 4.3 and in text).

2. The **depth of the hierarchical structure** underlying the scheme. If there is no underlying hierarchical structure, then depth=1. Depth = levels - 1, where levels are the number of nodes in the longest path from root to the leaf nodes.

4.3.2.3 NLP task and annotation scheme purpose:

We distinguish the following NLP System Tasks, abbreviated as indicated in square brackets in tables below: Machine Translation [MT], Text Summarisation [TS], Textual Summarisation of source code [TS(SC)], Type-level Universal Morphological Reinflection Task [MI], Automatic Translation Error Correction [EC(T)], Text Normalisation [TN], Grammar Error Correction [EC(G)], Morphological Error Detection and Classification [MDC], Error Generation [EG], None (Corpus Linguistics) [N(CL)], Dialogue [D], Error Type Classification [ETC] and Spelling Error Correction [EC(S)].

NLP System Task also includes the following automatic forms of error annotation: Automatic Error Annotation for Dataset Creation [EA(D)], Automatic Error Annotation of System outputs for evaluation [EA(S)], Automatic Corpus Error annotation/analysis [CE]. The NLP System Task is defined for what the error annotation scheme is used for in its respective papers.

Inspired by Machine Translation (MT) research which takes a very structured approach to error analysis (Stymne and Ahrenberg, 2012; Koponen, 2010), error classification (Vilar et al., 2006; Popović and Burchardt, 2011; Popović, 2021), and building error taxonomies (Costa et al., 2015; Al Sharou and Specia, 2022), we also categorise our error annotation schemes in terms of the **purpose for which an error annotation scheme was created** as follows: Error Classification is EC, Error Annotation is EAn, Evaluation for systems is E(S), Evaluation for corpus errors is E(C), Error Detection is ED and Error Analysis is EA. The purpose is defined for what the error annotation scheme that was created is used as in its respective papers.

4.3.2.4 Labelled annotation schemes:

Table 4.1[5] shows each of the 22 surveyed papers alongside their individual labels. Columns 3 and 4 indicate text type and NLP Task, Columns 5 and 6 whether Accuracy or Fluency is addressed, and the last three columns show number of different Error Types, Depth, and Purpose for which the scheme was created, respectively, all as defined in the preceding section.

As can be seen, there is an even distribution of papers over text type addressed (HGT vs. MGT). Moreover, none of the 11 papers addressing HGT address only accuracy errors, most address only fluency (8 out of 11), and just three address both accuracy and fluency errors. For the 11 MGT papers, we have a fair mix of different types of errors i.e., three address only accuracy errors, two only fluency

[5]Table 3.1 summarises structural properties for the 15 schemes used to derive our taxonomy, whereas Table 4.1 reports the full 22-paper survey with additional attributes (HGT/MGT, task, Accuracy/Fluency coverage, and purpose).

errors, and six address both. Determining the number of error types and the depth of the hierarchy (if any) has been a challenge due to lack of clarity within the papers. For example, Gayo et al. (2016) do not mention the error sub-types in the taxonomy clearly which makes counting them difficult. This means we have provided an estimate in some cases.

All 22 papers have a combination of purposes for which the scheme was created (last column).

4.3.3 Discussion

In this section, we present a structured survey of error taxonomies and error-annotation schemes for human and machine-generated text, covering both error of content/meaning and form. A subset of the works surveyed here is used in Chapter 3 to derive the consensus taxonomy for content/meaning errors.

4.3.3.1 Trends Observed:

Table 4.2 presents the overall trend in different types of semantic errors included in error annotation schemes over the years in our surveyed papers. We mark as 1 if we encounter any one of the semantic error types from Figure 4.1 in a paper (each paper can have more than one semantic error type). For example, we have a count of 4 for arbitrary content addition errors from 2020 which means four papers address such errors in the year 2020. The paper IDs are taken from Table 4.1.

Semantic Error	1991	2008	2010	2012	2014	2015	2016	2018	2019	2020	2021	2022	TOTAL
Arbitrary content addition (h)	1			1	1	1	1	1		4	4	1	15
Omission (o)	1			1	1	1		1		4	4	1	14
Replacement (re)	1				1					4	2	2	10
Repetition (r)						1	1			2	2	1	7
TOTAL	**3**			**2**	**3**	**3**	**2**	**2**		**14**	**12**	**5**	**46**

Table 4.2: Number of taxonomies that incorporated each of the four high-level semantic error types from Figure 4.1, shown per publication year in our set of 22 papers.

Purpose	(Paper)[Sem. Error]	MGT	HGT
Error Analysis & Error Annotation (EA+EAn)	(4,6)[*h,o,re*]	✓	
Error Detection & Error Analysis (ED+EA)	(15)[*n*]		✓
Evaluation of systems & Error Annotation (E(S)+EAn)	(2,7,9,11,13)[*h,o*],(2,11)[*re*], (7,9)[*r*],(14)[*n*]	✓	✓
Error Annotation & Error Analysis & Error Classification (EAn+EA+EC)	(1)[*h,o,r*]	✓	
Evaluation of corpus errors & Error Annotation (E(C)+EAn)	(19,21)[*o*],(21)[*h,re*], (16,18)[*n*]		✓
Error Annotation & Error Analysis & Evaluation of systems (EAn+EA+E(S))	(3,5)[*h,o,r*],(3)[*re*],(8)[*n*]	✓	
Error Annotation & Error Analysis & Evaluation of corpus errors (EAn+EA+E(C))	(10)[*h,o*]	✓	
Error Annotation & Evaluation of corpus errors & Evaluation of systems (EAn+E(C)+E(S))	(12)[*re*]		✓
Error Annotation & Evaluation of corpus errors & Error Classification (EAn+E(C)+EC)	(20,22)[*h,o,re*], (17)[*n*]		✓

Table 4.3: For each (combination of) purpose(s) in the 22 surveyed papers, the taxonomies to which it applies (round brackets), and the semantic error types covered by each of those taxonomies [square brackets]. We also show text type to which each (combination of) purpose(s) applies.

Four out of the five papers in our survey published more than ten years ago (2012 and earlier) are categorised as HGT (except the paper by Macklovitch (1991) which is categorised as MGT), and do not report any semantic errors in their error annotation scheme. In addition, another paper, by Di et al. (2019), grouped under MGT, also does not report any semantic errors. We observe a total of 46 semantic error types reported in the papers from our survey.

Table 4.3 shows in the first column, all the combinations of purposes for which an error annotation scheme was created that we encountered in our 22 surveyed papers. The second column shows paper number (e.g. "(15)"), type of semantic errors addressed in each paper (e.g. "[*h, o, r*]") or none ("[*n*]"). The last two columns show text type (HGT vs. MGT).

We observe that papers where text type is MGT typically address one or more semantic errors (10 out of 11 papers), except for the paper by Di et al. (2019) whose purpose is error analysis and evaluation of systems. Half of the papers labelled HGT do not address any semantic errors. The other half of the papers with error anno-

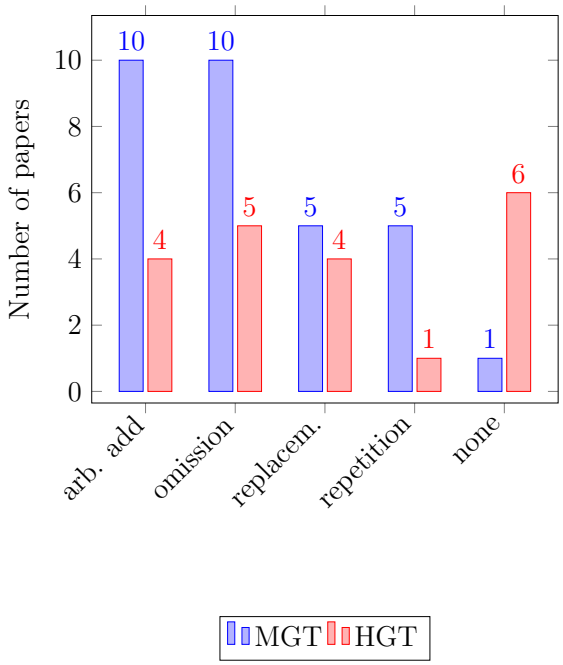

Figure 4.2: Number of papers reporting semantic errors in the taxonomies.

tation and evaluation of corpus or error classification as purpose in HGT addresses semantic errors. The statistics of how many papers address each of the high-level semantic error types in the MGT and HGT groups can easily be seen in Figure 4.2.

Table 4.2 shows the high-level semantic error types reported in each year in our survey. It is interesting to observe that addressing semantic errors has become increasingly frequent in very recent years. One reason is likely to be the shift from controlled pipeline approaches to end-to-end neural approaches for many NLP tasks.

4.3.3.2 Observations:

We discuss the issues we observed in labelling the error annotation schemes and taxonomies in our survey. We summarise these observations from the perspective of semantic errors as follows:

1. **Lack of standardisation across schemes** (e.g., definition, examples) is observed which hampers deriving a standardised framework for semantic errors. We found that 11 out of 22 papers (50%) mention only the name of an error type or its sub-type without defining them at all. It is highly observed in the HGT group, with eight out of 22 papers. In some cases it is difficult to

categorise schemes/taxonomies in terms of the *high-level error type(s)* from Figure 4.1.

2. **Differing and/or incompatible error names and definitions:** In our survey, we encountered only two papers Dickinson and Herring (2008); Dickinson (2010) with mutually compatible error type definitions and these are by the same first author. For the remaining papers, either the error definitions means the same but the error term is different, or vice versa.

3. **Borderline error types** that cannot clearly be assigned either to semantic accuracy or to fluency. Categorising the error annotation schemes for which this is the case in the survey as accuracy and/or fluency errors is sometimes difficult due to the (lack of) provided definitions, examples, etc. We found 12 out of 22 papers (which is more than 50%) to be difficult to categorise which corresponds to three out of 11 papers for the MGT group, and 9 out of 11 papers for HGT. This difficulty implies an unclear boundary between accuracy and fluency types of errors.

Further interesting observations can be drawn from Table 4.1 and Table 4.3 concerning the relationship between semantic errors on the one hand, and text types (MGT, HGT) and broad errors types (accuracy and fluency), on the other. Nine out of 11 papers in the MGT group address either accuracy error types only (3/9), or accuracy error types together with fluency error types (6/9). Arbitrary content addition (currently more commonly known as hallucination) and omission are the most common semantic errors reported and we see them in all nine papers under the MGT group. Repetition (in combination with other semantic errors) is the next frequently reported semantic error and we see it in five out of nine papers under the MGT group. Meanwhile, three papers in the HGT group address both accuracy and fluency error types with only one paper out of the three addressing omission.

4.3.3.3 Conclusion

In conclusion, we conducted a structured survey of work on error type annotation schemes (with a focus on error taxonomies) in human and machine generated text, covering both content/meaning and form errors, as reported in papers from the ACL Anthology. We conducted our study from the perspective of semantic error annotation. We observed a number of issues while analysing the papers in our survey which we characterised in terms of (1) lack of standardisation, (2) differing/incompatible error names and definitions across different papers, and (3) borderline error types which resist being classified as either fluency or accuracy related. Notably, we observed that these borderline cases occur more frequently in error annotation for human-generated text rather than machine-generated. Given the findings, our thesis adopts a consensus error taxonomy for semantic errors as described in Chapter 3. We use our consensus error taxonomy on its highest level in our experiment reported in Section 4.4.

4.4 Investigating Error Types in Existing Language Generation Systems

We conducted a comprehensive text annotation experiment yielding word span annotations of semantic errors made by a range of different data-to-text systems. The result is a dataset of system outputs with manually annotated semantic errors, providing a basis for valuable insights regarding semantic errors made by different systems. We provide an in-depth analysis of the annotated data to identify patterns and correlations between different types of errors. The resulting insights into how NLG system type, input length and seen vs. unseen inputs relate to specific semantic error types are crucial. By looking at if there are omitted, added or altered meaning in the generated output relative to the input, we can precisely identify each system's strengths and weaknesses.

We start by obtaining word-span error annotations of semantic errors in a vari-

ety of data-to-text system input/output pairs, using the brat annotation tool. We discuss our method in more detail in the next section. We then analyse the annotations to determine how many semantic errors different (types of) systems make, and what specific types of errors they make, and thus to get an overall understanding of semantic strengths and weaknesses among various types of NLG systems.

Appendix A.3 include the participant recruitment email text, feedback from pilot participants, annotation steps, and additional results tables and analyses.

4.4.1 Pilot experiment and feedback

We conducted a pilot experiment on a set of 10 triples/verbalisation pairs with 10 researchers from ADAPT Research Centre, Ireland. The pilot preceded the main annotation study and was used to test the guidelines and the brat setup. We collected feedback via a Google Form to identify questions or issues encountered during the annotation process, and to collect suggestions regarding ways to improve the evaluation design, etc. We paid each evaluator 15 Euros per hour for the pilot. We received ethical approval from DCU's Research Ethics Committee (REC) to perform this experiment (REC Reference: DCUREC/2023/106).

One common suggestion was to add more examples to the annotation guidelines, including special cases that annotators should look out for. Other feedback related to how to present the layout of triples/verbalisation pairs on brat, providing step-by-step instructions on using brat, and giving background information on what a triple and verbalisation are. More details can be found in Appendix A.3.2.

After improving the evaluation design based on the feedback from our pilot experiment, we conducted our main evaluation study with 15 evaluators on 30 triples/verbalisation pairs for each evaluator. We paid 25 Euros for our main study, estimating that it took about an hour to do. We raised the payment relative to the pilot experiment due to the task's increased complexity in the number of triples/verbalisation pairs to be evaluated. All communication for both the pilot and main experiments took place via email exchanges.

4.4.2 Word-span-based annotation method

4.4.2.1 Types of systems

We evaluate a total of 15 data-to-text systems, comprising three rule-based systems and 12 neural systems, of which two are off-the-shelf large language models (LLMs) without any training or fine-tuning. 13 systems are from the WebNLG 2020 Shared Task (Ferreira et al., 2020) and the other two systems are from Lorandi and Belz (2024b). The 13 systems from WebNLG were those that performed best in the shared task according to the multiple criteria used in WebNLG's human evaluation.

Table 4.4 provides an overview of the 13 systems in terms of their WebNLG categorisation (first column), the name of the participating WebNLG'20 team (where applicable), and the name of the model used by the submitted systems as per the WebNLG'20 system description reports. We colour-code system names by broad system type in orange (rule or template-based), blue (neural non-LLM) and pink (LLM), using inclusive color palettes,[6] following this scheme throughout the section. We release our annotated dataset publicly on GitHub.

4.4.2.2 Data selection and allocation

We randomly selected 450 input-output pairs by stratified sampling based on the number of triples in the input and WebNLG category.[7] We allocate these samples to each evaluator using repeated Latin squares[8] which has the effect that each evaluator annotated a different set of 30 input and system output pairs, and each evaluator assessed two system outputs from each system, given that we used two Latin squares where the size of each Latin square is the number of evaluators by the number of systems (15 x 15). The data selection and allocation process is illustrated in Figure 4.3. All our data are in English.

[6] https://www.nceas.ucsb.edu/

[7] We had intended to stratify by whether properties were seen or unseen in the training data, however, we used the WebNLG'17 list of unseen properties erroneously, so counts aren't in quite the same proportions as the whole dataset.

[8] We chose Latin-square design to optimise cost and benefit. The Latin-square design lets each evaluator review a small, balanced subset of input-output pairs, without having to annotate every possible input-system combination.

Categorisation	Participating Team (model type)
Monolingual, mono-task, template-based	[1]RALI (Template-based), [2]DANGNT-SGU (Template-based)
Baseline	[3]Baseline-Forge2020 (Rule-based)
Monolingual, mono-task, neural	[4]TGen (T5), [5]NILC (BART), [6]NUIG-DSI (T5)
Mono-task, bilingual approaches	[7]cuni-ufal (mBART), [8]Huawei Noah's Ark Lab (multilingual transformer-based seq2seq model), [9]OSU Neural NLG (T5), [10]FBConvAI (BART)
Bidirectional, monolingual approaches	[11]Amazon AI (T5), [12]CycleGT (T5)
Bidirectional, bilingual approaches	[13]bt5 (T5)
Large language models, no training or fine-tuning	[14]GPT 3.5, [15]Llama-chat-270B

Table 4.4: Color-coded (rule-based, non-LLM neural and large language models (LLM)) summary of the participating teams' systems categorisation, taken verbatim from WebNLG 2020 results report.

4.4.2.3 Participant selection

We invited researchers at the ADAPT Research Centre (Ireland) to participate in our study via an email (see email template in Appendix A.3.1) to the centre-wide mailing list, linking to a sign-up form that asked for English language proficiency (Proficient User – C1, C2, Independent User – B1, B2, Basic User – A1, A2), prior experience with error annotation (yes/no), and an example annotation. Participants were excluded if they had no prior experience with error annotation or if the example was incorrectly annotated.

We received a total of 11 sign-ups. Out of these, 10 marked their English language proficiency as Proficient User (C1, C2), and one marked it as Independent User (B1, B2). Six participants had prior experience with error annotation, while five did not. We selected six participants from the sign-ups based on their prior

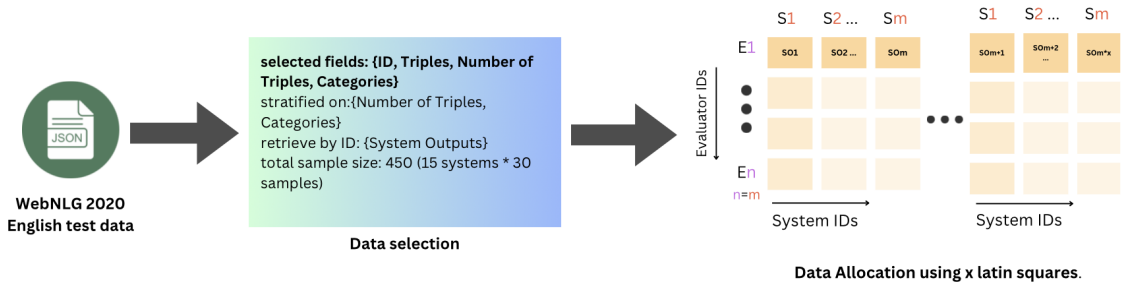

Figure 4.3: Data selection and allocation workflow.

experience with error annotation and the correctness of their example annotation.[9] An additional nine participants were selected from a previously conducted pilot experiment (see Section 4.4.1 for pilot details); these are proficient users of English and NLP researchers.

4.4.2.4 Error categories

We use three error types[10] and definitions for annotation, following Chapter 3. We refrain from using the term 'hallucination' due to its meaning in the field of psychology. For instance, Blom (2010) defines hallucination as "a percept, experienced by waking individual, in the absence of an appropriate stimulus from extracorporeal world." Instead, we use the term "addition" as defined below. In the following definitions, 'input' is the set of triples, and 'output' is the verbalisation (text).

- **Omission:** Some content that is present in the input and should be rendered in the output is not present in the output. Moreover, there are no word span(s) in the output that are intended to render it, but do so wrongly. I.e. this type of error can be fixed by adding something to the output.
- **Addition:** The output text contains word span(s) for which there is no corresponding part of the input that they render. In other words, some content that is not present in the input and should not be rendered in the output is

[9] Refer to the example annotation in the participation selection form; the link to this resource is provided in Appendix A.3.5.

[10] We did not include the Substitution error type because, in pre-pilot trials, annotating it reliably in a span-based setting proved cognitively demanding.

nevertheless rendered by some word span(s) in the output. Moreover, there is no content in the input that the word span(s) are intended to render, but render wrongly. I.e. this type of error can be fixed by removing something from the output.

- **Repetition:** Some content is repeated verbatim in the output, but there is no corresponding repetition in the input.

4.4.2.5 Annotation process

We record the word-span annotations of our set of input and system outputs pairs via the brat annotation tool.[11] The input here is a set of triples, and the system output is the generated verbalisation. Each triple consists of the elements Subject, Predicate, and ObjectFor example, an input triple could be `Take_It_Off (Subject), producer (Predicate), Wharton_Tier (Object)`, and the corresponding system output (verbalisation) could be *Wharton Tiers produced Take It Off.*

The annotation task is to mark and label omissions in the set of input triples, and additions and repetitions in the verbalisation. There can be any number of semantic errors, including none, in any triple-set/verbalisation pair.

4.4.2.6 Summarised annotation steps

The following is the summarised annotation steps. Verbatim annotation instructions can be found in Appendix A.3.3.

1. *Omission annotation:* The evaluator should check if each element in the input triples is verbalised. If any element is missing, it should be marked as an omission error. If the entire triple is not verbalised, each element of the triple should be marked as an omission. If all elements are verbalised, it means there are no omission errors.
2. *Addition annotation:* The evaluator should check if all content words and phrases in the verbalisation correspond to elements in the triples. If any con-

[11] `https://brat.nlplab.org`

tent word or phrase does not match an element in the triples, it should be marked as an addition error. If all content phrases correspond correctly, it means there are no addition errors.

3. *Repetition annotation:* The evaluator should check for repeated content in the output, including closing paraphrases. If any element in the triples is rendered more than once, it should be marked as a repetition error, unless there is corresponding repetition in the input triple elements. If all content words and phrases in the verbalisation correspond correctly to the triples without repetition, it means there are no repetition errors.

4.4.3 Human evaluation

4.4.3.1 Data

We use the system outputs from the WebNLG 2020 (Ferreira et al., 2020) on the English test dataset, which contains 1,779 different input triple sets. There are a total of 19 categories[12] in the WebNLG 2020 dataset, of which 16 are present in the training set, and three are unseen in the training set (Film, MusicWork, Scientist). Table 4.5 shows the overall counts of the number of triples and categories in the WebNLG 2020 English test dataset along with the counts in the stratified samples.

	Number of Triples							**Categories**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**Seen**	**Unseen**
O	369	349	350	305	213	114	79	966	813
S	90	90	90	75	60	30	15	285	165

Table 4.5: Triple size and category counts for the overall dataset (third row; 'O') and the stratified sample (fourth row; ; 'S').

4.4.3.2 Brat annotation tool setup

We use the brat annotation tool (Stenetorp et al., 2012), a web-based tool for text annotation, to record word-span annotations of semantic errors (omission, addition,

[12] A WebNLG dataset category groups RDF triples whose subject entities share the same DBpedia class (e.g., "City," "Astronaut," "Building"), enabling the evaluation of data-to-text systems across diverse semantic domains.

and repetition) in input triple sets and system output pairs. We use ngrok[13] to host brat for our experiment. The annotators were provided with the link to the brat annotation tool via email along with login credentials (username and password).

To annotate the errors, the evaluators have to (i) log in to the brat annotation tool using the provided credentials, (ii) open the correct data collection via the pop-up window and select the file containing the input–output pair to be annotated, (iii) select the word span to be marked as an error, which gives a pop-up window containing the list of semantic error types under the 'entity type' label in the interface, (iv) select the correct 'entity type' label for the selected word span (v) navigate to the next or previous file using the arrow buttons in the top-left corner, and (vi) log out of the brat annotation tool.

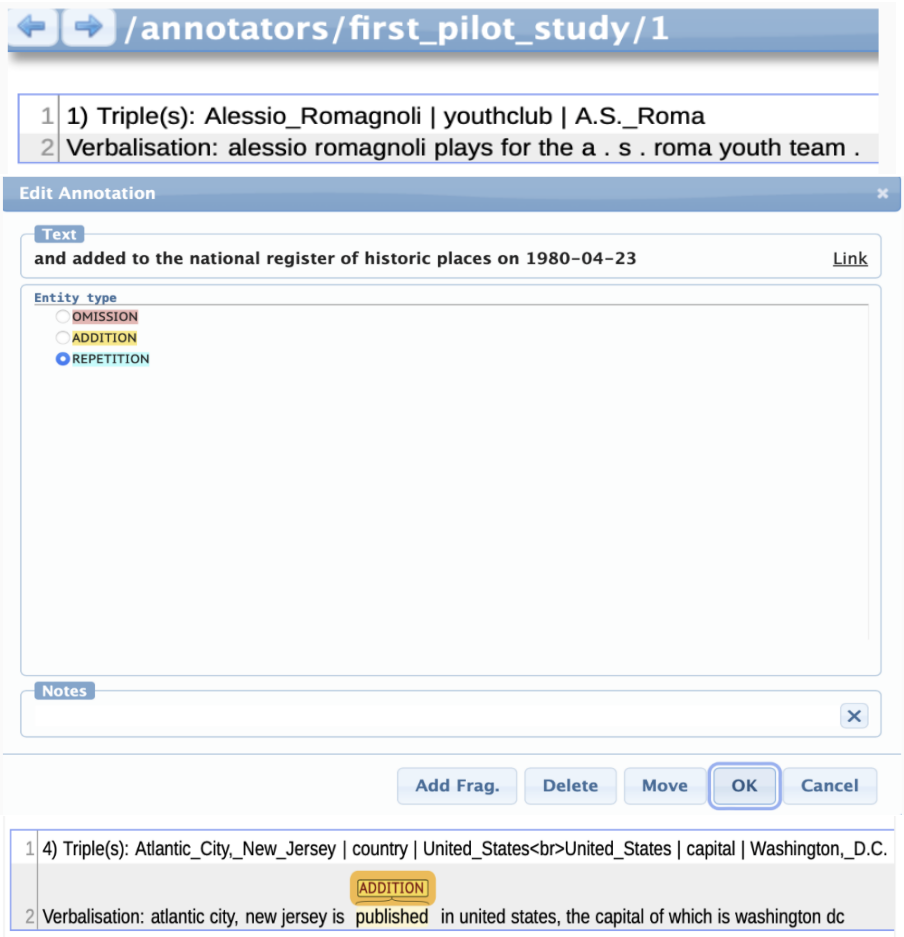

Figure 4.4: Screenshot of the brat annotation interface showing example input–output pairs and an Addition error annotation.

[13]A secure tunnelling service for exposing local web servers to the public internet. Available at `https://ngrok.com`

4.4.4 Results and analysis

In this Section, we present our results and analysis. We report the raw error counts (Table 4.6), followed by error rates for different input properties (Tables 4.7, 4.8, and 4.9). Lastly, we present further analysis on the correlation between error types and system type.

	System	#Omissions (n, %)	#Additions (n, %)	#Repetitions (n, %)	#Total errors	WebNLG 2020 (Avg. Raw)	
						Fluency	Data Coverage
Rule-based	Baseline-FORGE2020	12 (44.44%)	13 (48.15%)	2 (7.41%)	27	82.430	92.892
	DANGNT-SGU	14 (42.42%)	18 (54.55%)	1 (3.03%)	33	78.594	95.315
	RALI	13 (36.11%)	21 (58.33%)	2 (5.56%)	36	77.759	95.204
Non-LLM neural	Amazon-AI-Shanghai	15 (44.12%)	19 (55.88%)	0 (0.00%)	34	90.286	94.393
	NUIG-DSI	20 (58.82%)	14 (41.18%)	0 (0.00%)	34	88.898	92.063
	NILC	47 (54.65%)	36 (41.86%)	3 (3.49%)	86	74.851	81.605
	TGEN	18 (50.00%)	18 (50.00%)	0 (0.00%)	36	86.163	88.176
	CycleGT	19 (55.88%)	14 (41.18%)	1 (2.94%)	34	84.820	91.231
	FBConvAI	16 (38.10%)	23 (54.76%)	3 (7.14%)	42	90.837	93.169
	OSU-Neural-NLG	11 (44.00%)	8 (32.00%)	6 (24.00%)	25	90.066	95.123
	cuni-ufal	21 (52.50%)	15 (37.50%)	4 (10.00%)	40	87.642	93.291
	bt5	16 (45.71%)	19 (54.29%)	0 (0.00%)	35	88.688	93.836
	Huawei-Noah's-Ark-Lab	30 (46.88%)	30 (46.88%)	4 (6.25%)	64	75.205	84.743
LLM	GPT-3.5	13 (33.33%)	26 (66.67%)	0 (0.00%)	39	-	-
	LLAMA-2 70bchat	24 (42.11%)	32 (56.14%)	1 (1.75%)	57	-	-
	Mean	*19.267*	*20.4*	*1.8*	*41.467*		
	Standard Deviation	*9.177*	*7.763*	*1.859*	*15.95*		

Table 4.6: Counts and row percentages of each error type for each system. The last two columns present the average fluency and data coverage scores from the WebNLG'20 human evaluation analysis.

4.4.4.1 Raw error counts

Table 4.6 provides counts of each error type for each system, including the number of omissions, additions, repetitions, and total errors. The last two columns in Table 4.6 present fluency and data coverage scores copied verbatim from the WebNLG 2020 Shared Task human evaluation, obtained on the WebNLG 2020 Human Evaluation test set. Fluency measures "Is it possible to say that the text progresses naturally, forms a coherent whole and it is easy to understand the text?" Data coverage measures "Does the text include descriptions of all predicates presented in the data?" Human evaluators scored these criteria on a likert scale of 1-100.

We can see that omission and addition errors are more prevalent and consistent across systems, as indicated by their higher mean values and moderate standard

deviations. These errors occur relatively frequently, with less variation between systems, suggesting that their occurrence is more predictable. In contrast, repetition errors occur less frequently but have pronounced relative variability, as evidenced by a standard deviation that exceeds their mean. However, it has to be noted that due to their sparsity, repetition error counts and rates provide a less reliable picture than the other two error types investigated here. Omission and addition errors constitute 46.47% and 49.19% of all errors, respectively, while repetition errors account for just 4.34%.

Highlighting some system-specific observations, we can observe that[14] (i) NUIG-DSI, a non-LLM neural system, has a higher proportion of omission errors compared to the other two error types (58.82%); (ii) GPT-3.5 (LLM) shows a higher proportion of addition errors (66.67%) and has no repetition errors; (iii) OSU-Neural-NLG, a non-LLM neural system has a relatively high proportion of repetition errors (24%); and (iv) cuni-ufal, another non-LLM neural system, also has a high proportion of repetition errors (10%).

Rule-based systems have fewer total errors on average than neural systems. However, rule-based systems have a higher tendency towards addition errors, suggesting they struggle with filtering out unnecessary items. Non-LLM neural systems, show a balanced distribution between omission and addition errors. Repetition errors are relatively low across all non-LLM neural systems, except for OSU-Neural-NLG, which has a higher repetition rate (24%). LLM-based systems are observed to have a strong tendency to add extra content but manage to avoid repetitions effectively.

Systems with higher fluency scores tend to have lower total errors, especially omission and repetition errors. For example, Amazon-AI-Shanghai and OSU-Neural-NLG have fluency scores above 90 and these systems show below-average omissions (15 and 11, respectively) and repetitions (0 and 6) with total errors of 34 and 25. FBConvAI, despite a fluency of 90.837, has an average total error count (42) and a relatively higher number of additions (23) and repetitions (3).

[14]Each percentage is calculated relative to that system's total number of errors.

Systems with high data coverage tend to have higher addition errors. For example, DANGNT-SGU, Amazon-AI-Shanghai and OSU-Neural-NLG have data coverage score above 94 (see Table 4.6) and these systems exhibit low omission errors but sometimes have more additions as in DANGNT-SGU. Meanwhile, low fluency and low data coverage systems have higher errors across all types, in general. For example, NILC have the lowest fluency (74.851) and data coverage (81.605) score and highest total errors (86), suggests that low fluency and low data coverage correlates with higher errors, especially omission and addition errors. While specific fluency and data coverage scores are not available for the LLM systems, the error patterns suggest a tendency for over-generation (more addition errors).

Overall, the rule-based systems are more consistent and generally reliable with balanced error types, meaning that the rule-based systems tend to have a more uniform error distribution, with less variation in the number of omission, addition, and repetition errors between the different rule-based systems. Non-LLM neural systems can achieve higher fluency and data coverage but need careful management of errors, meaning that high fluency and high data coverage correlates with lower errors. LLM-based systems show potential but require improvement in addressing over-generation (additions) and missing content (omissions) issues effectively.

4.4.4.2 Error rates relative to different factors

In this section, we calculate error rates relative to (i) **input size** (number of triples); (ii) **system type** (rule/template-based, non-LLM neural, LLM-based); and (iii) **seen vs. unseen properties**, in order to gain a better understanding of how these factors relate to errors.

(i) Rates of omission, addition and repetition errors relative to input size. Table 4.7 shows occurrence rates for omission, addition and repetition errors relative to different numbers of input triples (1–7). We define these error rates as:

$$\text{Error Rate}_{input\ size} = \frac{E_{i,e}}{i \times T_i} \tag{4.1}$$

where e denotes the error type (one of omission, addition, and repetition), and i denotes input triple size (one of 1–7). $E_{i,e}$ is the number of errors found for the given error type e and input size i, while T_i is the total number of data items of length i. Multiplying T_i by i gives us the total number of triples in data items of input size i. Intuitively, these error rates thus capture how many e.g. omission errors there are per triple for a given input size. Note that we need to look at per-triple rates here to be able to compare error rates across input sizes. For consistency, we also report the other two error rates below per triple.

	Error Rate						
Error Type	**1 triple** (n=90)	**2 triples** (n=90)	**3 triples** (n=90)	**4 triples** (n=75)	**5 triples** (n=60)	**6 triples** (n=30)	**7 triples** (n=15)
Omissions	0.167	0.183	0.152	0.23	0.187	0.3	0.2
Additions	0.278	0.139	0.207	0.23	0.217	0.256	0.191
Repetitions	0	0.011	0.015	0.013	0.013	0.055	0.029

Table 4.7: Rates of omission, addition and repetition errors relative to input size.

None of the error types follow a uniformly increasing or decreasing trend according to Table 4.7. Omissions and repetitions have a slightly clearer tendency to increase with more triples, indicating greater challenges in handling larger input sizes. Notably, 4-triple inputs show the same error rate (0.23) in both omissions and additions. The error rate at 6 triples is observed to be the highest error rate in omissions and at 1 triple as the highest rate for addition. Repetitions follow a clearer upward trend with increasing numbers of triples, although they are the least frequent error type.

It is clear from Table 4.7 that the complexity introduced by higher numbers of triples impacts error rates to some extent, and this is clearer in the case of omissions and repetitions. Additions do not show any clear trend with changing input sizes.

(ii) Rates of omission, addition and repetition errors relative to system type. Second, we look at occurrence rates for omission, addition and repetition

errors for rule-based, non-LLM neural and LLM system types. We define this error rate as follows:

$$\text{Error Rate}_{system\ type} = \frac{E_{s,e}}{i_s \times T_s} \tag{4.2}$$

where e denotes the error type (one of omission, addition and repetition), and s denotes system type (one of rule-based, non-LLM neural and LLM). $E_{s,e}$ represents the number of errors found for the given error type e and system type s. T_s is the total number of data items produced by systems of type s, and i_s is the average number of input triples in data items of type s.

	Error Rate			
		Neural		
Error Type	**Rule-based**	**LLM + Non-LLM neural**	**LLM**	**Non-LLM neural**
Omissions	0.137	0.219	0.195	0.224
Additions	0.182	0.223	0.305	0.206
Repetitions	0.018	0.019	0.005	0.022

Table 4.8: Rates of omission, addition and repetition errors relative to system type.

Table 4.8 highlights substantial differences in error rates between the different types of system. Rule-based systems have the lowest omission rate (0.137), with non-LLM neural systems having the highest (0.224), and LLM systems (0.195) falling in between. The indication is that overall, neural architectures are more prone to omission errors than rule-based systems, although LLMs less so than other neural systems.

Overall, addition rates are higher than omission rates, except for non-LLM neural systems. The gap is particularly big for LLM systems which also have the highest overall addition rate (0.305); rule-based systems have the lowest (0.182).

Repetition rates are notably low across all systems because the inputs are structured rather than raw text, and the models are optimised to generate natural, fluent output. LLM systems have the lowest repetition rate (0.005), suggesting a superior ability to avoid redundancies. Non-LLM neural systems have the highest repetition

rates (0.022), followed closely by rule-based systems (0.018).

Rule-based systems generally show lower error rates in both omissions and additions compared to neural systems, suggesting a more controlled and predictable output. Non-LLM neural systems have lower addition error rates (0.206), but these are still higher than those of rule-based systems. LLM models, while showing high error rates in additions, perform well in minimising repetition errors.

(iii) Rates of omission, addition and repetition errors relative to seen vs. unseen category. Finally, we look at occurrence rates for omission, addition and repetition errors relative to seen vs. unseen WebNLG properties. We define this error rate as:

$$\text{Error Rate}_{seen/unseen} = \frac{E_{c,e}}{i_c \times T_c} \tag{4.3}$$

where e denotes error type (one of omission, addition and repetition), and c denotes category (one of seen and unseen). $E_{c,e}$ is the number of errors found for the given error type e and category c. T_c is the total number of data items in category c produced by systems, and i_c is the average number of input triples in data items of type c.

Error Type	**Error Rate**	
	Seen (size 1-6 only)	**Unseen**
Omissions	0.144	0.301
Additions	0.199	0.246
Repetitions	0.011	0.03

Table 4.9: Rates of omission, addition and repetition errors relative to seen vs. WebNLG unseen category.

Table 4.9 shows the resulting error rates. Note that error rates are computed on the subset of data items of input lengths 1–6, because that is all we have for the seen category. We observe that the omission rate for data items containing unseen properties (0.301) is more than twice that of data items with only seen properties (0.144). This suggests that when the systems encounter data it has previously been

exposed to, it is much better at ensuring that necessary elements are not omitted. For addition rates, the difference between items with seen (0.199) and unseen (0.246) properties is smaller, Repetition errors are the least frequent across both categories, with 0.011 for seen and 0.03 for unseen, but here nearly three times as many mistakes are made for unseen properties.

4.4.4.3 Correlation between error types by system type

In Table 4.10, we report Pearson's correlation coefficients between error types for all system types combined (last row), and separately by system type (rest of table). We aim to look at correlations between error types at the system level and identify any observable trends.

	Om vs Add	**Add vs Rep**	**Rep vs Om**
Rule-based	0.619	-0.143	-0.866
LLM	N/A	N/A	N/A
Non-LLM neural	0.847	0.046	0.152
All neural	0.712	-0.076	0.197
Overall correlation	*0.715*	*-0.068*	*0.192*

Table 4.10: Pearson's correlation coefficients for pairs of error types, separately for the three system types at the top (NA for LLMs where we only have two data points), and for all system types in the last row (*Overall correlation coefficient*). Om=Omission, Add=Addition, Rep=Repetition.

We observe a strongly positive overall correlation between omissions and additions (0.715), i.e. systems that make more omission errors also tend to make more addition errors. This holds true for all system types. In contrast, there is no correlation between additions and repetitions (-0.068), or between repetitions and omissions (0.192), when not differentiating between systems (overall correlation).

However, when looking at correlations for system types separately, we observe a strong inverse relationship between repetitions and omissions (-0.866) for rule-based systems, i.e. systems that make repetition errors does not tend to make omission errors. Both non-LLM neural and all neural systems show strong positive correlations between omissions and additions (0.847 for the former, and 0.712 for the latter), i.e. systems that make more omission errors also tend to make more addition er-

rors. Non-LLM neural systems have the highest correlation between omissions and additions.

4.4.5 Discussion

Correlation and Dependency Insights. We observe a strong positive correlation between omissions and additions across different types of systems, perhaps indicating a common underlying cause for these errors where they do occur. Notably, neural systems (both LLM and non-LLM) exhibit this trend, perhaps suggesting that when these systems fail to include expected elements, they overcompensate by adding unexpected ones.

Distinct System Type Behaviours. Rule-based systems show a strong negative correlation between repetitions and omissions, and have a higher tendency towards addition errors than the other two, possibly because they struggle with precision in filtering out unnecessary items despite their reliable factual accuracy (Gatt and Krahmer, 2018).

On the other hand, the neural systems all have strong positive correlations between omissions and additions. Non-LLM neural systems show the highest such correlation, emphasising the need for robust training and error-mitigation strategies for such systems.

Impact of Seen vs. Unseen Data. All error rates are higher in data containing unseen properties than in data containing only seen. We observe a clear trend where systems perform better on familiar (seen) data across all error types. This is consistent with the expectation that models or systems are generally more accurate when dealing with data they have previously encountered. The considerably higher error rates for the unseen category indicate that systems' cannot transfer all learning to unseen data. This is particularly evident in the substantial increase in omission and repetition errors, suggesting that the underlying model may require further training or fine-tuning to improve its generalisation capabilities. In contrast, addition errors

show a smaller increase.

Errors by Input Complexity (Number of Triples). We observe that omissions and repetition rates have an overall tendency to increase with more triples, indicating that handling larger input sizes presents greater challenges. The fluctuation in addition rates without a clear trend suggests that this error type might be affected by specific characteristics of the input data rather than its size alone.

4.4.5.1 Conclusion

In conclusion, we conducted a manual word-span annotation experiment with the aim of investigating the different types and numbers of semantic errors observed in the texts generated by 15 data-to-text generation systems, namely 13 WebNLG 2020 systems and two more recent LLM-based systems. We outlined our annotation guidelines and experimental setup, then examined both absolute error counts and error rates by input size, system type, and seen versus unseen categories. Symbolic (rule/template) systems showed the highest semantic consistency, non-LLM neural models delivered better fluency and coverage but still generated notable errors and, LLMs struggled most with over-generation and omissions, exhibiting an addition error rate relative of system type of 0.305.

In addition, there was partial agreement between the reference labels from the slot-error script and NLI-SED system in 12 cases, where both detected an omission (and one additionally detected a hallucination). There was no partial agreement on hallucinations.

4.5 Chapter Summary

In this chapter, we systematically investigated semantic consistency issues in language generation, focusing on semantic errors such as omissions, additions, and repetitions. Our work has resulted in two major contributions.

First, we conducted a structured survey of 22 papers on error annotation schemes

and taxonomies for both human and machine-generated text. We analysed the surveyed schemes and taxonomies in terms of their target text type, error types, NLP tasks, annotation purposes, and structural properties. Our findings revealed a lack of standardisation across schemes, inconsistencies in error definitions and terminologies, and the presence of borderline cases between semantic accuracy and fluency errors. Please note that Chapter 3 uses a subset of papers to derive a consensus taxonomy of content/meaning error type, whereas Chapter 4 reports a broader 22-paper survey (human and machine-generated text) that also covers errors of form.

Second, we presented a comprehensive word-span-based annotation experiment targeting semantic errors in outputs from 15 data-to-text systems, including rule-based, non-LLM neural, and LLM-based models. Our analysis highlighted trends in error distribution relative to system type, input size, and seen versus unseen categories. Notably, neural systems exhibited higher omission rates, LLMs showed a tendency for addition errors, and rule-based systems maintained better overall semantic consistency. However, all system types demonstrated some degree of semantic inconsistency, underscoring the persistence of this challenge across different architectures and methodologies. Please note that we use Omission, Addition and Repetition error types from our consensus taxonomy in Chapter 3. We did not include the Substitution error type because, in our pre-pilot trials, annotating it reliably in a span-based setting proved cognitively demanding.

Overall, this chapter deepens the understanding of semantic errors in text generation, identifies challenges in error annotation practices, and highlights the importance of standardisation and rigorous reproducibility in evaluation studies. Crucially, our findings show that all system types suffer from semantic consistency issues, making it clear that for high-quality, controlled text generation, these errors must be systematically addressed.

The following chapter builds on this insight by exploring methods for automatically identifying semantic consistency errors.

Chapter 5

Automatic Methods for Identifying Semantic Consistency Errors

5.1 Background

Ensuring semantic consistency between input and output is a known weakness in state-of-the-art neural generators. Here, some or all of the input must be matched in specific ways by the output, whether mapping structured data to text or producing free-form text under specified control attributes. There are currently no sufficiently reliable methods for (i) for testing semantic consistency as part of the neural generation process, or (ii) for evaluating systems in terms of the degree to which they achieve it.

This poses particular problems for tasks like data-to-text generation (Corbelle et al., 2022), where the goal is for the output to be entirely controlled by the information given in the input. Having an automatic way to check that input and output carry the same content would be invaluable, both as an online metric, e.g. to rank candidate outputs, and offline for system improvement. However, currently the most reliable way to perform such semantic consistency assessment is through

manual evaluation. An automatic method that can reliably assess the semantic consistency between inputs and outputs directly would be useful not only for evaluation post development and during iterative improvement cycles, but also potentially as part of the text generation method itself, e.g. via reranking of outputs (Harkous et al., 2020b). Nevertheless, reliably assessing semantic consistency via automated techniques continues to be difficult, and evaluation by comparison to human-written reference outputs or through human judgement is predominantly used instead in data-to-text system evaluation, as we previously presented in Chapter 4 (Sections 4.4 and Section A.5).

Prior work has typically framed semantic consistency assessment as either a binary classification task Harkous et al. (2020b) or a mutual textual entailment problem Dušek and Kasner (2020b) between inputs and outputs. Subsequently, methods such as PARENT, which augments n-gram overlap with source-data coverage for table-to-text evaluation (Dhingra et al., 2019), and AlignScore, a unified alignment function trained on diverse tasks to measure factual consistency (Zha et al., 2023), have improved the evaluation of generated text in terms of source-data coverage and factual alignment. More recently, transformer-based metrics have been developed explicitly to assess the semantic fidelity of LLM-generated outputs (Raj et al., 2023). However, fully automatic approaches to semantic consistency evaluation still fall short of reliably matching human judgements.

In this chapter, we propose an automatic way for checking semantic consistency as input/output pair testing. Here, we investigate to what extent off-the-shelf semantic similarity metrics, as well as commonly used lexical and distance-based metrics, correlate with human evaluation scores of data-to-text system outputs, using a two-point comparison setup where we compare only input triples to generated text. Our aim is to determine which of these metrics are best predictors of human judgements of semantic consistency.

Furthermore, we extend the paradigm of input/output pair testing by using LLMs themselves to perform automated assessment. We prompt LLMs to judge

whether each verbalisation accurately represents its corresponding RDF triple(s). While semantic consistency is our primary focus, we also examine non-semantic aspects of LLM-as-Judge evaluation, such as fluency and grammaticality, to provide a more complete picture of the approach. Our investigation focuses on the reliability and generalisability of the LLM-as-Judge approach. In particular, we aim to determine: (i) whether the LLM-as-Judge produces reliable and consistent evaluation results, and (ii) whether the metrics generated by this method generalise well across different datasets and settings, without needing additional meta-evaluation based on human ratings.

Some of the content of this chapter was previously reported in the following publications:

- Huidrom et al. (2025)
- Huidrom and Belz (2025a)
- Huidrom and Belz (2025b)

5.2 Chapter Overview

We systematically investigate the extent to which similarity metrics are reliable predictors of human judgements. In addition, we investigate the reliability and generalisability of LLM-as-Judge approaches to assess semantic consistency. Our explorations have resulted in:

1. A new WebNLG-style data-to-text dataset of People and Cities categories, based on named entities from the GREC corpus (Belz et al., 2009).
2. A human-annotated dataset of 100 pairs of input triples and output texts, with 20 input triples from the above new dataset, each verbalised by five different data-to-text systems. These 100 input triple(s) and output text pairs were evaluated for semantic adequacy on a 1-3 Likert scale by three evaluators, the thesis author and two colleagues from the same lab, following the WebNLG

2017 shared task human evaluation guidelines (Shimorina et al., 2018a) (Section 5.3.1.1).

3. A two-point comparison method, comparing off-the-shelf semantic similarity metric scores with human judgements on input triples and generated text pairs, with and without textification of input triples (Section 5.3.1.3).

4. In-depth analysis of results, including recommendations on which similarity metrics best align with human judgements and discussion of their application in data-to-text generation research (Sections 5.3.2 and 5.3.3; Tables 5.1 to 5.3).

5. A novel LLM-as-Judge method for generalisable meta-evaluation (Section 5.4 and Tables 5.4 to 5.7).

6. A novel LLM-as-Judge method for sanity checking human evaluation (Section 5.5 and Tables 5.8 to 5.13).

7. In-depth analysis of results for the above LLM-as-Judge methods, including key observations and findings (Sections 5.4.4 and 5.5.4).

5.3 Semantic Consistency Checking as Input/Output Pair Testing

5.3.1 Two-point comparison method

In this section, we introduce an automatic approach for assessing semantic consistency using input/output pair testing. We examine how well existing semantic similarity metrics correlate with human evaluation scores for data-to-text system outputs. Specifically, we use a two-point comparison that matches input triples directly to the generated text. Our goal is to identify which metrics most effectively predict human judgements of semantic consistency.

5.3.1.1 Data

In our experiments, we use two datasets: (1) human evaluation data from the WebNLG 2017 shared task; and (2) a newly created WebNLG-like data-to-text dataset consisting of triple set and text pairs in the People and Cities categories, based on named entities from the GREC corpus, together with human judgments assessing semantic consistency for these pairs. Fresh data is essential because most available datasets have been used to train systems, particularly the LLM-based ones, making it difficult to test these systems without a new, unseen dataset.

The WebNLG 2017 shared task human evaluation dataset (Shimorina et al., 2018a) comprises 223 samples per system across ten systems, yielding a total of 2,230 generated texts. Each text was evaluated by three evaluators on a three-point likert scale, where '1' corresponds to Incorrectly, '2' to Medium, '3' to Correctly, on three criteria (semantic adequacy, grammaticality and fluency). For each item, an evaluator was presented with an input (set of RDF triples) and a system output (a text) and were asked the question "Does the text correctly represent the meaning in the data?" to evaluate semantic adequacy on a three-point likert scale. In our experiment, we focus only on the semantic adequacy criterion. We randomly selected 100 samples from this dataset. This is our *Dataset A*. Please note that semantic adequacy errors occur when an output fails to preserve the meaning of its input. In other words, the generated text must neither omit nor introduce any information that wasn't present originally. We refer to this as *semantic consistency*, which is defined as to have everything conveyed by the output fully align with and not exceed the information present in the input (See Chapter 1 for more details).

Our new dataset (Dataset B) was created using the named entities from the GREC corpus (Belz et al., 2009) of People and Cities categories. The GREC corpus (version 2.0) comprises 1,941 texts in total, which are introduction sections from Wikipedia articles across five different domains (cities, countries, rivers, people and mountains). In this work, we consider only the People (442 texts) and Cities (243 texts) categories. The dataset creation process is described in detail through Algo-

rithms 1 and 2.

Algorithm 1 Property-to-Entity Mapping Construction

Input:
(a) `GREC_NE.json`: list of entities by category
(b) Ontology property definitions: set of valid properties for each category from the Ontology
(c) DBpedia triple data: all RDF triples (subject, property, object) from DBpedia
Output:
`dico_entities_for_triple_configuration_GREC_NEs.json`: property-to-entity mapping

if `dico_entities_for_triple_configuration_GREC_NEs.json` does NOT exist **then**
  for each category in `GREC_NE.json` **do**
    for each property relevant to the category (from Ontology) **do**
      for each entity in the category **do**
        if there exists at least one triple in DBpedia where the entity is subject and the predicate is the property **then**
          Add the entity to the property's entity list for that category
  Save as `dico_entities_for_triple_configuration_GREC_NEs.json`

The first algorithm constructs a mapping from each category to its ontology-defined properties and, in turn, to the named entities from the GREC corpus that are associated with those properties in DBpedia. Specifically, for a given category (e.g., People), the algorithm identifies each valid property (e.g., birthPlace) and collects all named entities (e.g., "Albert Einstein") that appear as the subject in at least one RDF triple where the predicate matches that property. The result is a structured mapping of the form: category → property → [entities], where each entity is drawn from the GREC corpus and is grounded in DBpedia through at least one relevant triple. This mapping is subsequently used for targeted triple extraction in Algorithm 2. The steps we follow in Algorithm 1 are as follows:

- Read the list of named entities grouped by category from `GREC_NE.json`.
- For each category (e.g., *People*, *Cities*):
  - Retrieve the set of valid properties for the category, as defined in the Ontology.

  - For each entity in the category:
    * Check DBpedia for at least one RDF triple where the entity appears as the subject and the property as the predicate.
    * If such a triple exists, add the entity to the list for that property in the current category.

- Repeat this process for all categories, properties, and entities.
- Save the resulting mapping (category → property → [entities]) to `dico_entities_for_triple_configuration_GREC_NEs.json`.
- This mapping enables targeted triple extraction, which we use in the second algorithm.

The second algorithm builds a dataset of RDF triples for each named entity, applies filtering and sampling, and formats the output in a WebNLG-like XML format. We call this new dataset as 'Dataset B.' The algorithmic process is outlined in Algorithm 2 and consists of the following steps:

- Check whether the file `dico_input_contents_DBp_GREC_NEs.pickle` exists.
  - If not:
    * Traverse the property-to-entity mapping from Algorithm 1.
    * For each category, property, and entity:
      · If the entity is in the target list from `GREC_NE.json`, extract all RDF triples from DBpedia where the entity is the subject and the property is the predicate.
      · Aggregate all such triples per entity.
    * Save the resulting entity-to-triples mapping as `dico_input_contents_DBp_GREC_NEs.pickle`.
  - If the pickle file already exists, load it directly.
- Build a list of all target entities and their categories from `GREC_NE.json`.

Algorithm 2 Triple Extraction, Filtering, Sampling, and WebNLG-like XML Data Creation

Input:
(a) `GREC_NE.json`
(b) `dico_entities_for_triple_configuration_GREC_NEs.json`
Output:
XML file containing the WebNLG-like dataset, grouped by entity

---

if `dico_input_contents_DBp_GREC_NEs.pickle` does NOT exist **then**
for each category in `dico_entities_for_triple_configuration_GREC_NEs.json` **do**
for each property in the category **do**
for each entity in the property's entity list **do**
if the entity is in the target list from `GREC_NE.json` **then**
Extract all (subject, property, object) triples for that entity from DBpedia data
Add these (subject, property, object) triples to the list of triples for the entity in the mapping
Save as `dico_input_contents_DBp_GREC_NEs.pickle` then Load
else
Load `dico_input_contents_DBp_GREC_NEs.pickle`

.....

Initialize an empty list `target_entities`
for each category in `GREC_NE.json` **do**
for each named entity in the category **do**
Add a record with the entity name and category to `target_entities`
for each record in `target_entities` **do**
Let `entity` be the entity name and `category` its category
For this entity and category, retrieve all properties for which the entity is listed in `dico_entities_for_triple_configuration_GREC_NEs.json`
Save these as the valid properties for this entity
for each entity in `target_entities` **do**
Retrieve available triples for this entity from `dico_input_contents_DBp_GREC_NEs.pickle`
For each (entity, property) for this entity:
Get the actual triple (subject, property, object) from the pickle file
Keep only triples whose property is valid for this entity
Optionally, remove triples with unsuitable object values
Set N as the target number of triples for this entity (from WebNLG distribution, e.g., $N = 3$)
if number of triples is more than N **then**
Randomly select N triples
else
Keep all triples
Save these triples for the entity
for each entity in `target_entities` **do**
Group the final triples for the entity
Write all grouped triples to an XML file

- For each entity:
  - Look up the set of valid properties using the property-to-entity mapping.
  - Retrieve all available triples for the entity from the pickle file.
  - Filter the triples to keep only those with valid properties.
  - Optionally, remove triples with unsuitable object values.
  - Set the target number N of triples per entity according to the WebNLG distribution (e.g., $N = 3$).
  - If more than N triples remain, randomly select N; otherwise, keep all.
- Group the final triples for each entity.
- Write the grouped triples to an XML file for downstream use. This is our 'Dataset B.'

For Dataset B, we collected a total of 100 samples, with each of the following five systems generating 20 samples:

1. FORGe (Mille and Dasiopoulou, 2017; Mille et al., 2019) is a portable grammar-based system that maps predicate-argument structures onto sentences by applying a series of rule-based graph-transducers.
2. CycleGT (Guo et al., 2020) is a weakly supervised framework that iteratively bootstraps generation and semantic parsing models by mapping between two modalities (unaligned text and RDF data) to enable joint model improvement.
3. DCU-NLG-Small (Mille et al., 2024) is a combination of FORGe rule-based system with a language model of reduced size (T5), where the rule-based system converts input triples into semantically correct English text and then a language model to paraphrase these text to make it more fluent.
4. DCU-NLG-PBN (Lorandi and Belz, 2024a) is a Mistral 7B Instruct model fine-tuned with Low-Rank Adaptation (LoRA) to improve performance while maintaining computational efficiency.

5. DCU-ADAPT-modPB (Osuji et al., 2024) explores two approaches: (1) using a fine-tuned Flan-T5-large model for triple ordering and structuring, followed by prompt-based surface realisation with five-shot prompting; and (2) directly generating text from input triples using a prompt-based model with five examples. We use the latter approach with the Mistral 7B Instruct model.

For Dataset B, following the WebNLG 2017 human evaluation protocol for semantic adequacy criterion, each triple(s)-text pair was independently assessed by three evaluators. See Figure 5.1 for our human evaluation interface. All evaluators are fluent in English (two of the evaluators are native speakers and colleagues from the same lab, while the third is the author of this thesis) and they were provided with the original WebNLG 2017 evaluation guidelines. In our experiment, we report the average of the three evaluator scores.

In this context, we reuse a selected sample of the WebNLG 2017 human evaluation dataset as Dataset A, and introduce a newly created Dataset B for this study.

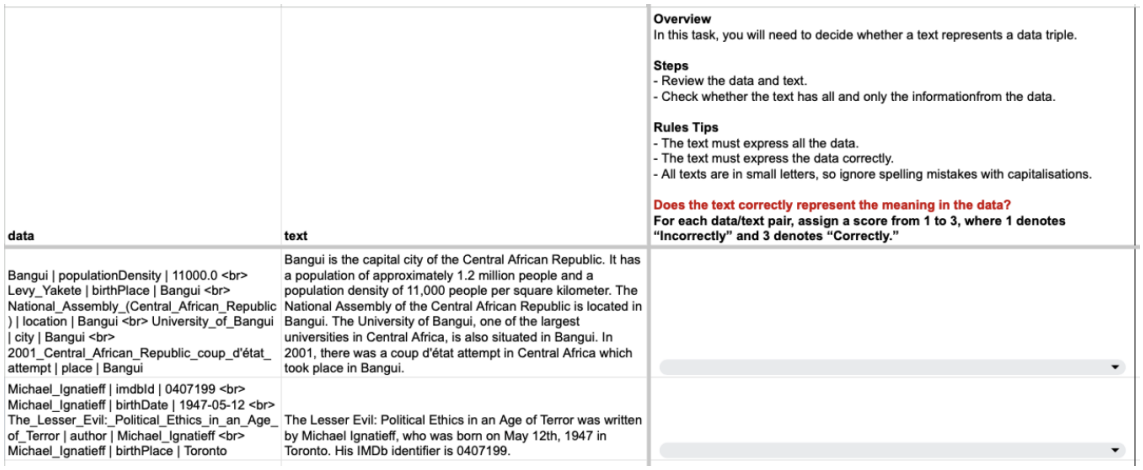

Figure 5.1: Human evaluation interface used for Dataset B evaluation.

5.3.1.2 Metrics used

We define semantic consistency as the requirement that an output's content fully aligns with, and does not exceed, the information in its input. To assess both semantic similarity and dissimilarity under this definition, we use the following 25 automatic metrics (in this experiment, we do not use any reference text):

- **Embedding-based metrics (BERTScore, SBERT variants)**:
  - **BERTScore:**[1] BERTScore (Zhang et al., 2019) compares two sequences by computing similarity between contextual embeddings produced by pre-trained Transformer models. It evaluates how well the tokens in one sequence align with those in the other using cosine similarity.
    - **Precision (P)**: Average of the maximum similarity each token in the first sequence has with tokens in the second.
    - **Recall (R)**: Average of the maximum similarity each token in the second sequence has with tokens in the first.
    - $\mathbf{F}_1$: Harmonic mean of Precision and Recall.
  - **SBERT:**[2] Sentence-BERT (Reimers and Gurevych, 2019a) produces dense sentence-level embeddings using a Siamese or triplet network based on BERT. The similarity between sequences is calculated via standard vector-based measures:
    - **Cosine:** $\dfrac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$, in $[-1, 1]$.
    - **Dot Product:** $\mathbf{u} \cdot \mathbf{v}$, unbounded.
    - **Euclidean:** $\|\mathbf{u} - \mathbf{v}\|_2 = \sqrt{\sum_i (u_i - v_i)^2}$.
    - **Manhattan:** $\|\mathbf{u} - \mathbf{v}\|_1 = \sum_i |u_i - v_i|$.
- **Overlap-based metrics (SacreBLEU, ROUGE, METEOR)**:
  - **SacreBLEU:**[3] SacreBLEU (Post, 2018) standardises the BLEU metric by controlling for tokenisation, smoothing, and formatting, allowing consistent comparison. It computes n-gram precision and penalises overly short outputs via a brevity penalty.
  - **ROUGE:**[4] ROUGE (Lin, 2004) measures the overlap between sequences based on n-grams and longest common subsequences (LCS). It emphasises

[1] `https://github.com/Tiiiger/bert_score`
[2] `https://www.sbert.net`
[3] `https://github.com/mjpost/sacrebleu`
[4] `https://pypi.org/project/rouge/`

recall, capturing how much of a reference is matched.

  * **ROUGE-1**: Based on unigram overlap.
  * **ROUGE-2**: Based on bigram overlap.
  * **ROUGE-L**: Based on longest common subsequence.

    Each ROUGE variant (1, 2, L) is accompanied by:

    · Precision: LCS length divided by candidate length.

    · Recall: LCS length divided by reference length.

    · F_1: Harmonic mean.

- **METEOR:**[5] METEOR (Banerjee and Lavie, 2005) aligns unigrams between sequences using exact, stemmed, and synonym matches. It balances precision and recall, and applies a fragmentation penalty to discourage disordered matches.

- **Edit- and set-based metrics (Levenshtein, Jaccard, Dice)**:

  - **Levenshtein Distance:**[6] Levenshtein (Levenshtein et al., 1966) computes the minimum number of character edits (insertions, deletions, substitutions) needed to transform one string into another. It can be normalised to obtain a similarity score.

  - **Jaccard Similarity:**[7] The Jaccard Index (Jaccard, 1901) measures similarity between two sets as the ratio of their intersection to their union:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

  - **Jaccard Distance:**[8] A dissimilarity measure defined as $1 - J(A, B)$ (Rogers and Tanimoto, 1960).

  - **Dice Coefficient:**[9] The Dice coefficient (Dice, 1945) measures similarity

[5] https://huggingface.co/spaces/evaluate-metric/meteor/tree/main
[6] https://pypi.org/project/Levenshtein/
[7] https://www.geeksforgeeks.org/jaccard-similarity/
[8] https://en.wikipedia.org/wiki/Jaccard_index
[9] https://en.wikipedia.org/wiki/Dice-Srensen_coefficient

as:

$$D(A, B) = \frac{2|A \cap B|}{|A| + |B|},$$

which weights shared elements more heavily than Jaccard.

- **Learned regressors (BLEURT, InferSent, USE)**:
  - **BLEURT:**[10] BLEURT (Sellam et al., 2020) is a regression model fine-tuned on perturbed and human-annotated sentence pairs to predict quality scores aligned with human judgement.
  - **InferSent:**[11] InferSent (Conneau et al., 2017) is a sentence embedding model trained on Natural Language Inference datasets. It produces embeddings useful for similarity tasks using cosine or other distance measures.
  - **Universal Sentence Encoder:**[12] USE (Cer et al., 2018) generates fixed-length embeddings from various architectures trained on multitask learning objectives. Similarity is typically computed using cosine distance.

We apply these metrics to our input/output pairs, where the input RDF triple(s) are processed either with or without textification (see Algorithm 3 for further details).

5.3.1.3 Our approach

In data-to-text generation, human evaluation remains the gold standard for assessing output quality but is both time-consuming and expensive and often requires annotators with domain expertise to ensure reliable judgements (Thomson et al., 2024; Sellam et al., 2020). Consequently, researchers have increasingly turned to automatic, embedding-based metrics and traditional overlap or edit-distance measures (e.g., BERTScore (Zhang et al., 2019); Sentence-BERT (SBERT) (Reimers and Gurevych, 2019b); bleurt (Sellam et al., 2020); SacreBLEU (Post, 2018); ROUGE

[10] `https://github.com/google-research/bleurt`

[11] `https://github.com/facebookresearch/InferSent`

[12] `https://www.tensorflow.org/hub/tutorials/semantic_similarity_with_tf_hub_universal_encoder`

Algorithm 3 Minimal Normalisation of Linearised Triples

Input: List of linearised triple strings L
Output: List of cleaned and normalised triple strings S

$S \leftarrow$ empty list
for each *line* in L **do**
    Replace all occurrences of `|`, `_`, and `<br>` in *line* with a single whitespace
    Replace all consecutive whitespace in *line* with a single whitespace
    Trim leading and trailing whitespace in *line*
    Append *line* to S
return S

(Lin, 2004); Levenshtein distance (Levenshtein et al., 1966)) as low-cost proxies for human ratings. Although these metrics have demonstrated strong correlations with human judgements in purely textual tasks such as machine translation and summarisation, ranging from traditional measures like ROUGE and METEOR (Lin, 2004; Banerjee and Lavie, 2005) to neural metrics such as BERTScore (Zhang et al., 2019) and bleurt (Sellam et al., 2020), they were not originally designed to compare structured inputs, namely, RDF-style triples, to their verbalisations. As such, it is unclear whether off-the-shelf similarity and distance measures remain valid when the source material is graph-structured rather than linear text.

We address this gap by empirically evaluating semantic-similarity, lexical, and edit-distance metrics for data-to-text. We use two complementary datasets of triple–text pairs, each annotated by trained evaluators under the official WebNLG 2017 guidelines (Shimorina et al., 2018b): Dataset A, reused from prior work, and Dataset B, newly created for this study. Full construction and annotation details are in Section 5.3.1.1. For every example, we test two versions: (1) a linearisation ('textification') of the sequence of triples, and (2) the triples in their original RDF format. The linearisation and minimal normalisation procedure is described in Algorithm 3. Here, each linearised triple string is processed by replacing occurrences of special characters (`|`, `_`, and `<br>`) with whitespace, collapsing consecutive spaces into a single space, and trimming leading and trailing whitespace.

We then compute scores from four families of metrics: embedding-based (BERTScore; Zhang et al. (2019); SBERT variants; Reimers and Gurevych (2019b)), overlap-

based (ROUGE; Lin (2004); SacreBLEU; Post (2018); Meteor Banerjee and Lavie (2005)), edit- and set-based (Jaccard similarity and distance; Jaccard (1901); Rogers and Tanimoto (1960); Dice coefficient; Dice (1945)), and learned regression models (bleurt Sellam et al. (2020); InferSent Conneau et al. (2017); USE Cer et al. (2018)). Details of metrics were provided in Section 5.3.1.2. To quantify how well these automatic scores reflect expert judgements, we calculate Pearson's correlation coefficient r between each metric's outputs and the mean human ratings, separately for linearised and non-linearised inputs. By ranking metrics according to their correlation strengths and analysing the effect of input representation, we pinpoint which methods best approximate human assessments and highlight areas where current metrics fall short.

5.3.2 Results and Analysis

Tables 5.1 and 5.2 summarise the Pearson correlation coefficients r between human judgement scores and each automatic metric score, separately for linearised and non-linearised inputs.

5.3.2.1 Dataset A

In Dataset A (human evaluation dataset from WebNLG'17 Shared Task) (see Table 5.1), we observe the following patterns:

- **Embedding-based metrics** (BERTScore, SBERT variants):
  - All three BERTScore variants improve slightly after processing: F_1 rises from 0.421 to 0.441.
  - SBERT–cosine and SBERT–dot product both increase markedly from 0.518 to 0.573.
  - SBERT Euclidean and Manhattan distances see the largest gains, from around 0.50 to 0.582.
- **Overlap-based metrics** (SacreBLEU, ROUGE, METEOR):

	r (−Textification)	r (+Textification)
SBERT (Euclidean)	0.507	0.582
SBERT (Manhattan)	0.502	0.582
SBERT (Cosine)	0.518	0.573
SBERT (Dot Product)	0.518	0.573
bleurt	0.530	0.501
universal sentence encoder	0.248	0.500
BERTScore R	0.417	0.447
BERTScore F_1	0.421	0.441
BERTScore P	0.424	0.433
InferSent	0.372	0.379
meteor	0.206	0.379
ROUGE-L F_1	0.305	0.338
ROUGE-L Precision	0.287	0.337
ROUGE-1 Precision	0.292	0.330
ROUGE-1 F_1	0.309	0.329
ROUGE-L Recall	0.328	0.324
Dice Coefficient	0.290	0.321
Jaccard Similarity	0.290	0.320
ROUGE-1 Recall	0.329	0.313
ROUGE-2 Precision	0.203	0.264
ROUGE-2 F_1	0.202	0.263
ROUGE-2 Recall	0.198	0.257
SacreBLEU	0.228	0.198
Levenshtein Distance	–0.044	–0.062
Jaccard Distance	–0.290	–0.320

Table 5.1: Pearson's r for Dataset A, with and without textification, sorted by r (+Textification).

  - SacreBLEU correlation drops from 0.228 to 0.198, suggesting n-gram precision alone is a weak predictor for structured inputs.
  - ROUGE-1 F_1 rises modestly (0.309 → 0.329) after processing, while ROUGE-2 and ROUGE-L also improve on F_1.
  - METEOR increases notably from 0.206 to 0.379, indicating that synonym and paraphrase matching benefits from normalization.

- **Edit and set-based metrics** (Levenshtein, Jaccard, Dice):
  - Levenshtein distance remains uncorrelated (negative r), dropping further from –0.044 to –0.062.
  - Jaccard similarity and Dice coefficient both rise from ∼0.29 to ∼0.32,

with corresponding negative correlations for Jaccard distance.

- **Learned regressors** (bleurt, InferSent, USE):
  - bleurt shows strong correlation before processing (0.530) but slightly decreases to 0.501 after.
  - InferSent remains relatively flat (0.372 → 0.379).
  - Universal Sentence Encoder jumps substantially from 0.248 to 0.500.

Overall, embedding-based metrics, especially SBERT Euclidean and Manhattan, achieve the highest correlations ($r \approx 0.58$) after triple processing. METEOR and the Universal Sentence Encoder also benefit appreciably from linearisation and minimal normalisation.

5.3.2.2 Dataset B

Table 5.2 shows correlations for Dataset B (our newly created dataset, constructed using GREC-named entities and evaluated by humans following the WebNLG'17 human evaluation guidelines):

- **Embedding-based metrics:**
  - BERTScore F_1 improves Pearson's r from 0.572 to 0.670, the single largest gain among BERTScore variants.
  - SBERT–cosine and SBERT–dot product slightly decrease (0.654 → 0.638), while SBERT Euclidean and Manhattan stay stable around 0.655.
- **Overlap-based metrics:**
  - SacreBLEU plummets from 0.400 to 0.127, indicating that brevity and precision penalties heavily penalize processed triples.
  - ROUGE-1 F_1 increases from 0.246 to 0.387; ROUGE-L F_1 rises from 0.223 to 0.356.
  - METEOR shows a modest uptick (0.296 → 0.306).

	r (−Textification)	r (+Textification)
bleurt	0.713	0.733
BERTScore F_1	0.572	0.670
SBERT (Euclidean)	0.655	0.655
SBERT (Manhattan)	0.662	0.655
BERTScore P	0.636	0.647
SBERT (Cosine)	0.654	0.638
SBERT (Dot Product)	0.654	0.638
BERTScore R	0.431	0.609
universal sentence encoder	0.271	0.402
ROUGE-1 F_1	0.246	0.387
ROUGE-L F_1	0.223	0.356
ROUGE-1 Recall	0.182	0.323
meteor	0.296	0.306
ROUGE-1 Precision	0.268	0.305
ROUGE-L Precision	0.248	0.304
ROUGE-L Recall	0.158	0.281
SacreBLEU	0.400	0.127
ROUGE-2 F_1	0.097	0.127
ROUGE-2 Recall	0.051	0.082
ROUGE-2 Precision	0.111	0.078
Jaccard Similarity	-0.010	0.078
Dice Coefficient	-0.021	0.074
InferSent	-0.148	-0.022
Jaccard Distance	0.010	-0.078
Levenshtein Distance	-0.384	-0.400

Table 5.2: Pearson's r for Dataset B, with and without textification, sorted by r (+Textification).

- **Edit- and set-based metrics:**
  - Levenshtein distance becomes more negatively correlated (–0.384 → –0.400).
  - Jaccard similarity and Dice coefficient show small numerical increases (e.g., Jaccard similarity –0.010 → 0.078), but remain in the range of negligible correlation ($|r| < 0.1$).
- **Learned regressors:**
  - bleurt achieves the highest overall correlation, increasing from 0.713 to 0.733.
  - InferSent remains uncorrelated (–0.148 → –0.022).
  - Universal Sentence Encoder improves from 0.271 to 0.402.

On Dataset B, bleurt consistently yields the strongest alignment with human ratings, while SBERT distances and BERTScore F_1 also perform competitively ($r \approx 0.67$–0.73).

5.3.2.3 Cross-Dataset Comparison

Rank	Metric	Mean r
1	SBERT (Euclidean)	0.6185
2	SBERT (Manhattan)	0.6185
3	bleurt	0.6170
4	SBERT (Cosine)	0.6055
5	SBERT (Dot Product)	0.6055
6	BERTScore F_1	0.5555
7	BERTScore Precision	0.5400
8	BERTScore Recall	0.5280
9	Universal Sentence Encoder	0.4510
10	ROUGE-1 F_1	0.3580
11	ROUGE-L F_1	0.3470
12	METEOR	0.3425
13	ROUGE-L Precision	0.3205
14	ROUGE-1 Recall	0.3180
15	ROUGE-1 Precision	0.3175
16	ROUGE-L Recall	0.3025
17	Jaccard Similarity	0.1990
18	Dice Coefficient	0.1975
19	ROUGE-2 F_1	0.1950
20	InferSent	0.1785
21	ROUGE-2 Precision	0.1710
22	ROUGE-2 Recall	0.1695
23	SacreBLEU	0.1625
24	Jaccard Distance	−0.1990
25	Levenshtein Distance	−0.2310

Table 5.3: Ranking of metrics by average Pearson's r across both datasets with textification. The **Mean r** column represents the arithmetic mean of each metric's r on Dataset A and Dataset B with textification.

Comparing both datasets (Datasets A and B):

- **Triple processing**: Processing (linearisation/normalisation) generally boosts the performance of embedding and set-based metrics, with larger effects on metrics that incorporate semantic matching (USE, METEOR) than on standard n-gram precision (SacreBLEU).

- **Best proxies**:
  - *Dataset A*: SBERT Euclidean/Manhattan and SBERT–cosine (all $r > 0.58$).
  - *Dataset B*: bleurt (up to $r = 0.733$), followed by BERTScore F_1 and SBERT variants ($r \approx 0.67$).
- **Limitations of edit-distance**: Levenshtein distance fails to correlate positively in either dataset, reaffirming that simple character-level edits are poor proxies for semantic fidelity in structured data-to-text tasks.

These findings suggest that for data-to-text evaluation, practitioners should favor embedding-based and learned-regressor metrics, particularly bleurt and SBERT-based distances, over traditional n-gram or edit-distance measures. However, even the best-performing metrics achieve only moderate correlations (for e.g., top mean $r \approx 0.62$), indicating that none are reliably aligned with human judgments in this task context. Table 5.3 provides a ranking of all metrics by mean r across both datasets with textification.

5.3.3 Discussion

Our results demonstrate that off-the-shelf embedding-based metrics are substantially better proxies for human judgements in data-to-text generation than traditional n-gram or edit-distance measures. In both Dataset A and Dataset B, SBERT-based distances (Euclidean and Manhattan) and SBERT–cosine exhibit the largest gains from triple linearisation and achieve the strongest correlations with human ratings in Dataset A ($r \approx 0.58$ after processing). In Dataset B, bleurt emerges as the top performer ($r = 0.733$ after processing), closely followed by BERTScore F_1 and SBERT variants ($r \approx 0.67$).

The consistent improvement of embedding-based metrics after applying minimal linearisation and normalisation to the RDF triples suggests that pre-processing structured inputs to resemble natural text benefits semantic matching. Metrics

that rely on contextualised token embeddings can exploit this pre-processed form to better align source and target, whereas pure n-gram precision (SacreBLEU) and character-level edits (Levenshtein distance) either fail to improve or become more negatively correlated once the triples are linearised.

Overlap-based metrics such as ROUGE and METEOR show only modest gains: ROUGE-1 and ROUGE-L F_1 scores improve by about 0.02–0.04 in both datasets, while METEOR's correlation jumps substantially in Dataset A (0.206 $\rightarrow$ 0.379) but remains relatively low overall. This indicates that synonym/stem matching and paraphrase modules help to some extent, but cannot match the semantic depth captured by embedding-based approaches.

Interestingly, the Universal Sentence Encoder exhibits one of the largest relative improvements in both datasets (Dataset A: 0.248 $\rightarrow$ 0.500; Dataset B: 0.271 $\rightarrow$ 0.402), highlighting its versatility across data-to-text comparisons. In contrast, InferSent fails to provide reliable alignment (Dataset B: $r \approx 0$), suggesting that its NLI-trained embeddings are less suited to the WebNLG-style verbalisation task.

The variation in top-performing metrics across datasets underscores that the choice of automatic proxy may depend on the specific characteristics of the data-to-text domain. bleurt's strong performance in Dataset B likely benefits from its fine-tuning on human-judgement data, whereas SBERT distances prove more robust across both datasets.

Below are our recommendations:

- **Preprocessing structured inputs:** Simple linearisation and normalisation of RDF triples is highly recommended before applying semantic-similarity metrics.
- **Embedding-based metrics:** SBERT distances, BERTScore and bleurt should be preferred as automatic proxies in data-to-text evaluation.
- **Learned regressors:** bleurt offers the highest correlation when available, but requires additional model downloads and computational resources.
- **Traditional metrics:** SacreBLEU and Levenshtein distance are not reliable

indicators of semantic fidelity in structured data-to-text tasks and should be used with caution.

Overall, we observe that embedding-based metrics, particularly SBERT distances, BERTScore F_1 and bleurt, consistently correlate more strongly with human judgements in data-to-text generation than traditional n-gram or edit-distance measures. Simple linearisation and normalisation of RDF triples helps. While no metric achieves perfect alignment ($r < 0.74$) in our evaluation, learned regressors like bleurt offer the best proxy when resources permit. However, our maximum r of 0.733 remains below the system-level correlations reported in the literature, Zhang et al. (2019) found BERTScore system-level Pearson r above 0.80 on machine translation benchmarks, and Sellam et al. (2020) report bleurt correlations up to 0.85 on the WMT Metrics shared task. This gap is certainly notable and points to clear room for improvement. That said, it is worth keeping in mind that the scores reported in the literature mentioned above are for text-to-text tasks, while our work focuses on data-to-text generation. We evaluate how well metrics developed for text-to-text settings capture semantic consistency in data-to-text generation, where judgments are grounded in structured RDF triples. We provide a task-specific baseline with a comparative ranking and analysis of off-the-shelf metrics for data-to-text. This is intended as a practical starting point for selecting metrics and a reference for developing new, input-grounded metrics. In future work, it would be valuable to explore simple ensembles, such as majority-vote or weighted sums, of top-performing metrics from different families to see whether their combination can exceed our current 0.74 ceiling.

5.4 Generalisable Meta-Evaluation of LLM-as-Judge Metrics

Human evaluation is widely considered the most reliable form of system evaluation in natural language processing (NLP) (van Miltenburg et al., 2023b). However, the

substantial cost and expertise required for human evaluation experiments constrain its broader application. Moreover, studies have shown a range of issues in how human evaluations are conducted that cast doubt on some of their results, including loose application of experimental and statistical methods (Thomson et al., 2024).

The emergence of large language models (Touvron et al., 2023; Chaplot, 2023; Cohere, 2024; Yang et al., 2025) has caused a paradigm shift in text generation and understanding across many domains (Ouyang et al., 2022; Kojima et al., 2022). LLMs are exhibiting state-of-the-art performance in problem-solving and reasoning tasks (Mizrahi et al., 2024; Zhang et al., 2024b). LLMs also hold out the appealing vision of cheaper human-like evaluation, demonstrating adaptability and generalisation capabilities Li et al. (2024).

Given their ability to understand and follow provided instructions, they are now beginning to be used in place of human evaluators in approaches known as 'LLM-as-Judge.' In this section, we conduct a series of LLM-as-Judge experiments using verbatim human evaluation instructions to standardise prompts. We then compare the results to those obtained with more conventional LLM prompt formulations, evaluating all approaches against human judgement scores in our data-to-text meta-evaluation.

The aim of this study is to examine how closely LLM judgements align with human assessments across multiple criteria in two NLP data-to-text tasks. To standardise prompt formulation, we use the same instructions as those provided in human evaluations, and compare them with more conventional LLM prompts, in conjunction with single models and model combinations of both varying and comparable sizes. LLMs offers the appealing vision of cheaper human-like evaluation, demonstrating adaptability and generalisation capabilities (Li et al., 2024). While individual human judges are subject to inter-rater variability and require multiple annotators for reliability, LLMs may provide more consistent judgements when resources are constrained. LLM-as-Judge approaches do address some of the issues with human evaluation, such as cost and evaluator inconsistency, but their reliabil-

ity when applied to new tasks needs to be demonstrated via correlation tests with human judgements.

5.4.1 Experimental Requirements

5.4.1.1 Dataset descriptions

In this experiment, we use two datasets. The WebNLG 2020 dataset (Castro Ferreira et al., 2020) is a data-to-text benchmark that aligns sets of RDF triples (subject, predicate and object) with corresponding text. The English dataset has 1,779 triple sets in test set. For the human evaluation, 10% of the test dataset (178 items) was sampled and evaluated on outputs from each team's primary submission (14 submission systems + 3 baseline systems). Each sample, comprising RDF triple(s) and a system-generated text (no reference text), was rated on a 0–100 Likert scale, with each receiving three annotations.

Rotowire (Wiseman et al., 2017) is a widely used data-to-text benchmark which contains NBA basketball game statistics and textual summaries for them ($\sim$ 5k instances). The ReproNLP 2023 shared task (Belz and Thomson, 2023) carried out two reproductions (Arvan and Parde, 2023; van Miltenburg et al., 2023a) of the human evaluation in Puduppully and Lapata (2021) which uses this dataset. In the human evaluation, five systems were evaluated across three criteria on 200 instances per criterion, resulting in a total of 600 evaluated instance. Each instance received three ratings, and participants were asked to rank summaries as either 'A' or 'B'.

5.4.1.2 Evaluation criteria

In our LLM-as-judge experiment on WebNLG'20 dataset, using the same human evaluation instruction, we use the following five criteria taken verbatim from the WebNLG'20 Shared Task (Castro Ferreira et al., 2020):

- **Data Coverage:** Does the output text include descriptions of all predicates presented in the data?

- **Relevance:** Does the output text describe only such predicates (with related subjects and objects), which are found in the data?

- **Correctness:** When describing predicates which are found in the data, does the text mention correct the objects and adequately introduces the subject for this specific predicate?

- **Text Structure:** Is the text grammatical, well-structured, written in acceptable English?

- **Fluency:** Is it possible to say that the text progresses naturally, forms a coherent whole and it is easy to understand the text?

In our LLM-as-judge experiment on ROTOWIRE dataset, using the same human evaluation instruction, we use the original definitions of the following three criteria in Puduppully and Lapata (2021):

- **Grammaticality**: Is the summary written in well-formed English?

- **Coherence**: Is the summary well structured and well organized and does it have a natural ordering of the facts?

- **Repetition**: Does the summary avoid unnecessary repetition including whole sentences, facts or phrases?

5.4.1.3 LLMs used

We use the following LLM models:

- Llama3-8B-Instruct (Touvron et al., 2023):[13] Meta's Llama 3 series model in the smaller 8B parameter size is pretrained, instruction-tuned, but also optimised for dialogue-based applications.

[13] `https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct`

- Mistral-7B-Instruct-v0.2 (Chaplot, 2023):[14] Mistral-7B-Instruct is a language model designed to follow instructions, generate creative text, and handle requests, fine-tuned from Mistral-7B-v0.2 using a diverse range of public conversation datasets.
- C4AI Command R+ (Cohere, 2024):[15] Cohere's open-weights research release of a 104B parameter model; a multilingual model evaluated in 10 languages for performance, and optimised for a variety of tasks including reasoning, summarization, and question answering.
- Qwen2.5-7B-Instruct-1M (Yang et al., 2025):[16] Alibaba's Qwen series model in the smaller 7B parameter size is fine-tuned, instruction-tuned and is optimised to handle long-context tasks while maintaining its capability in short tasks.

5.4.2 LLM-as-judge method for generalisable meta-evaluation

WebNLG'20 LLM-as-judge experiments

In the original WebNLG 2020 evaluation, each paired RDF triple set and system output was evaluated by three human evaluators. We obtain individual scores with each of the following three LLMs, then compute the mean of the three scores from different model and prompt combinations:

- J_H: LLM judgements using as the prompt the verbatim instructions from the original human evaluation in WebNLG 2020.
- J_{C+D}: LLM judgements using as the prompt conventional minimal zero-shot LLM prompts also incorporating the verbatim evaluation criterion definitions.
- J_{C-D}: Same as J_{C+D} minus the definitions.
- H: For comparison, we also test single human judgements from WebNLG'20 as predictors.

[14] `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2`
[15] `https://huggingface.co/CohereForAI/c4ai-command-r-plus-4bit`
[16] `https://huggingface.co/Qwen/Qwen2.5-7B-Instruct-1M`

We use the following models in our experiments: Llama3-8B-Instruct, Mistral-7B-Instruct-v0.2 and C4AI Command R+ models.

Rotowire LLM-as-judge experiments

In the original ROTOWIRE evaluation, system summaries were evaluated by three human evaluators. We obtain individual ratings with each of our three LLMs, then compute the majority vote of the three ratings. This time we use just the for-human instructions as in the original human evaluation. We test the correlations between the following LLM (combination)s and human judgements:

- H_1 and H_2: Two sets of human judgements obtained from two reproductions Arvan and Parde (2023); van Miltenburg et al. (2023a) of Puduppully and Lapata (2021).
- J_{H_V}: Majority vote of LLM judgements by models of varying sizes (7B, 8B, 104B) and using the same human instructions (same models as in the WebNLG 2020 tests).
- J_{H_C}: Majority vote of LLM judgements on models of comparable sizes (two 7Bs and one 8B) and using the same human instructions. These are Llama3-8B-Instruct, Mistral-7B-Instruct-v0.2 and Qwen2.5-7B-Instruct-1M.

We use the same models as for WebNLG in the J_{H_V} tests, and replace the Cohere model with Qwen2.5-7B-Instruct-1M.

Experiment Setup

We execute (a) both the human evaluation instructions from the WebNLG'20 shared task (Ferreira et al., 2019) and our custom prompts designed for the WebNLG'20 LLM-as-Judge experiments (see Appendix A.4.4 for details), as well as (b) the human evaluation instructions from (Puduppully and Lapata, 2021) for the Rotowire LLM-as-Judge experiments. All prompts are applied as zero-shot inference inputs to the LLMs listed in Section 5.4.1.3. We run the experiments with three different

seeds (42; 1738; 1,234), meaning each score in tables below is the average of the outputs from the different seed runs. All experiments use English data.

For hyperparameters, we set the temperature to 0.001 and top-p to 1, with the maximum length set to 1024 for WebNLG'20 experiments and 128 for ROTOWIRE experiments, respectively. We quantise the models to 4-bit and run our experiments on a single rtxa6000/a100 GPU.

5.4.3 Results and Analysis

	Correctness				Data Coverage				Fluency				Relevance				Text Structure			
	H	J_H	J_{C+D}	J_{C-D}	H	J_H	J_{C+D}	J_{C-D}	H	J_H	J_{C+D}	J_{C-D}	H	J_H	J_{C+D}	J_{C-D}	H	J_H	J_{C+D}	J_{C-D}
AAI	93.53	97.62	97.04	95.16	94.39	96.29	97.37	91.21	90.29	95.58	94.59	90.67	95.20	99.57	97.69	92.89	92.95	97.19	95.40	88.47
F17	90.14	97.13	97.88	94.54	92.07	94.67	97.72	90.56	80.94	95.11	94.70	90.29	92.59	99.62	98.25	92.15	85.74	97.14	95.41	87.45
F20	92.31	97.78	97.99	95.47	93.42	96.32	97.96	91.49	82.6	95.76	95.09	91.19	94.31	99.97	98.28	93.19	87.89	97.31	95.58	88.53
bt5	93.58	96.57	95.71	94.33	93.84	95.54	96.23	90.69	88.69	94.66	93.75	90.23	95.22	99.68	97.29	92.26	91.91	97.29	95.04	87.73
cuni	91.59	95.63	95.30	95.06	93.29	94.71	96.19	91.51	87.64	94.18	92.94	90.84	94.56	99.67	96.92	93.03	90.75	97.2	94.42	88.48
CGT	89.85	94.56	96.17	94.35	91.23	93.86	97.02	90.63	84.82	92.83	93.21	90.07	93.37	99.40	98.27	92.28	87.88	96.98	94.91	87.48
D-SGU	92.49	96.12	95.44	93.66	95.32	95.08	96.27	90.05	78.59	93.31	90.92	88.61	94.86	99.8	97.28	91.46	83.50	96.52	92.38	86.33
FB-AI	92.70	97.35	97.31	95.18	93.17	96.30	97.50	91.25	90.84	95.87	94.98	90.86	93.9	99.88	98.06	92.90	93.09	97.51	95.67	88.48
H_Lab	80.76	85.82	88.54	90.48	84.74	86.93	92.52	88.24	75.21	82.78	83.92	85.24	85.27	96.11	94.55	88.59	80.22	92.16	88.16	82.86
NILC	76.70	77.64	81.75	88.34	81.61	79.28	86.64	84.58	74.85	77.17	78.93	82.82	83.52	91.87	90.56	84.74	80.46	88.62	86.88	80.98
NUIG	92.05	96.06	95.49	95.02	92.06	95.18	96.53	91.41	88.90	94.68	93.83	90.61	94.06	99.14	97.31	92.85	91.59	97.35	95.06	88.23
O-NLG	74.98	74.29	77.35	85.00	79.96	77.68	82.68	83.94	75.68	73.12	74.83	79.90	79.89	88.03	86.74	81.65	80.46	85.14	84.43	78.71
OSU	93.41	96.57	95.78	95.16	95.12	95.48	96.67	91.14	90.07	95.50	93.83	90.72	94.62	99.31	97.38	93.04	92.44	97.41	95.10	88.65
RALI	92.13	97.54	96.52	94.56	95.20	96.20	96.69	90.82	77.76	94.86	92.53	89.83	94.81	99.74	97.52	92.48	81.84	97.07	94.12	87.46
TGEN	88.63	95.64	95.02	96.29	88.18	94.62	95.55	92.64	86.16	94.43	92.83	91.28	92.64	99.46	96.99	94.14	89.04	97.31	94.42	89.01
UPC	74.37	79.59	83.86	89.27	75.85	81.59	89.06	87.61	72.28	77.63	79.57	84.00	82.05	94.66	93.68	87.68	78.50	88.82	86.46	81.77
W-REF	94.15	97.59	97.64	95.01	95.44	95.99	97.71	91.02	89.85	95.54	95.38	90.67	94.39	99.80	98.35	92.96	92.11	97.28	95.83	88.16
Avg	88.43	92.56	93.22	**93.35**	90.29	92.10	**94.72**	89.93	83.25	**90.77**	90.34	88.7	91.49	**97.98**	96.18	91.08	87.08	**95.19**	92.90	86.4

Table 5.4: System-level average scores for each quality criterion by WebNLG'20 human judges (H), average over Llama3-8B-Instruct/Mistral-7B-Instruct-v0.2/Command R+ prompted with full human instructions (J_H); conventional zero-shot prompt with (J_{C+D}) and without definitions (J_{C-D}). System names (rows) with length > 4 letters are shortened by concatenating the first letter or first two letters with the last two/three letters.

	Coherence				Repetition				Grammaticality			
	H_1	H_2	J_{H_V}	J_{H_C}	H_1	H_2	J_{H_V}	J_{H_C}	H_1	H_2	J_{H_V}	J_{H_C}
Gold	49.79	**56.25**	**70.00**	**70.00**	49.16	**52.92**	**70.83**	**73.75**	54.62	**57.08**	**70.83**	**64.58**
Template	**62.76**	40.00	18.75	24.58	**72.15**	47.08	22.92	26.25	**58.58**	38.33	32.08	42.08
ED+CC	42.50	46.25	42.08	41.67	36.97	47.50	44.17	41.67	40.17	45.83	37.50	40.42
Hier	44.77	54.58	60.42	56.67	42.62	50.42	56.25	51.67	45.19	54.58	52.92	49.17
Macro	50.21	52.92	58.75	57.08	49.15	52.08	55.83	56.67	51.48	54.17	56.67	53.75

Table 5.5: System-level average scores for each quality criterion by two sets of Rotowire human judges (H_1, H_2), average majority vote by varying-size models Llama3-8B-Instruct/Mistral-7B-Instruct-v0.2/Command R+ (J_{H_V}), and average majority vote by Llama3-8B-Instruct/Mistral-7B-Instruct-v0.2/Qwen2.5-7B-Instruct-1M (J_{H_C}).

	Single Human Judges Avg	Human Instructions as prompt mean of 3 scores by:				Zero-shot + original definitions mean of 3 scores by:				Zero-shot - original definitions mean of 3 scores by:			
		Mistral	Llama	CRplus	Mistral+ Llama+ CRplus	Mistral	Llama	CRplus	Mistral+ Llama+ CRplus	Mistral	Llama	CRplus	Mistral+ Llama+ CRplus
Correctness	0.69	0.93	0.94	0.99	**0.97**	0.72	0.93	0.98	0.95	0.90	0.25	0.98	0.92
Data Coverage	0.68	0.89	0.86	**0.96**	0.93	0.62	0.84	**0.96**	0.88	0.77	0.21	0.93	0.79
Fluency	0.68	0.67	0.75	**0.81**	0.78	0.48	0.84	**0.81**	0.80	0.74	0.68	0.79	0.79
Relevance	0.69	0.85	0.90	**0.98**	0.94	0.67	0.93	0.96	0.91	0.93	0.66	0.96	0.93
Text Structure	0.69	0.49	0.70	0.79	0.76	0.16	0.79	**0.87**	0.83	0.79	0.74	0.79	0.82

Table 5.6: Pearson's correlations with the aggregated WebNLG'20 human scores, achieved by single human judges and different LLMs.

Mean scores

Table 5.4 presents the system-level average scores per evaluation criterion for the WebNLG evaluation. We observe that human evaluators and LLM judges generally agree with each other, where AAI, F17, F20, OSU, and W-REF often emerging as top performers and, O-NLG and UPC consistently rated lower by both human and LLM judges across multiple criteria.

Moreover, the averages of system-level scores (last row) by LLMs are higher than those by humans in all cases except three averages produced by the zero-shot prompt without definitions (J_{C-D}).

Table 5.5 presents the system-level average scores per evaluation criterion for the two Rotowire human evaluations (H_1, H_2), and the two types of majority vote, one with a much larger model in the mix (J_{H_V}), and one with similar sized models (J_{H_C}) Overall, both human and LLM judges agree on the high performance of the Gold (human-produced reference outputs), although H_1 uniquely favours the Template system. Additionally, while J_{H_V} and J_{H_C} yield similar evaluations for top-performing systems, J_{H_C} tends to assign slightly higher scores for lower-performing systems (e.g., Template) in Coherence and Repetition.

Correlations with human judgements

Table 5.6 reports the correlations with the original WebNLG'20 human judgements achieved by: (i) individual human judges on average, (ii) each of the LLM model (combination)s. Strikingly, individual human judges have far lower agreement with

	H_1	H_2	J_{H_V}	J_{H_C}
Coherence				
H_1	1.00	-0.59	-0.63	-0.55
H_2	-0.59	1.00	0.99	0.98
J_{H_V}	-0.63	0.99	1.00	0.99
J_{H_C}	-0.55	0.98	0.99	1.00
Grammaticality				
H_1	1.00	-0.19	0.13	0.36
H_2	-0.19	1.00	0.93	0.81
J_{H_V}	0.13	0.93	1.00	0.97
J_{H_C}	0.36	0.81	0.97	1.00
Repetition				
H_1	1.00	-0.28	-0.62	-0.48
H_2	-0.28	1.00	0.90	0.94
J_{H_V}	-0.62	0.90	1.00	0.98
J_{H_C}	-0.48	0.94	0.98	1.00

Table 5.7: Pearson's correlation matrix for Rotowire / Coherence, Grammaticality & Repetition.

the mean of the other judges (on the same outputs) than the LLMs. Another clear result is that the different models are affected very differently by differences in prompts: all perform broadly similarly with the verbatim human instructions; Mistral scores collapse when human instructions are removed and definitions are retained (J_{C+D}), but recover when the definitions are also removed(J_{C-D}); and Llama scores are unaffected by the removal of human instructions (J_{C+D}), but collapse when the definitions are also removed (J_{C-D}). The Command R+ models does best with the human instructions but largely retains its performance under the other two conditions.

Table 5.7 shows the complete correlation matrices between the two sets of Rotowire human judges and the two majority voting combinations of LLMs, for each of the three evaluation criteria. Here, the most striking result is the stark discrepancy between the two sets of human judges: H_1 has a medium strong *negative* correlation with both H_2 and the LLMs for Coherence, weak or no correlation for Grammaticality, and weak or medium *negative* correlation for Repetition. In contrast H_2 and LLM combinations all agree strongly with each other. H_1 and H_2 also produced different reproducibility assessments compared to the original evaluation by Pudup-

pully and Lapata (2021), as reported in the ReproNLP 2023 shared task report (Belz and Thomson, 2023). We discuss this further in the next section.

5.4.4 Discussion

We have presented results for experiments with LLM-as-judge approaches for two types of data-to-text tasks and eight evaluation methods, using various prompt formulations. For the WebNLG'20 experiments, we tested three types of prompts: (i) standardised prompts i.e., verbatim human evaluation instructions from previous shared evaluations, (ii) conventional prompts i.e., minimal zero-shot LLM prompts that also incorporated the verbatim evaluation criterion definitions, and (iii) another conventional prompts i.e., minimal zero-shot prompts without any criterion definitions. For the Rotowire experiments, we used standardised prompt formulations only i.e., the verbatim human evaluation instructions as prompts. See Appendix A.4 for more details. Throughout all scenarios, standardised prompt formulations were shown to work better than more conventional prompt formulations, irrespective of task or the length of input/output.

An unexpected discovery was that LLMs can serve as sanity-checks for human evaluations. The ReproNLP shared task organisers had no basis for deciding which of two reproductions of Puduppully and Lapata (2021) they reported gave a truer picture: either Repro 1 (H_2 in this section) was right and the work had excellent reproducibility, or Repro 2 (H_1) was right and it had terrible reproducibility. Because both of our LLM majority votes strongly agreed with Repro 1 and strongly disagreed with Repro 2, the indication is that Repro 1 (H_2) gave the truer results out of the two reproductions. Further discussion of LLM-as-Judge as sanity checker for human evaluation experiments can be found in Section 5.5.

Overall, we have found our best LLMs to be highly reliable predictors of human evaluations, and to benefit from human-type detailed evaluation instructions. The result that individual human judges correlate far less well with overall human judgements than LLMs do, implies that if the choice is between a small number

of human judges and an LLM you are better off using the LLM. Accordingly, two factors plausibly explain this pattern. First, alignment to human rubrics where modern instruction-tuned LLMs are trained to follow natural-language instructions and approximate human preferences. When our prompts restate the evaluation criteria and require explicit comparison to the input triples, the model's decision rule closely matches the human task. Second, reduced underspecification, i.e, detailed, human-type instructions specify the evidence to inspect, the weighting of that evidence, and the intended use of the rating scale, in turn, discourages reliance on surrogate heuristics (for e.g., judging by fluency alone). This focuses the model on the same evidence humans use, yielding higher consistency and more stable use of the scoring scale.

5.5 Sanity Checking Human Evaluation Experiments with LLM-as-Judge

In this section, we investigate an application of LLM-as-judge approaches which comes with its own built-in reliability test, namely for sanity checking the results of reproduction studies. The ReproNLP shared tasks produced sets of three highly comparable human evaluations, one of which was the original study and two were reproductions. In some cases, there were substantial discrepancies between the results of the two reproductions. In such cases, the ReproNLP shared task organisers had no basis for deciding which of the two reproductions gave a truer picture: either reproduction A was right and the work had excellent reproducibility, or reproduction B was right and it had terrible reproducibility. Below we examine how LLM-as-judge results behave in such scenarios and whether they can provide the basis that was missing in the ReproNLP shared task.

The primary aim of our study is to investigate an application of LLM-as-judge approaches to serve as sanity checkers in human evaluation experiments. We conduct LLM-as-judge experiments to compare with the results of reproduction studies that

emerged from the ReproNLP shared tasks (Belz and Thomson, 2023, 2024). In our analysis, each reproducibility study includes (i) a set of human-produced system level scores from the original study (O); (ii) two sets of human-produced system level scores derived from reproduction efforts ($H1$ and $H2$) from the ReproNLP shared tasks; as well as (iii) LLM-produced scores obtained from different ensembles of LLMs (J) from our experiments.

As secondary objectives, we also explore the generalisability and reliability of LLM-as-judge, building on the work in Section 5.4, by: (a) testing verbatim human evaluation instructions in LLM-as-Judge metrics as a way of standardising prompt formulation, (b) testing combinations of models of varying vs. similar sizes and, and our main contribution (c) investigating the behaviour of LLM-as-judge methods when compared to multiple identical human evaluations, more specifically, an original human evaluation experiment and two reproduction studies.

Here, we consider three assessment scenarios, each with a unique task, dataset, quality criterion, and rating instrument. The tasks are: (i) generation of fact-checking explanations, (ii) dialogue summarisation, and (iii) data-to-text generation with macro planning. These correspond to the LIAR-PLUS dataset, the AMI Meeting Corpus, and the ROTOWIRE dataset, respectively. For quality evaluation, each scenario uses a specific criterion from among coverage, *Informativeness*, *Grammaticality*, *Coherence*, or *Conciseness/Repetition*. Finally, each scenario employs a different rating instrument: ranking explanations as 1, 2, or 3; a 5-point Likert scale; or ranking summaries as best or worst.

The reliability of using LLMs as sanity checkers in this scenario is assessed through correlation tests with the human judgements, as reported in Tables 5.9, 5.11 and 5.13.

5.5.1 Experimental Requirements

5.5.1.1 Dataset descriptions

LIAR-PLUS (Alhindi et al., 2018) is a PolitiFact-based dataset that contains 12,836 statements along with their veracity justifications. Atanasova et al. (2020) used this dataset in their original study, which was later reproduced during the ReproNLP'24 Shared Task (Belz and Thomson, 2024) by two teams (Gao et al., 2024; Loakman and Lin, 2024). The reproduction human evaluation experiment was conducted on 40 out of the 80 instances used in the original study, which were randomly selected from the test set, along with three veracity explanations that were collected for each instance.

AMI Meeting Corpus (Carletta et al., 2005) is a meeting summary dataset that contains roughly 100 hours of recorded meetings where each meeting features four participants discussing a remote control design project. Feng et al. (2021) used this dataset in their original study, which was later reproduced during ReproNLP'24 Shared Task (Belz and Thomson, 2024) by two teams (Fresen et al., 2024; Lango et al., 2024b) of the human evaluation. The human evaluation experiment was conducted on summaries generated for 10 randomly selected dialogues.

ROTOWIRE (Wiseman et al., 2017) is a widely used benchmark which contains NBA basketball game statistics and textual summaries for them ($\sim$ 5k instances). ReproNLP'23 Shared Task (Belz and Thomson, 2023) carried out two reproductions (Arvan and Parde, 2023; van Miltenburg et al., 2023a) of the human evaluation in Puduppully and Lapata (2021) which uses this dataset. The human evaluation experiment was conducted on the basis of 20 summaries and 10 system pairs (200 items).

5.5.1.2 Evaluation criteria

In our LLM-as-judge experiments, we use the criteria and their definitions verbatim from the three reproduction studies.

First, Atanasova et al. (2020) used coverage, non-redundancy and non-contradiction

as the evaluation criteria in their human evaluation experiment, out of which the reproduction studies consider *Coverage* only, defined as follows.

Coverage: The explanation includes important, salient information and does not omit any key points that contribute to the fact-check.

Second, Feng et al. (2021) used informativeness, conciseness and coverage as the evaluation criteria in their human evaluation experiment, out of which the reproduction studies consider *Informativeness* only, defined as follows.

Informativeness: Informativeness measures whether the abstract contains the key information from the original conversation.

Third, Puduppully and Lapata (2021) used *Grammaticality*, *Coherence* and *Conciseness/Repetition* as the evaluation criteria in their human evaluation experiment. The reproduction studies consider all criteria in the original study. The following are the original definitions of the three criteria.

- **Grammaticality**: Is the summary written in well-formed English?
- **Coherence**: Is the summary well structured and well organized and does it have a natural ordering of the facts?
- **Conciseness/Repetition**: Does the summary avoid unnecessary repetition including whole sentences, facts or phrases?

5.5.1.3 LLMs used

We use the following LLM models:

- C4AI Command R+ Cohere (2024) (abbreviating as, C):[17] Cohere's open-weights research release of a 104B parameter model; a multilingual model evaluated in 10 languages for performance, and optimised for a variety of tasks including reasoning, summarisation, and question answering.
- Deepseek-Llama3-70B-Instruct (DeepSeek-AI, 2025) (abbreviating as, D):[18] Deepseek release their first-generation reasoning models including versions that

[17] `https://huggingface.co/CohereForAI/c4ai-command-r-plus-4bit`
[18] `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B`

have been distilled. Deepseek-Llama3-70B-Instruct model is a distilled variant of a 70B parameter Llama model, fine-tuned with comprehensive reasoning instructions.

- Granite-7B-Instruct (Sudalairaj et al., 2024) (abbreviating as, G):[19] IBM's Granite 7B model is instruction-tuned with curated human instructions and optimised for task-specific performance and efficient in-context learning.

- Llama3-8B-Instruct Touvron et al. (2023) (abbreviating as, L):[20] Meta's Llama 3 series model in the smaller 8B parameter size is pretrained, instruction-tuned, but also optimised for dialogue-based applications.

- Llama3.3-70B-Instruct[21] Grattafiori et al. (2024) (abbreviating as, L70): Meta's Llama 3.3 series model in the 70B parameter size is an instruction-tuned text-only model optimised for multilingual dialogues.

- Mistral-7B-Instruct-v0.2 (Chaplot, 2023) (abbreviating as, M):[22] Mistral-7B-Instruct is a language model designed to follow instructions, generate creative text, and handle requests, fine-tuned from Mistral-7B-v0.2 using a diverse range of public conversation datasets.

- Qwen2.5-7B-Instruct-1M (Yang et al., 2025) (abbreviating as, Q):[23] Alibaba's Qwen series model in the smaller 7B parameter size is fine-tuned, instruction-tuned and is optimised to handle long-context tasks while maintaining its capability in short tasks.

- Qwen2-72B-Instruct (qwe, 2024) (abbreviating as, Q72):[24]Alibaba's Qwen2 series model in 72B parameter size is fine-tuned, instruction-tuned and supports a context length of up to 131,072 tokens, enabling the processing of extensive inputs.

[19] https://huggingface.co/ibm-granite/granite-7b-instruct
[20] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[21] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct
[22] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[23] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct-1M
[24] https://huggingface.co/Qwen/Qwen2-72B-Instruct

5.5.2 LLM-as-judge method for sanity checking human evaluation experiments

LLM-as-judge method for Atanasova et al. (2020) experiment

In the original evaluation, system explanations were evaluated by four human evaluators. We obtain individual per-item rankings (measured as ranks 1, 2 or 3) with each of our nine LLMs ensembles, then compute the mean average rank (MAR) of the four rankings. We test the correlations between the following LLM (combination)s and human judgements:

- O: Original human judgements from Atanasova et al. (2020).
- H_1 and H_2: Two sets of human judgements obtained from two reproductions Gao et al. (2024); Loakman and Lin (2024) of Atanasova et al. (2020).
- Recalc. of O, H_1 and H_2: We recalculated human judgments for O, H_1, and H_2. To enable a direct comparison, we reprocessed the original paper's raw data, restricting it to the 40 inputs shown to evaluators in the reproduction studies. Because the MAR code was unavailable, we computed MAR ourselves to verify the results.
- J_V: Majority vote of LLM judgements by models of varying sizes (7B, 8B, 104B) and using the same human instructions — C+M+L.
- J_{C_S}: Majority vote of LLM judgements on comparable small-sized models (two 7Bs and one 8B) and using the same human instructions — M+L+Q.
- J_{C_L}: Majority vote of LLM judgements on comparable large-sized models (two 70Bs and one 72B) and using the same human instructions — D+L70+Q72.

LLM-as-judge method for Feng et al. (2021) experiment

In the original evaluation, meeting summaries were evaluated by three human evaluators. We obtain individual per-item ratings (measured on a scale of 1 to 5) with each of our three LLMs, then compute the average rating of the three rankings.

We test the correlations between the following LLM (combination)s and human judgements:

- O: Original human judgements from Feng et al. (2021).
- H_1 and H_2: Two sets of human judgements obtained from two reproductions Fresen et al. (2024); Lango et al. (2024b) of Feng et al. (2021).
- J_1, J_4, J_7: Aggregation of LLM judgements by models of varying sizes (two 7Bs, one 70B, one 104B) and using the same human instructions — G+M+C+L70, G+Q+C+L70, Q+M+C+L70.
- J_2, J_5, J_8: Aggregation of LLM judgements by models of varying sizes (two 7Bs, one 72B, one 104B) and using the same human instructions — G+M+C+Q72, G+Q+C+Q72, Q+M+C+Q72.
- J_3, J_6, J_9: Aggregation of LLM judgements by models of varying sizes (two 7Bs, one 70B, one 72B) and using the same human instructions — G+M+L70+Q72, G+Q+L70+Q72, Q+M+L70+Q72.

LLM-as-judge method for Puduppully and Lapata (2021) experiment

In the original evaluation, system summaries were evaluated by three human evaluators. We obtain individual per-item rankings (measured by ranking best or worst) with each of our three LLMs, then compute the majority vote of the three ratings. We test the correlations between the following LLM (combination)s and human judgements:

- O: Original human judgements from Puduppully and Lapata (2021).
- H_1 and H_2: Two sets of human judgements obtained from two reproductions Arvan and Parde (2023); van Miltenburg et al. (2023a) of Puduppully and Lapata (2021).
- J_V: Majority vote of LLM judgements by models of varying sizes (7B, 8B, 104B) and using the same human instructions — C+M+L.

- J_{C_S}: Majority vote of LLM judgements on models of comparable sizes (two 7Bs and one 8B) and using the same human instructions — M+L+Q.

Experiment Setup

We use the for-human instructions as zero-shot inference prompts on the LLMs. We run each experiment with three different seeds (42; 1,738; 1,234), meaning each score in tables below is the average/majority vote of the outputs from these runs. All experiments use English data and the original human evaluation instructions.

We briefly outline the experimental setup. We use the seven large language models listed in Section 5.5.1.3 for our experiments with the following hyperparameters: temperature = 0.001, maximum length = 128, and top-p = 1. We quantise the models to 4-bit and run our experiments on a single rtxa6000/a100 GPU.

5.5.3 Results and analysis

	O	O Recalc. by H_1	O Recalc. by H_2	**O Recalc. by us**	Repro H_1	**H_1 Recalc. by us**	Repro H_2	**H_2 Recalc. by us**	J_V	J_{C_S}	J_{C_L}
Just	**1.48**	**1.47**	**1.52**	**1.46**	2.18	2.18	**1.62**	**1.58**	**1.83**	**1.83**	**1.78**
Explain-MT	1.68	1.70	1.66	1.71	**1.62**	**1.62**	1.78	1.83	1.84	2.02	1.89
Explain-Extr	1.89	1.88	1.87	1.88	1.93	1.93	2.05	2.03	1.97	2.08	2.14

Table 5.8: System-level MAR scores (lower is better) for *Coverage* on LIAR-PLUS dataset for Atanasova et al. (2020). O = original study; H_1, H_2 = reproductions; J_i = LLM ensemble by model size. Recalculated results are also shown.

Atanasova et al. (2020) results

Table 5.8 presents the system-level average MAR scores for *Coverage* on the LIAR-PLUS dataset for the original and reproduction studies for Atanasova et al. (2020), and different LLM judges, grouped for size diversity. In Atanasova et al. experiments, three items at a time were ranked by three human evaluators and the ranks aggregated into a single score via **mean average rank (MAR)**. A lower MAR indicates a better average ranking. For each column, the best results are in bold. We observe the Just system to obtain the best results in the original study O, repro-

	O	O Recalc. by H_1	O Recalc. by H_2	**O Recalc. by us**	Repro H_1	**H_1 Recalc. by us**	Repro H_2	**H_2 Recalc. by us**	J_V	J_{C_S}	J_{C_L}
Original	1.00	1.00	0.99	0.99	-0.43	-0.43	0.99	1.00	0.90	0.96	0.98
Recalc. (H_1)	1.00	1.00	0.98	1.00	-0.51	-0.5	0.98	1.00	0.86	0.98	0.96
Recalc. (H_2)	0.99	0.98	1.00	0.97	-0.34	-0.33	1.00	0.99	0.94	0.93	1.00
Recalc. (Ours)	0.99	1.00	0.97	1.00	-0.55	-0.54	0.96	1.00	0.84	0.99	0.95
Repro (H_1)	-0.43	-0.51	-0.34	-0.55	1.00	1.00	-0.31	-0.48	0	-0.67	-0.25
Repro Recalc. (Ours)	-0.43	-0.50	-0.33	-0.54	1.00	1.00	-0.3	-0.48	0.01	-0.66	-0.25
Repro (H_2)	0.99	0.98	1.00	0.96	-0.31	-0.3	1.00	0.98	0.95	0.92	1.00
Repro **Recalc. (Ours)**	1.00	1.00	0.99	1.00	-0.48	-0.48	0.98	1.00	0.87	0.98	0.97
J_V	0.90	0.86	0.94	0.84	0	0.01	0.95	0.87	1.00	0.75	0.97
J_{C_S}	0.96	0.98	0.93	0.99	-0.67	-0.66	0.92	0.98	0.75	1.00	0.89
J_{C_L}	0.98	0.96	1.00	0.95	-0.25	-0.25	1.00	0.97	0.97	0.89	1.00

Table 5.9: Pearson's r correlation matrix for LIAR-PLUS dataset / *Coverage* for Atanasova et al. (2020) by the original study O, reproduction studies H_1 and H_2, LLM ensembles J_i and the recalculations.

duction study H_2 and the LLM judgements, except for H_1 where the Explain-MT system is the best, for both reported and recalculated MAR values.

Table 5.9 reports the correlations (Pearson's r) between the human judgements and the LLM judgements. We observe one set of reproduction results (H_1) in contradiction with the original evaluation (O), with the other set of reproduction results (H_2) and with the LLM judgement results (J_V, J_{C_S}, J_{C_L}). The original results (O) and their recalculations (by H_1, H_2, and by us) demonstrate near-perfect correlations (0.99), confirming the reliability of the H_1 and H_2 evaluation. In contrast, one set of reproduction results (H_1) shows negative correlations (-0.43) with the original, while the other set (H_2) is highly consistent with the original (0.99-1.00). Furthermore, the LLM-based judgements (J_V, J_{C_S}, J_{C_L}) strongly correlate with both the original and H_2 results, but inversely with H_1, providing a sanity check of such contradictory results (see also discussion section).

Please note that due to the lack of code for calculating MAR scores in the original study (O), both reproduction studies (H_1 and H_2) independently recalculated the original scores by following the algorithm described in Atanasova et al.'s pa-

per. Similarly, we also performed our own recalculations for both the original and reproduction results, adhering to the same published algorithm. In the tables, the recalculated values by us and LLM-based values are displayed in black, while all other values (from the original study, the reproduction studies, and the recalculations by the reproduction studies) are shown in gray.

				LLM judges								
	O	H_1	H_2	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9
Gold	4.70	2.40	4.60	4.63	4.78	4.63	4.3	4.45	4.3	4.53	4.68	4.53
PGN	2.92	2.18	1.53	4.13	3.66	3.58	3.58	3.11	3.03	3.93	3.46	3.38
HMNet	**3.52**	2.20	**2.68**	**4.30**	**3.83**	**3.72**	**3.83**	**3.35**	**3.24**	**4.12**	**3.64**	**3.53**
PGN(DKE)	3.20	2.18	1.93	4.08	3.60	3.53	3.58	3.10	3.03	3.99	3.52	3.44
PGN(DRD)	3.15	**3.00**	1.90	4.22	3.72	3.64	3.69	3.19	3.12	3.93	3.43	3.36
PGN(DTS)	3.05	2.28	1.85	4.08	3.63	3.46	3.57	3.12	2.95	3.98	3.53	3.36
PGN(DALL)	3.33	2.53	1.85	4.01	3.58	3.35	3.43	3.00	2.77	3.87	3.44	3.21

Table 5.10: System-level average scores (higher the better) for *Informativeness* on AMI dataset for Feng et al. (2021). O = original study; H_1, H_2 = reproductions; J_* = LLM ensembles of varying model sizes.

	O	H_1	H_2	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9
O	1.00	0.01	0.99	0.89	0.96	0.93	0.91	0.96	0.93	0.95	0.97	0.94
H_1	0.01	1.00	-0.03	0.06	0.02	0.02	0.01	0	0	-0.15	-0.08	-0.09
H_2	0.99	-0.03	1.00	0.94	0.97	0.96	0.96	0.98	0.96	0.98	0.98	0.97
J_1	0.89	0.06	0.94	1.00	0.96	0.98	0.99	0.96	0.97	0.95	0.91	0.94
J_2	0.96	0.02	0.97	0.96	1.00	0.99	0.97	1.00	0.99	0.97	0.99	0.99
J_3	0.93	0.02	0.96	0.98	0.99	1.00	0.98	0.99	1.00	0.97	0.97	0.99
J_4	0.91	0.01	0.96	0.99	0.97	0.98	1.00	0.97	0.99	0.97	0.94	0.96
J_5	0.96	0	0.98	0.96	1.00	0.99	0.97	1.00	0.99	0.98	0.99	0.99
J_6	0.93	0	0.96	0.97	0.99	1.00	0.99	0.99	1.00	0.98	0.97	0.99
J_7	0.95	-0.15	0.98	0.95	0.97	0.97	0.97	0.98	0.98	1.00	0.98	0.99
J_8	0.97	-0.08	0.98	0.91	0.99	0.97	0.94	0.99	0.97	0.98	1.00	0.99
J_9	0.94	-0.09	0.97	0.94	0.99	0.99	0.96	0.99	0.99	0.99	0.99	1.00

Table 5.11: Pearson's r correlation matrix for AMI dataset / *Informativeness* for Feng et al. (2021).

Feng et al. (2021) results

Table 5.10 presents the system-level average scores for *Informativeness* on the AMI dataset for the original study, the reproduction studies, and different LLM judges of varied sizes, for Feng et al. (2021). The higher the system-level average scores, the better. The best results are by the gold (first row), which consists of the human-written references. For each column, the second best results are in bold. We observe

the HMNet system to obtain the second best result for the original study O, reproduction study H_2 and LLM judgements, except for H_1 where the PGN (DRD) system is the second best, for the reported average scores.

Table 5.11 reports correlations (Pearson's r) between the human judgements and the LLM judgements. We observe one set of reproduction results (H_1) in contradiction with the original (O), with the other set of reproduction results (H_2) and with the LLM judgement results (J_1 to J_9). The correlation between the original results and the second reproduction (H_2) is extremely high (0.99), which indicates that the H_2 reproduction replicates the original findings with near-perfect fidelity. In contrast, the correlation between the original results and the first reproduction (H_1) is nearly zero (0.01), indicating a random relationship between the scores. Additionally, the direct correlation between H_1 and H_2 is also random (-0.03), suggesting that the H_1 reproduction yields results that are not aligned with those of the original study, thereby raising concerns regarding its validity. Furthermore, the LLM-based judgements (J_1 to J_9) strongly correlate with both the original and the H_2 results, but inversely with H_1, providing a sanity check of such contradictory results.

We observe near-zero correlations between H_1 and both J_5 and J_6 (-0.00158 and -0.00470, respectively, which round off to 0 when expressed to two decimal places as shown in Table 5.11). These near-zero values indicate an extremely weak or negligible linear relationship between the corresponding evaluation scores, suggesting that the correlation is entirely random. Such results occur in correlations between H_1 and J_5 and J_6 (the LLM combinations of G+Q+C+Q72 and G+Q+L70+Q72 respectively).

	Coherence					Repetition					Grammaticality				
	O	H_1	H_2	J_{C_S}	J_V	O	H_1	H_2	J_{C_S}	J_V	O	H_1	H_2	J_{C_S}	J_V
Gold	**46.25***	-0.42	**12.5**	**40.00***	**40.00**	**30.83**	-1.67	**5.83**	**47.50***	**41.67***	**38.33**	9.17	**14.17**	**29.17**	**41.67**
Template	-52.92*	**25.42**	-20.00*	-50.83*	-62.50*	-36.67*	**43.75***	-5.83	-47.50*	-54.17	–61.67	**17.08**	-23.33*	-15.83	-35.83*
ED+CC	-8.33	-15.00	-7.50	-16.67*	-15.83*	-4.58	-25.83	-5.00	-16.67	-11.67	5.00	-19.58	-8.33	-19.17	-25.00*
Hier	4.58	-10.42	9.17	13.33	20.83	3.75	-14.58	0.83	3.33	12.50	13.33	-9.58	9.17	-1.67	5.83
Macro	10.42	0.42	5.83	14.17	17.50	6.67	-1.67	4.17	13.33	11.67	5.00	2.92	8.33	7.50	13.33

Table 5.12: System-level average scores for *Coherence*, *Repetition* and *Grammaticality* on Rotowire dataset (Puduppully and Lapata, 2021). O = original study; H_1, H_2 = reproductions; J_* = LLM ensemble by model size. * indicates a statistically significant difference ($p<0.05$) between Macro and the other systems.

	O	H_1	H_2	J_{C_S}	J_V
Coherence					
O	1.000	-0.572	0.930	0.980	0.964
H_1	-0.572	1.000	-0.584	-0.547	-0.625
H_2	0.930	-0.584	1.000	0.982	0.992
J_{C_S}	0.980	-0.547	0.982	1.000	0.993
J_V	0.964	-0.625	0.992	0.993	1.000
Grammaticality					
O	1.000	-0.420	0.912	0.695	0.831
H_1	-0.420	1.000	-0.185	0.358	0.133
H_2	0.912	-0.185	1.000	0.814	0.931
J_{C_S}	0.695	0.358	0.814	1.000	0.969
J_V	0.831	0.133	0.931	0.969	1.000
Repetition					
O	1.000	-0.622	0.871	0.984	0.991
H_1	-0.622	1.000	-0.277	-0.482	-0.619
H_2	0.871	-0.277	1.000	0.935	0.898
J_{C_S}	0.984	-0.482	0.935	1.000	0.981
J_V	0.991	-0.619	0.898	0.981	1.000

Table 5.13: Pearson's correlation matrix for Rotowire / Coherence, Grammaticality & Repetition for Puduppully and Lapata (2021).

Puduppully and Lapata (2021) results

Table 5.12 presents the system-level average scores for *Coherence, Grammaticality, Repetition* on the ROTOWIRE dataset for the reproduction studies of Puduppully and Lapata (2021), and different LLM judges of varied and similar sizes. The higher the system-level average scores, the better. For each column, the best results are in bold. We observe the Gold system as the best results for original study O, reproduction study H_2 and LLM judgements (J_{C_S} and J_V), except for H_1 where the Template system is the best, for the reported average scores across all criteria.

Table 5.13 shows the complete correlation matrices between the three sets of Rotowire human judges and the two majority voting combinations of LLMs, for each of the three evaluation criteria. Here, the most striking result is the stark discrepancy between the two sets of human judges: H_1 shows a moderately strong *negative* correlation with both O and H_2 for Coherence and with O for Repetition; and a weak to moderate *negative* correlation with both O and H_2 for Grammaticality as well as with H_2 for with Repetition. In contrast H_2 and LLM combinations all

correlate strongly and positively with each other and with O, providing evidence of the reliability of H_2.

5.5.4 Discussion

Our findings demonstrate that the best-performing LLMs are highly reliable predictors of human evaluations and benefit from human-style detailed evaluation instructions.

In all three assessment scenarios, we encountered a situation where one of a set of directly comparable human evaluations contradicted the others. In each case, the directly comparable LLM judges we created came down strongly against the contradicting evaluation, and strongly in favour of the others. If we consider that an evaluation can be wrong in many different ways, but right only in a small number of ways, then the likelihood of the LLMs all *agreeing* with a *wrong* evaluation, while *disagreeing* with a *right* evaluation is vanishingly small. We take this to mean that in this situation (strong agreement with one human evaluation, strong disagreement with another), the LLM can be used to sanity-check the two contradicting human evaluations, casting the deciding vote which is more reliable.

Although our strongest LLM judges correlate well with human judgements, several risks remain. Even with careful prompts, models can rely on proxies such as fluency rather than strict input–output alignment, scores can be sensitive to prompt wording and model version, and biases such as length, order, or family self-preference may appear as well as absolute scales can drift over time, and there are risks of data leakage as well as privacy and auditability constraints for closed models. In this study we mitigate these issues by using input-grounded, criterion-specific rubrics, randomising item order, fixing prompts and decoding settings while logging model versions, evaluating across model families, and anchoring results to human ratings. As further safeguards, future work can combine multiple judges through simple ensembles, include calibration items to monitor scale drift, and allow judges to abstain when uncertain.

5.6 Chapter Summary

In this chapter, we explore methods for automatically identifying semantic consistency errors. Our work has resulted in three main contributions.

First, we investigated an automatic approach to assessing semantic consistency via input/output pair testing. We examine how well a representative range of existing semantic similarity metrics correlate with human evaluation scores for data-to-text system outputs. Specifically, we use a two-point comparison that matches input triples directly to the generated text. Our findings revealed embedding-based metrics such as SBERT and BERTScore, as well as learned metrics like bleurt, showed the strongest alignment with human ratings, particularly when inputs were preprocessed through linearisation and normalisation.

Second, we examined how closely LLM judgments align with human assessments across multiple criteria in two NLP data-to-text tasks, using both existing human evaluation guidelines and custom-designed prompts to guide the LLMs. Our analysis found that LLM-based evaluations correlate strongly with human assessments and, in some cases, outperform individual human annotators in aligning with the overall consensus. These findings underscore the effectiveness of LLMs as highly reliable proxies for human evaluation, particularly when provided with human-type detailed evaluation instructions.

Third, we investigated an application of LLM-as-judge approaches to serve as sanity checkers in human evaluation experiments. Specifically, we examined their ability to resolve contradictory human evaluations by serving as a deciding vote. Our experiments showed that LLM-as-Judge can reliably identify the more accurate evaluation in cases of disagreement, and demonstrate promising reliability and generalisability across tasks and prompts.

Overall, this chapter deepens the understanding of automated semantic consistency assessment and highlights the utility of LLMs as well as existing automatic metrics for input/output pair testing, as effective tools for enhancing the evaluation of semantic consistency.

Chapter 6

Conclusion

6.1 Contributions

This thesis contributes to advancing the methods and evaluation of semantic consistency in input-controlled NLG, with a particular focus on data-to-text generation. The contributions are grouped as follow:

1. Theoretical and Formal Foundations of Semantic Consistency:

   (a) **Definition and formalisation of semantic consistency and semantic error types:** The thesis defines semantic consistency as the requirement that everything conveyed by the output must fully align with, and not exceed, the information in the input. It also clearly defines semantic errors as content/meaning errors and through our proposed consensus taxonomy introduces three main types: *omission*, *addition*, and *substitution*. These definitions address the lack of standardisation across existing error annotation schemes and provide a consistent foundation for evaluating semantic consistency in generated text. In Chapter 4, we apply the highest level of error types from our consensus taxonomy in a span-based human-annotation study of data-to-text systems.

   (b) **Development of a task-agnostic consensus error taxonomy:** The thesis investigates the extent to which existing semantic error annotation

schemes and taxonomies reflect a common standard. Building on this analysis, it creates a standardised, task-agnostic consensus taxonomy for semantic errors, designed to be applicable across diverse input-controlled language generation tasks and domains. We later apply this taxonomy in a span-based human-annotation study, reported in Chapter 4.

2. Human-Centred Methods for Semantic Error Analysis

   (a) **Comprehensive survey of existing annotation schemes and taxonomies:** The thesis reviews error annotation schemes across both human- and machine-generated text, identifying gaps and inconsistencies in prior work. This review motivates the adoption of a simplified, generalisable taxonomy suitable for evaluating input-controlled generation settings, reported in Chapter 4.

   (b) **Span-based semantic error annotation study across multiple systems:** The thesis applies the consensus taxonomy to a span-level annotation study of outputs from different data-to-text systems, enabling fine-grained analysis of where and how semantic inconsistencies manifest in generated text. Unlike prior output-only schemes (for e.g., Kasner and Dusek (2024)), we annotate aligned spans on both the input (RDF triples) and the output, annotating them and using empty-span annotation for omissions/additions. This dual-sided annotation makes omissions directly observable and, to our knowledge, is novel across multiple data-to-text systems evaluation.

3. Automatic Methods for Semantic Consistency Evaluation

   (a) **Semantic consistency checking via input/output comparison:** This thesis assesses the extent to which common automatic metrics such as, lexical, embedding-based, and distance-based, correlate with human judgments of semantic consistency in system outputs, establishing baseline reliability of these methods.

(b) **Evaluating LLM-as-judge methods for generalisability:** This thesis introduces a novel method of using prompted large language models to predict human-like quality ratings. It investigates the extent to which these models can reliably assess multiple quality dimensions, including semantic consistency, across different systems and evaluation conditions.

(c) **Using LLM-as-judge for sanity-checking human evaluation:** This thesis contributes a novel LLM-as-judge method for sanity-checking human evaluation experiments. By comparing LLM-based judgments with results from an original human evaluation and two reproduction studies, it examines the alignment and divergence across repeated human assessments. Additionally, it tests prompt formulation using verbatim human instructions and analyses model size effects, offering new insights into the reliability and generalisability of LLM-based evaluation.

6.2 Research Questions Revisited

RQ1. How can semantic errors in generated text be systematically categorised and annotated in a way that enables task-agnostic evaluation of semantic consistency in input-controlled generation tasks? This question is addressed in Chapter 3, where a consensus semantic error taxonomy is developed by synthesising multiple existing error categorisation schemes and error taxonomies for semantic errors. The standardised and generalised taxonomy of content/meaning error types has three semantic error types on its highest level, *omission*, *addition*, and *substitution*. These error types are defined based on the relationship between the output content and the input. Specifically, an error occurs when the output (1) omits required input content (*omission*), (2) introduces content not supported by the input (*addition*), or (3) distorts input content in a meaning-changing way (*substitution*). The taxonomy is intended to serve as a task-agnostic framework for identifying and annotating semantic errors in input-controlled generation settings.

In Chapter 4, we evaluate this taxonomy by applying its top-level error categories in a span-based human-annotation study of data-to-text systems.

RQ2. How can human annotation be used to investigate semantic consistency errors in system-generated text, and what can it reveal about the kinds of semantic errors present in existing language generation systems? This question is addressed in Chapter 4, where existing semantic error annotation schemes and taxonomies developed for both human- and machine-generated text are systematically reviewed. The surveyed error annotation schemes and taxonomies are analysed in terms of their target text types, error categories, NLP tasks, annotation purposes, and structural properties. This analysis is conducted to identify common limitations, such as inconsistent definitions, overlapping categories, and ambiguity between fluency and meaning errors, and to motivate the adoption of a simplified, task-agnostic taxonomy suited for systematic evaluation of semantic consistency. Following this, the top-level categories from our consensus taxonomy are used in a span-based human annotation study to evaluate semantic consistency in outputs from multiple data-to-text systems. Please note that a subset of the works surveyed here is used in Chapter 3 to derive the consensus taxonomy for content/meaning errors.

RQ3. To what extent can automatic evaluation methods, including lexical, embedding-based, and distance-based similarity metrics and prompt-based large language models, reliably reflect human judgements of semantic consistency and non-semantic quality dimensions (e.g., fluency, coherence) in NLG outputs, and how generalisable are these methods across systems and evaluation conditions? This question is addressed in Chapter 5, which first examines how well different types of automatic metrics such as, lexical, embedding-based, and distance-based, correlate with human evaluation scores on system-generated outputs from a data-to-text generation task. This provides a baseline understanding of how effectively traditional automatic metrics capture aspects

of meaning preservation. The chapter then introduces a prompt-based large language model setup (LLM-as-judge) to evaluate system outputs across both semantic consistency and non-semantic quality dimensions, including fluency, coherence, and grammaticality. The chapter also examines the generalisability and reliability of the LLM-as-judge approach across different systems and evaluation prompts, providing insights into the extent to which such models can simulate or support human evaluation. In addition, the chapter explores the use of LLM-as-judge as a tool for sanity-checking human evaluation experiments by comparing LLM predictions with results from repeated human evaluations. This enables an assessment of how consistently LLMs align with human ratings across different assessment scenarios.

6.3 Limitations

While this thesis makes several contributions toward improving the method and evaluation of identifying and assessing semantic consistency in input-controlled natural language generation (NLG), it also presents certain limitations that should be acknowledged:

Task and Domain Scope: While the core empirical focus of the thesis is on data-to-text (D2T) generation tasks involving structured input, some components, such as the LLM-as-judge sanity-checking experiments, also include non-D2T generation tasks. However, the development and primary validation of the consensus taxonomy and annotation approaches were conducted within D2T settings. Therefore, further work is needed to confirm the taxonomy's applicability and effectiveness across a wider range of NLG tasks such as summarisation, open-domain dialogue, etc (Fabbri et al., 2021; Durmus et al., 2020).

Annotation Complexity and Scalability: Span-level semantic error annotation, while offering fine-grained insights, is time-intensive and cognitively demanding for annotators. This limits the scalability of such methods, especially in large-scale

evaluations or in real-world deployment scenarios. Moreover, inter-annotator agreement in reproduction studies revealed variability depending on annotator expertise and training, suggesting that consistent annotation remains a non-trivial challenge (Stenetorp et al., 2012; Belz and Thomson, 2023, 2024).

Dependence on Pretrained Models: The LLM-as-judge framework introduced in the thesis leverages large pretrained models for evaluating semantic consistency. While the results demonstrate promising alignment with human ratings, these models themselves are susceptible to biases, prompt sensitivity, and inconsistent outputs depending on their configuration or training data. This introduces uncertainty in their use as robust evaluators (Belz and Thomson, 2023, 2024).

Limited Exploration of Semantic Control in Generation: Although the thesis investigates how semantic errors can be detected and annotated, and how evaluation can be improved, it does not propose or evaluate generation models or decoding strategies that explicitly aim to improve semantic controllability during the generation process. However, the semantic consistency methods developed here, in particular, span-based human annotation protocols and LLM-as-judge scoring techniques, can be used to guide generation indirectly. For example, automatic semantic scoring can support decoding-time reranking to select more consistent outputs, or serve as a training-time signal in reinforcement or contrastive learning settings. Future work could close the loop between evaluation and generation by feeding evaluation signals back into model training or decoding (Dušek and Kasner, 2020a).

Focus on Error Types, Not Severity: The consensus taxonomy focuses on categorising the type of semantic error (e.g., omission, addition, substitution), but not the severity or impact of these errors on user understanding or task performance. An error's significance may vary by domain, yet the framework treats all instances equally, which may oversimplify the practical implications of semantic failure (Lommel et al., 2014).

6.4 Future Work

This thesis provides a foundation for identifying and assessing semantic consistency in input-controlled NLG, particularly in data-to-text (D2T) scenarios. Building on the findings and limitations discussed, several avenues for future research emerge:

Broader Validation Across NLG Tasks: The consensus taxonomy and evaluation methods developed in this thesis were primarily validated in structured data-to-text generation. Future work could extend this validation to other input-controlled generation tasks such as abstractive summarisation, question answering, or instruction following, to assess the generalisability of both the taxonomy and annotation methodology (Fabbri et al., 2021; Durmus et al., 2020; Kryscinski et al., 2020; Laban et al., 2022).

Model-Integrated Semantic Control: The evaluation tools developed in this thesis could be used to guide or regularise the training and decoding of generation models. Future work could investigate how semantic error detection signals—either from human annotations or automatic checkers, can be incorporated into training objectives, constrained decoding, or reinforcement learning setups to enhance semantic controllability during generation (Dušek and Kasner, 2020a; Dhingra et al., 2019).

Improving Annotation Efficiency and Reliability: As span-based annotation is cognitively demanding, developing semi-automated or assistive tools for guiding human annotators may increase both efficiency and consistency. Active learning, annotation recommendation, or error highlighting techniques could be explored to support more scalable human evaluation workflows (Stenetorp et al., 2012).

Robustness and Transparency in LLM-as-Judge Frameworks: While LLMs showed potential as evaluation agents, their performance is sensitive to prompt design, temperature settings, and internal biases. Future work should explore ro-

bust prompting techniques, calibration strategies, and interpretability tools to make LLM-as-judge setups more transparent, reliable, and fair across different evaluation conditions and domains (Liu et al., 2023; Zhong et al., 2022; Wataoka et al., 2024; Belz and Thomson, 2023, 2024; Wataoka et al., 2024).

From Evaluation to Intervention: Finally, future work could explore how the insights gained from semantic error analyses can be translated into practical system interventions. This includes using diagnostic signals to adapt model architectures, improve fine-tuning datasets, or personalise generation behaviour based on user needs and error tolerance in downstream applications such as healthcare or finance (Huidrom et al., 2024).

6.5 Concluding Remarks

This thesis has investigated the challenge of evaluating semantic consistency in input-controlled natural language generation (NLG), with a particular focus on data-to-text (D2T) generation. Despite advances in the fluency and coherence of neural generation systems, ensuring that generated text faithfully reflects the meaning of its structured input remains a persistent problem. This work addresses that gap by introducing principled methods for detecting, categorising, and evaluating semantic errors, defined as content-level discrepancies between input and output.

The thesis makes several contributions toward improving semantic controllability in generation systems. First, it formalises core definitions of semantic consistency and semantic error types, providing a clear conceptual foundation. Second, it develops a task-agnostic consensus taxonomy for semantic errors, *omission*, *addition*, and *substitution*, by synthesising prior annotation schemes and validating it through span-level human annotation studies on D2T outputs. Third, it evaluates both traditional similarity-based metrics and prompt-based large language models (LLM-as-judge) for their ability to reflect human judgments of semantic consistency, including a novel use of LLMs to sanity-check human evaluation experiments.

The results suggest that using a standardised taxonomy in combination with structured human annotation can support more consistent and fine-grained evaluation of semantic errors. In parallel, LLM-based methods show promise as reliable, generalisable evaluators when carefully prompted, offering complementary support for human assessment. Together, these findings reinforce the need for evaluation frameworks that prioritise semantic consistency, moving beyond surface-level metrics to address the semantic integrity of system outputs.

While the scope of the thesis is primarily grounded in D2T tasks, the proposed approaches are intended to be extensible to other input-controlled generation settings. Limitations related to task diversity, annotation scalability, and model biases remain, suggesting avenues for future work.

In summary, this thesis contributes both theoretical and practical advancements toward more robust evaluation of semantic consistency in NLG. As language generation systems continue to be deployed in meaning-critical applications, ensuring semantic consistency will be essential to their trustworthiness, safety, and effectiveness.

Bibliography

(2024). Qwen2 technical report.

Al Sharou, K., Li, Z., and Specia, L. (2021). Towards a better understanding of noise in natural language processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 53–62.

Al Sharou, K. and Specia, L. (2022). A taxonomy and study of critical errors in machine translation. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 171–179.

Alhindi, T., Petridis, S., and Muresan, S. (2018). Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the first workshop on fact extraction and verification (FEVER)*, pages 85–90.

Arvan, M. and Parde, N. (2023). Human evaluation reproduction report for data-to-text generation with macro planning. In Belz, A., Popović, M., Reiter, E., Thomson, C., and Sedoc, J., editors, *Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems*, pages 89–96, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Atanasova, P., Simonsen, J. G., Lioma, C., and Augenstein, I. (2020). Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364.

Balloccu, S., Belz, A., Huidrom, R., Reiter, E., Sedoc, J., and Thomson, C. (2024). Proceedings of the fourth workshop on human evaluation of nlp systems

(humeval)@ lrec-coling 2024. In *Proceedings of the Fourth Workshop on Human Evaluation of NLP Systems (HumEval)@ LREC-COLING 2024*.

Banerjee, S. and Lavie, A. (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.

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., et al. (2023). A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Bao, F. S., Li, M., Qu, R., Luo, G., Wan, E., Tang, Y., Fan, W., Tamber, M. S., Kazi, S., Sourabh, V., et al. (2024). Faithbench: A diverse hallucination benchmark for summarization by modern llms. *arXiv preprint arXiv:2410.13210*.

Barbagli, A., Lucisano, P., Dell'Orletta, F., Montemagni, S., and Venturi, G. (2016). Cita: An l1 italian learners corpus to study the development of writing competence. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 88–95.

Bavaresco, A., Bernardi, R., Bertolazzi, L., Elliott, D., Fernández, R., Gatt, A., Ghaleb, E., Giulianelli, M., Hanna, M., Koller, A., et al. (2024). Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. *arXiv preprint arXiv:2406.18403*.

Belem, C. G., Pezeshkpour, P., Iso, H., Maekawa, S., Bhutani, N., and Hruschka, E. (2024). From single to multi: How llms hallucinate in multi-document summarization. *arXiv preprint arXiv:2410.13961*.

Belkebir, R. and Habash, N. (2021). Automatic error type annotation for arabic. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 596–606.

Belz, A. (2025). Qra++: Quantified reproducibility assessment for common types of results in natural language processing. *arXiv preprint arXiv:2505.17043*.

Belz, A., Agarwal, S., Graham, Y., Reiter, E., and Shimorina, A. (2021). Proceedings of the workshop on human evaluation of nlp systems (humeval). In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*.

Belz, A., Agarwal, S., Shimorina, A., and Reiter, E. (2020). ReproGen: Proposal for a shared task on reproducibility of human evaluations in NLG. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 232–236, Dublin, Ireland. Association for Computational Linguistics.

Belz, A., Kow, E., Viethen, J., and Gatt, A. (2009). Generating referring expressions in context: The grec task evaluation challenges. In *Conference of the European Association for Computational Linguistics*, pages 294–327. Springer.

Belz, A., Mille, S., Thomson, C., and Huidrom, R. (2024). Qcet: An interactive taxonomy of quality criteria for comparable and repeatable evaluation of nlp systems. In *Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations*, pages 9–12.

Belz, A., Popović, M., Reiter, E., and Shimorina, A. (2022). Proceedings of the 2nd workshop on human evaluation of nlp systems (humeval). In *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval)*.

Belz, A., Popović, M., Reiter, E., Thomson, C., and Sedoc, J. (2023). Proceedings of the 3rd workshop on human evaluation of nlp systems. In *Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems*.

Belz, A. and Thomson, C. (2023). The 2023 ReproNLP shared task on reproducibility of evaluations in NLP: Overview and results. In Belz, A., Popović, M., Reiter, E., Thomson, C., and Sedoc, J., editors, *Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems*, pages 35–48, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Belz, A. and Thomson, C. (2024). The 2024 ReproNLP shared task on reproducibility of evaluations in NLP: Overview and results. In Balloccu, S., Belz, A., Huidrom, R., Reiter, E., Sedoc, J., and Thomson, C., editors, *Proceedings of the Fourth Workshop on Human Evaluation of NLP Systems (HumEval) @ LREC-COLING 2024*, pages 91–105, Torino, Italia. ELRA and ICCL.

Benkirane, K., Gongas, L., Pelles, S., Fuchs, N., Darmon, J., Stenetorp, P., Adelani, D. I., and Sánchez, E. (2024). Machine translation hallucination detection for low and high resource languages using large language models. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9647–9665, Miami, Florida, USA. Association for Computational Linguistics.

Biten, A. F., Gómez, L., and Karatzas, D. (2022). Let there be a clock on the beach: Reducing object hallucination in image captioning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1381–1390.

Blom, J. D. (2010). *A dictionary of hallucinations*. Springer.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Caines, A., Yannakoudakis, H., Edmondson, H., Allen, H., Pérez-Paredes, P., Byrne, B., and Buttery, P. (2020). The teacher-student chatroom corpus. In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 10–20.

Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., et al. (2005). The ami meeting

corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.

Caseli, H. and Inácio, M. (2020). Nmt and pbsmt error analyses in english to brazilian portuguese automatic translations. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3623–3629.

Castro Ferreira, T., Gardent, C., Ilinykh, N., van der Lee, C., Mille, S., Moussallem, D., and Shimorina, A. (2020). The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In Castro Ferreira, T., Gardent, C., Ilinykh, N., van der Lee, C., Mille, S., Moussallem, D., and Shimorina, A., editors, *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.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. (2018). Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Chaplot, D. S. (2023). 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, william el sayed. *arXiv preprint arXiv:2310.06825*.

Cohere (2024). Introducing command r+: A scalable llm built for business. *https://cohere.com/blog/command-r-plus-microsoft-azure*.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Corbelle, J. G., Diz, A. B., Alonso-Moral, J., and Taboada, J. (2022). Dealing with hallucination and omission in neural natural language generation: A use case

on meteorology. In *Proceedings of the 15th International Conference on Natural Language Generation*, pages 121–130.

Costa, Â., Ling, W., Luís, T., Correia, R., and Coheur, L. (2015). A linguistically motivated taxonomy for machine translation error analysis. *Machine Translation*, 29(2):127–161.

Costa, Â., Luís, T., Ribeiro, J., Mendes, A. C., and Coheur, L. (2012). An english-portuguese parallel corpus of questions: translation guidelines and application in smt. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2172–2176.

DeepSeek-AI (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Dhingra, B., Faruqui, M., Parikh, A., Chang, M.-W., Das, D., and Cohen, W. W. (2019). Handling divergent reference texts when evaluating table-to-text generation. *arXiv preprint arXiv:1906.01081*.

Di, Q., Vylomova, E., and Baldwin, T. (2019). Modelling tibetan verbal morphology. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 35–40.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Dickinson, M. (2010). Generating learner-like morphological errors in russian. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 259–267.

Dickinson, M. and Herring, J. (2008). Developing online icall resources for russian. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9.

Dreyer, M., Liu, M., Nan, F., Atluri, S., and Ravi, S. (2021). Evaluating the trade-off between abstractiveness and factuality in abstractive summarization. *arXiv preprint arXiv:2108.02859*.

Du, X., Xiao, C., and Li, S. (2024). Haloscope: Harnessing unlabeled llm generations for hallucination detection. *Advances in Neural Information Processing Systems*, 37:102948–102972.

Durmus, E., He, H., and Diab, M. (2020). FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Dušek, O. and Kasner, Z. (2020a). Evaluating semantic accuracy of data-to-text generation with natural language inference. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 131–137.

Dušek, O. and Kasner, Z. (2020b). Evaluating semantic accuracy of data-to-text generation with natural language inference. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 131–137.

Dušek, O., Novikova, J., and Rieser, V. (2020). Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Comput. Speech Lang.*, 59(C):123–156.

Dziri, N., Madotto, A., Zaïane, O., and Bose, A. J. (2021). Neural path hunter: Reducing hallucination in dialogue systems via path grounding. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2197–2214, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., and Radev, D. (2021). Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Federico, M., Negri, M., Bentivogli, L., and Turchi, M. (2014). Assessing the impact of translation errors on machine translation quality with mixed-effects models. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1643–1653.

Feng, X., Feng, X., Qin, L., Qin, B., and Liu, T. (2021). Language model as an annotator: Exploring dialogpt for dialogue summarization. *arXiv preprint arXiv:2105.12544*.

Ferreira, T. C., Gardent, C., Ilinykh, N., Van Der Lee, C., Mille, S., Moussallem, D., and Shimorina, A. (2020). The 2020 bilingual, bi-directional webnlg+ shared task overview and evaluation results (webnlg+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*.

Ferreira, T. C., van der Lee, C., Van Miltenburg, E., and Krahmer, E. (2019). Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. 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 552–562.

Fraile Navarro, D., Coiera, E., Hambly, T. W., Triplett, Z., Asif, N., Susanto, A., Chowdhury, A., Azcoaga Lorenzo, A., Dras, M., and Berkovsky, S. (2025). Expert evaluation of large language models for clinical dialogue summarization. *Scientific Reports*, 15(1):1195.

Fresen, V., Wu-Urbanek, M.-S., and Eger, S. (2024). Reprohum# 0043: Human evaluation reproducing language model as an annotator: Exploring dialogue sum-

marization on ami dataset. In *Proceedings of the Fourth Workshop on Human Evaluation of NLP Systems (HumEval)@ LREC-COLING 2024*, pages 199–209.

Gao, M., Ruan, J., and Wan, X. (2024). Reprohum# 0087-01: A reproduction study of the human evaluation of the coverage of fact checking explanations. In *Proceedings of the Fourth Workshop on Human Evaluation of NLP Systems (HumEval)@ LREC-COLING 2024*, pages 269–273.

Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (2017a). The webnlg challenge: Generating text from rdf data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133.

Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (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.

Gatt, A. and Krahmer, E. (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.

Gayo, I. D. R., Antunes, S., Mendes, A., and Janssen, M. (2016). Towards error annotation in a learner corpus of portuguese. In *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition*, pages 8–17.

Geigle, G., Timofte, R., and Glavaš, G. (2024). Does object grounding really reduce hallucination of large vision-language models? In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2728–2742, Miami, Florida, USA. Association for Computational Linguistics.

George, C. and Stuhlmüller, A. (2023). Factored verification: detecting and reducing hallucination in summaries of academic papers. *arXiv preprint arXiv:2310.10627*.

Gouws, S., Hovy, D., and Metzler, D. (2011). Unsupervised mining of lexical variants from noisy text. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 82–90.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guo, Q., Jin, Z., Qiu, X., Zhang, W., Wipf, D., and Zhang, Z. (2020). Cyclegt: Unsupervised graph-to-text and text-to-graph generation via cycle training. *arXiv preprint arXiv:2006.04702*.

Han, B. and Baldwin, T. (2011). Lexical normalisation of short text messages: Makn sens a# twitter. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 368–378.

Harkous, H., Groves, I., and Saffari, A. (2020a). Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2410–2424, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Harkous, H., Groves, I., and Saffari, A. (2020b). Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2410–2424.

He, J., Peng, B., Liao, Y., Liu, Q., and Xiong, D. (2021). Tgea: An error-annotated dataset and benchmark tasks for textgeneration from pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6012–6025.

Himoro, M. Y. and Pareja-Lora, A. (2020). Towards a spell checker for Zamboanga Chavacano orthography. In *Proceedings of the 12th Language Resources and Eval-*

uation Conference, pages 2685–2697, Marseille, France. European Language Resources Association.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Huang, C., Ghaddar, A., Kobyzev, I., Rezagholizadeh, M., Zaiane, O. R., and Chen, B. (2024). Ottawa: Optimal transport adaptive word aligner for hallucination and omission translation errors detection. *arXiv preprint arXiv:2406.01919*.

Huang, D., Cui, L., Yang, S., Bao, G., Wang, K., Xie, J., and Zhang, Y. (2020). What have we achieved on text summarization? *arXiv preprint arXiv:2010.04529*.

Huidrom, R. and Belz, A. (2022). A survey of recent error annotation schemes for automatically generated text. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 383–398, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Huidrom, R. and Belz, A. (2023). Towards a consensus taxonomy for annotating errors in automatically generated text. In Mitkov, R. and Angelova, G., editors, *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 527–540, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Huidrom, R. and Belz, A. (2025a). Ask me like i'm human: Llm-based evaluation with for-human instructions correlates better with human evaluations than human judges. In *Proceedings of the 4th Workshop on Table Representation Learning*, Vienna, Austria. Association for Computational Linguistics.

Huidrom, R. and Belz, A. (2025b). Using llm judgements for sanity checking results and reproducibility of human evaluations in nlp. In *Proceedings of the 4th Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, Vienna, Austria. Association for Computational Linguistics.

Huidrom, R., Belz, A., and Lorandi, M. (2024). Differences in semantic errors made by different types of data-to-text systems. In Mahamood, S., Minh, N. L., and Ippolito, D., editors, *Proceedings of the 17th International Natural Language Generation Conference*, pages 609–621, Tokyo, Japan. Association for Computational Linguistics.

Huidrom, R., Dušek, O., Kasner, Z., Castro Ferreira, T., and Belz, A. (2022). Two reproductions of a human-assessed comparative evaluation of a semantic error detection system. In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, Waterville, Maine, USA. Association for Computational Linguistics.

Huidrom, R., Lorandi, M., Mille, S., Thomson, C., and Belz, A. (2025). Assessing semantic consistency in Data-to-Text generation: A Meta-Evaluation of textual, semantic and model-based metrics. In *Proceedings of the 18th International Conference on Natural Language Generation: Generation Challenges*, Hanoi, Vietnam. Association for Computational Linguistics.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A., and Fung, P. (2022). Survey of hallucination in natural language generation. *arXiv preprint arXiv:2202.03629*.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Kasner, Z. and Dusek, O. (2022). Neural pipeline for zero-shot data-to-text generation. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 3914–3932, Dublin, Ireland. Association for Computational Linguistics.

Kasner, Z. and Dusek, O. (2024). Beyond traditional benchmarks: Analyzing behaviors of open LLMs on data-to-text generation. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12045–12072, Bangkok, Thailand. Association for Computational Linguistics.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Koponen, M. (2010). Assessing machine translation quality with error analysis. In *Electronic proceeding of the KaTu symposium on translation and interpreting studies*.

Korre, K., Chatzipanagiotou, M., and Pavlopoulos, J. (2021). Elerrant: Automatic grammatical error type classification for greek. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 708–717.

Kryscinski, W., McCann, B., Xiong, C., and Socher, R. (2020). Evaluating the factual consistency of abstractive text summarization. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Laban, P., Schnabel, T., Bennett, P. N., and Hearst, M. A. (2022). SummaC: Revisiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Lango, M., Schmidtová, P., Balloccu, S., and Dušek, O. (2024a). Reprohum# 0043-4: Evaluating summarization models: investigating the impact of education and

language proficiency on reproducibility. In *Proceedings of the Fourth Workshop on Human Evaluation of NLP Systems (HumEval)@ LREC-COLING 2024*, pages 229–237.

Lango, M., Schmidtova, P., Balloccu, S., and Dusek, O. (2024b). ReproHum #0043-4: Evaluating summarization models: investigating the impact of education and language proficiency on reproducibility. In Balloccu, S., Belz, A., Huidrom, R., Reiter, E., Sedoc, J., and Thomson, C., editors, *Proceedings of the Fourth Workshop on Human Evaluation of NLP Systems (HumEval) @ LREC-COLING 2024*, pages 229–237, Torino, Italia. ELRA and ICCL.

Lebret, R., Grangier, D., and Auli, M. (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.

Levenshtein, V. I. et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Li, H., Dong, Q., Chen, J., Su, H., Zhou, Y., Ai, Q., Ye, Z., and Liu, Y. (2024). Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Lin, J., Kovacs, G., Shastry, A., Wuebker, J., and DeNero, J. (2022). Automatic correction of human translations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 494–507.

Liu, J., Shi, Z., and Lipani, A. (2024). Summequal: Summarization evaluation via question answering using large language models. In *Proceedings of the 2nd*

Workshop on Natural Language Reasoning and Structured Explanations (@ ACL 2024), pages 46–55.

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. (2023). G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Loakman, T. and Lin, C. (2024). Reprohum# 0087-01: Human evaluation reproduction report for generating fact checking explanations. In *Proceedings of the Fourth Workshop on Human Evaluation of NLP Systems (HumEval)@ LREC-COLING 2024*, pages 255–260.

Lommel, A., Uszkoreit, H., and Burchardt, A. (2014). Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Revista Tradumàtica: tecnologies de la traducció*, 12:455–463.

Lorandi, M. and Belz, A. (2024a). Dcu-nlg-pbn at the gem'24 data-to-text task: Open-source llm peft-tuning for effective data-to-text generation. In *Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges*, pages 76–83.

Lorandi, M. and Belz, A. (2024b). High-quality data-to-text generation for severely under-resourced languages with out-of-the-box large language models. *arXiv e-prints*, pages arXiv–2402.

Macklovitch, E. (1991). Evaluating commercial mt systems. In *Evaluators' Forum on MT systems, organized by ISSCO at Ste. Croix, Switzerland*.

Mahmud, J., Faisal, F., Arnob, R. I., Anastasopoulos, A., and Moran, K. (2021a). Code to comment translation: A comparative study on model effectiveness & errors. In *Proceedings of the 1st Workshop on Natural Language Processing for Programming (NLP4Prog 2021)*, pages 1–16.

Mahmud, J., Faisal, F., Arnob, R. I., Anastasopoulos, A., and Moran, K. (2021b). Code to comment translation: A comparative study on model effectiveness & errors. In Lachmy, R., Yao, Z., Durrett, G., Gligoric, M., Li, J. J., Mooney, R., Neubig, G., Su, Y., Sun, H., and Tsarfaty, R., editors, *Proceedings of the 1st Workshop on Natural Language Processing for Programming (NLP4Prog 2021)*, pages 1–16, Online. Association for Computational Linguistics.

Martins, D. B. d. J. and Caseli, H. d. M. (2015). Automatic machine translation error identification. *Machine Translation*, 29(1):1–24.

Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Mille, S. and Dasiopoulou, S. (2017). Forge at webnlg 2017. Technical report, Technical report, Dept. of Engineering and Information and Communication ....

Mille, S., Dasiopoulou, S., and Wanner, L. (2019). A portable grammar-based nlg system for verbalization of structured data. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 1054–1056, New York, NY, USA. Association for Computing Machinery.

Mille, S., Sabry, M., and Belz, A. (2024). 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.

Mizrahi, M., Kaplan, G., Malkin, D., Dror, R., Shahaf, D., and Stanovsky, G. (2024). State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Nagata, R., Sato, T., and Takamura, H. (2018). Exploring the influence of spelling errors on lexical variation measures. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2391–2398.

Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., and Bryant, C. (2014). The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14.

Novikova, J., Anderson, C., Blili-Hamelin, B., and Majumdar, S. (2025). Consistency in language models: Current landscape, challenges, and future directions. *arXiv preprint arXiv:2505.00268*.

Osuji, C. C., Huidrom, R., Adebayo, K. J., Ferreira, T. C., and Davis, B. (2024). Dcu-adapt-modpb at the gem'24 data-to-text generation task: Model hybridisation for pipeline data-to-text natural language generation. In *Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges*, pages 66–75.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Pagnoni, A., Balachandran, V., and Tsvetkov, Y. (2021). Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. *arXiv preprint arXiv:2104.13346*.

Paudel, B., Lyzhov, A., Joshi, P., and Anand, P. (2025). Hallucinot: Hallucination detection through context and common knowledge verification. *arXiv preprint arXiv:2504.07069*.

Perri, R. L., Spinelli, D., and Di Russo, F. (2017). Missing the target: the neural processing underlying the omission error. *Brain topography*, 30(3):352–363.

Popović, M. (2020). Informative manual evaluation of machine translation output. In Scott, D., Bel, N., and Zong, C., editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5059–5069, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Popović, M. (2021). On nature and causes of observed mt errors. In *Proceedings of the 18th Biennial Machine Translation Summit (Volume 1: Research Track)*, pages 163–175.

Popović, M. and Burchardt, A. (2011). From human to automatic error classification for machine translation output. In *Proceedings of the 15th Annual conference of the European Association for Machine Translation*.

Popović, M. and Ney, H. (2007). Word error rates: Decomposition over pos classes and applications for error analysis. In *Proceedings of the second workshop on statistical machine translation*, pages 48–55.

Post, M. (2018). A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Puduppully, R. and Lapata, M. (2021). Data-to-text generation with macro planning. *Transactions of the Association for Computational Linguistics*, 9:510–527.

Raj, H., Gupta, V., Rosati, D., and Majumdar, S. (2023). Semantic consistency for assuring reliability of large language models. *arXiv preprint arXiv:2308.09138*.

Rei, R., De Souza, J. G., Alves, D., Zerva, C., Farinha, A. C., Glushkova, T., Lavie, A., Coheur, L., and Martins, A. F. (2022). Comet-22: Unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Reimers, N. and Gurevych, I. (2019a). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Reimers, N. and Gurevych, I. (2019b). 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.

Rivera-Trigueros, I. (2021). Machine translation systems and quality assessment: a systematic review. *Language Resources and Evaluation*, pages 1–27.

Rogers, D. J. and Tanimoto, T. T. (1960). A computer program for classifying plants. *Science*, 132(3434):1115–1118.

Rohrbach, A., Hendricks, L. A., Burns, K., Darrell, T., and Saenko, K. (2018). Object hallucination in image captioning. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.

Scialom, T., Dray, P.-A., Lamprier, S., Piwowarski, B., Staiano, J., Wang, A., and Gallinari, P. (2021). QuestEval: Summarization asks for fact-based evaluation. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Sellam, T., Das, D., and Parikh, A. P. (2020). Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Sharou, K. A. and Specia, L. (2022). A taxonomy and study of critical errors in machine translation. In Moniz, H., Macken, L., Rufener, A., Barrault, L., Costa-jussà, M. R., Declercq, C., Koponen, M., Kemp, E., Pilos, S., Forcada, M. L., Scarton, C., Van den Bogaert, J., Daems, J., Tezcan, A., Vanroy, B., and Fonteyne, M., editors, *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 171–180, Ghent, Belgium. European Association for Machine Translation.

Shimorina, A., Gardent, C., Narayan, S., and Perez-Beltrachini, L. (2018a). WebNLG Challenge: Human Evaluation Results. Technical report, Loria & Inria Grand Est.

Shimorina, A., Gardent, C., Narayan, S., and Perez-Beltrachini, L. (2018b). *WebNLG challenge: Human evaluation results*. PhD thesis, Loria & Inria Grand Est.

Specia, L., Blain, F., Fomicheva, M., Zerva, C., Li, Z., Chaudhary, V., and Martins, A. F. T. (2021). Findings of the WMT 2021 shared task on quality estimation. In Barrault, L., Bojar, O., Bougares, F., Chatterjee, R., Costa-jussa, M. R., Federmann, C., Fishel, M., Fraser, A., Freitag, M., Graham, Y., Grundkiewicz, R., Guzman, P., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Kocmi, T., Martins, A., Morishita, M., and Monz, C., editors, *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Specia, L., Harris, K., Blain, F., Burchardt, A., Macketanz, V., Skadin, I., Negri, M., and Turchi, M. (2017). Translation quality and productivity: A study on rich morphology languages. In *Proceedings of Machine Translation Summit XVI: Research Track*, pages 55–71.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Stymne, S. and Ahrenberg, L. (2012). On the practice of error analysis for machine translation evaluation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1785–1790.

Subramaniam, L. V., Roy, S., Faruquie, T. A., and Negi, S. (2009). A survey of

types of text noise and techniques to handle noisy text. In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, pages 115–122.

Sudalairaj, S., Bhandwaldar, A., Pareja, A., Xu, K., Cox, D. D., and Srivastava, A. (2024). Lab: Large-scale alignment for chatbots. *arXiv preprint arXiv:2403.01081*.

Sui, P., Duede, E., Wu, S., and So, R. (2024). Confabulation: The surprising value of large language model hallucinations. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14274–14284, Bangkok, Thailand. Association for Computational Linguistics.

Tang, L., Goyal, T., Fabbri, A. R., Laban, P., Xu, J., Yahvuz, S., Kryściński, W., Rousseau, J. F., and Durrett, G. (2022a). Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. *arXiv preprint arXiv:2205.12854*.

Tang, X., Fabbri, A., Li, H., Mao, Z., Adams, G. T., Wang, B., Celikyilmaz, A., Mehdad, Y., and Radev, D. (2022b). Investigating crowdsourcing protocols for evaluating the factual consistency of summaries.

Tezcan, A., Hoste, V., and Macken, L. (2017). Scate taxonomy and corpus of machine translation errors. *Trends in E-tools and resources for translators and interpreters*, pages 219–244.

Thomson, C. and Reiter, E. (2020). A gold standard methodology for evaluating accuracy in data-to-text systems. In Davis, B., Graham, Y., Kelleher, J., and Sripada, Y., editors, *Proceedings of the 13th International Conference on Natural Language Generation*, pages 158–168, Dublin, Ireland. Association for Computational Linguistics.

Thomson, C., Reiter, E., and Anya, B. (2024). Common flaws in running human evaluation experiments in nlp. *Computational Linguistics*.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

van der Goot, R., van Noord, R., and van Noord, G. (2018). A taxonomy for in-depth evaluation of normalization for user generated content. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

van Miltenburg, E., Braggaar, A., Braun, N., Damen, D., Goudbeek, M., van der Lee, C., Tomas, F., and Krahmer, E. (2023a). How reproducible is best-worst scaling for human evaluation? a reproduction of 'data-to-text generation with macro planning'. In Belz, A., Popović, M., Reiter, E., Thomson, C., and Sedoc, J., editors, *Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems*, pages 75–88, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

van Miltenburg, E., Braggaar, A., Braun, N., Damen, D., Goudbeek, M., van der Lee, C., Tomas, F., and Krahmer, E. (2023b). How reproducible is best-worst scaling for human evaluation? a reproduction of 'data-to-text generation with macro planning'. *Human Evaluation of NLP Systems*, page 75.

van Miltenburg, E., Clinciu, M., Dušek, O., Gkatzia, D., Inglis, S., Leppänen, L., Mahamood, S., Manning, E., Schoch, S., Thomson, C., and Wen, L. (2021). Underreporting of errors in NLG output, and what to do about it. In Belz, A., Fan, A., Reiter, E., and Sripada, Y., editors, *Proceedings of the 14th International Conference on Natural Language Generation*, pages 140–153, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Vilar, D., Xu, J., Luis Fernando, D., and Ney, H. (2006). Error analysis of statistical machine translation output. In *LREC*, pages 697–702.

Wan, Y., Wu, F., Xu, W., and Srinivasan Sengamedu, S. (2024). Sequence-level cer-

tainty reduces hallucination in knowledge-grounded dialogue generation. *Amazon Science Publication*.

Wang, A., Cho, K., and Lewis, M. (2020). Asking and answering questions to evaluate the factual consistency of summaries. *arXiv preprint arXiv:2004.04228*.

Wang, C., Zhou, W., Ghosh, S., Batmanghelich, K., and Li, W. (2024). Semantic consistency-based uncertainty quantification for factuality in radiology report generation. *arXiv preprint arXiv:2412.04606*.

Wataoka, K., Takahashi, T., and Ri, R. (2024). Self-preference bias in llm-as-a-judge. *arXiv preprint arXiv:2410.21819*.

Weng, R., Yu, H., Wei, X., and Luo, W. (2020). Towards enhancing faithfulness for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2675–2684.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wiseman, S., Shieber, S., and Rush, A. (2017). Challenges in data-to-document generation. In Palmer, M., Hwa, R., and Riedel, S., editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Xia, Y., Liu, X., Yu, T., Kim, S., Rossi, R. A., Rao, A., Mai, T., and Li, S. (2024). Hallucination diversity-aware active learning for text summarization. *arXiv preprint arXiv:2404.01588*.

Yang, A., Yu, B., Li, C., Liu, D., Huang, F., Huang, H., Jiang, J., Tu, J., Zhang, J., Zhou, J., et al. (2025). Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*.

Yang, J., Chen, D., Sun, Y., Li, R., Feng, Z., and Peng, W. (2024). Enhancing semantic consistency of large language models through model editing: An interpretability-oriented approach. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3343–3353, Bangkok, Thailand. Association for Computational Linguistics.

Yu, J., Zhang, X., Xu, Y., Lei, X., Yao, Z., Zhang, J., Hou, L., and Li, J. (2024). A cause-effect look at alleviating hallucination of knowledge-grounded dialogue generation. *arXiv preprint arXiv:2404.03491*.

Zha, Y., Yang, Y., Li, R., and Hu, Z. (2023). Alignscore: Evaluating factual consistency with a unified alignment function. *arXiv preprint arXiv:2305.16739*.

Zhang, D., Gangal, V., Lattimer, B. M., and Yang, Y. (2024a). Enhancing hallucination detection through perturbation-based synthetic data generation in system responses. *arXiv preprint arXiv:2407.05474*.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. (2022). Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhang, S., Wan, D., and Bansal, M. (2023). Extractive is not faithful: An investigation of broad unfaithfulness problems in extractive summarization. In Rogers, A.,

Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2153–2174, Toronto, Canada. Association for Computational Linguistics.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zhang, Y., Mao, S., Ge, T., Wang, X., de Wynter, A., Xia, Y., Wu, W., Song, T., Lan, M., and Wei, F. (2024b). Llm as a mastermind: A survey of strategic reasoning with large language models. *arXiv preprint arXiv:2404.01230*.

Zhong, M., Liu, Y., Yin, D., Mao, Y., Jiao, Y., Liu, P., Zhu, C., Ji, H., and Han, J. (2022). Towards a unified multi-dimensional evaluator for text generation. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zou, D. (2022). Multi-dimensional consideration of cognitive effort in translation and interpreting process studies. In Campbell, J., Larocca, S., Marciano, J., Savenkov, K., and Yanishevsky, A., editors, *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 416–426, Orlando, USA. Association for Machine Translation in the Americas.

Zou, L., Carl, M., Yamada, M., and Mizowaki, T. (2022). Proficiency and external aides: Impact of translation brief and search conditions on post-editing quality. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Workshop 1: Empirical Translation Process Research)*, pages 60–74.

Appendix A

Appendices

A.1 Original Definitions of Content Error Categories from Papers

This section presents definitions, to the extent provided in the original papers, of the content error categories incorporated as the nodes in our minimally merged taxonomy (Figure **??**). In those cases where no definition is provided in the original paper, we list just the name of the error category.

Note we do not include syntactic, discourse-level and other error categories not relating to content/meaning errors, as included in some of the work cited here.

The error categories listed are the lowest(most specific) level of the original error hierarchy in each case.

In one or two cases, the original work additionally provides syntactic labels (e.g. Huang et al.) which we omit if they can apply to any of the error categories (are orthogonal to them).

A.1.1 Top-level error categories

1. *Accuracy* (Zou et al., 2022)

    (a) *Critical.*

(b) *Minor.*

2. *Adequacy* (Popović, 2020)[1]

   (a) *Major.*

   (b) *Minor.*

A.1.2 Omission-type error categories

1. *Omission error* (Costa et al., 2015): "omission errors happen when the translation of a word present in the source text is missing in the resulting translation."

   (a) *Omission error (content words).*

   (b) *Omission error (function words).*

2. *Missing words* (Costa et al., 2012): "when one or more words are missing in the translation."

   (a) *Missing filler words.*

   (b) *Missing content words.*

3. *Omission* (Huang et al., 2020): "Key point is missing from the output."

4. *Missing context* (Mahmud et al., 2021b):

   (a) *Missing Prog. Language Information*: "Missing Attributes that refer to PL specific information."

   (b) *Missing Database Information*: "Missing database attributes that provide needed context to method functionality."

5. *Missing information* (Mahmud et al., 2021b):

   (a) *Missing conditional information*: "Misses code branching information."

   (b) *Missing critical information*: "Comment is missing critical semantic information."

[1]The other error type, Comprehensibility, is not included here, as it is more to do with understanding content that has been correctly included.

(c) *Missing Task Elaboration*: "Did not describe what code was doing properly."

(d) *Missing Non-Critical Information*: "Useful comment but non-critical info missing."

(e) *Missing Web-Related Information*: "Comment failed to mention web-related identifier."

(f) *Failed to Mention Identifiers*: "Does not mention specific variable/attribute names, often using a generic identifier."

(g) *Missing Identifier*: "No identifier mentioned at all."

(h) *Missing Data Structure Information*: "Does not capture relevant data structure info."

(i) *Missing Syntax Information*: "Important syntactic information (e.g. code ordering) is missing."[2]

(j) *Missing Exception*: "Does not mention relevant exception info."

6. *Absent word* (Caseli and Inácio, 2020).

7. *Absent n-gram* (Caseli and Inácio, 2020).

8. *Deletion* (Specia et al., 2021; Al Sharou and Specia, 2022): "critical content that is in the source sentence is not present in the translation."the translation."

   (a) *TOX* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in toxicity (hate, violence or profanity)."

   (b) *SAF* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in health or safety risks."

   (c) *NAM* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in named entities."

   (d) *SEN* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in sentiment polarity or negation."

[2]This refers to programming language syntax, rather than linguistic.

(e) *NUM* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in units/time/date/numbers."

(f) *INS* (Al Sharou and Specia, 2022): "Deviation in instructions."

(g) *OTH* (Al Sharou and Specia, 2022): "Other critical meaning deviation."

9. *Omission* (Specia et al., 2017).

10. *Missing function words* (Specia et al., 2017).

11. *Incorrect Number* (Thomson and Reiter, 2020): "This includes numbers which are spelled out as well as digits."

12. *Incorrect Named Entity* (Thomson and Reiter, 2020): "This includes people, places, organisations, and days of the week."

13. *Other* (Thomson and Reiter, 2020): "Any other type of mistake."

14. *Omissions* (Kasner and Dusek, 2022).

A.1.3 Addition-type error categories

1. *Addition error* (Costa et al., 2015): "the translation of a word that was not present in the source text and was added to the target text."

   (a) *Addition error (content word).*

   (b) *Addition error (function word).*

2. *Extra words* (Costa et al., 2012): "cases where the translation engine generates sentences containing words, most commonly filler words, that should be removed in order to obtain a correct sentence."

3. *Addition* (Huang et al., 2020): "Unnecessary and irrelevant snippets from the source are included in the summary."

4. *Inaccuracy Extrinsic* (Huang et al., 2020): "The summary has content not presented in the source and factually incorrect."

5. *Duplication* (Huang et al., 2020): "A word or longer portion of the text is repeated unnecessarily."

6. *Extraneous/Unnecessary Information Included* (Mahmud et al., 2021b):

   (a) *Unnecessary Data Structure Info*: "Adds unnecessary data structure info to comment."

   (b) *Unnecessary File Information*: "Adds unnecessary file information to comment."

   (c) *Unnecessary Incorrect Information*: "Adds information to comment that is both incorrect and unnecessary."

7. *Extra word* (Caseli and Inácio, 2020).

8. *Extra n-gram* (Caseli and Inácio, 2020).

9. *Addition* Tezcan et al. (2017): "refer[s] to target words not represented in the source."

10. *Omission* Tezcan et al. (2017): "refer[s] to source words not represented in the target text."

11. *Hallucination* (Specia et al., 2021): "critical content that is not in the source is introduced in the translation, for example, profanity words are introduced that were not in the source."

    (a) *TOX* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in toxicity (hate, violence or profanity)."

    (b) *SAF* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in health or safety risks."

    (c) *NAM* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in named entities."

    (d) *SEN* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in sentiment polarity or negation."

(e) *NUM* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in units/time/date/numbers."

(f) *INS* (Al Sharou and Specia, 2022): "Deviation in instructions."

(g) *OTH* (Al Sharou and Specia, 2022): "Other critical meaning deviation."

12. *Addition* (Specia et al., 2017).
13. *Extraneous function words* (Specia et al., 2017).
14. *Incorrect Number* (Thomson and Reiter, 2020): "This includes numbers which are spelled out as well as digits."
15. *Incorrect Named Entity* (Thomson and Reiter, 2020): "This includes people, places, organisations, and days of the week."
16. *Other* (Thomson and Reiter, 2020): "Any other type of mistake."
17. *Hallucinations* (Kasner and Dusek, 2022).
18. *Redundancies* (Kasner and Dusek, 2022).
19. *Extrinsic Noun-Phrase* (Tang et al., 2022a): "A model introduces word(s) not from the source text that function(s) in a summary as subject, object, or prepositional object but cannot be verified from the source."

    (a) *Named Entity* (Tang et al., 2022a).

    (b) *Quantity* (Tang et al., 2022a).

    (c) *Negation* (Tang et al., 2022a).

A.1.4 Substitution-type error categories

1. *Untranslated error* (Costa et al., 2015): "when the engine cannot find any translation candidate for a given source word, [and] cop[ies] it to the translation output 'as is'."
2. *Confusion of senses* (Costa et al., 2015): "is the case of a word that was translated into something representing one of its possible meanings, but, in the given context, the chosen translation is not correct."

3. *Wrong choice* (Costa et al., 2015): "occur when a wrong word, without any apparent relation, is used to translate a given source word."
4. *Collocational errors* (Costa et al., 2015): as wrong choice, but for "blocks of words" rather than single words.
5. *Idiomatic errors* (Costa et al., 2015): "concern errors in idiomatic expressions that the system does not know and translates as regular text."
6. *Lexical Choice* (Costa et al., 2012): "the translation engine chose the wrong translation candidate word."
7. *Disambiguation* (Costa et al., 2012): "the system is not able to disambiguate the correct meaning of a source word in a given context."
8. *Idiomatic Expressions* (Costa et al., 2012): "expressions that should have not been translated literally."
9. *Inaccuracy Intrinsic* (Huang et al., 2020): "Terms or concepts from the source are misrepresented and thus unfaithful."
10. *Positive-Negative Aspect* (Huang et al., 2020): "The output summary represents positive statements whereas the source segment is negative, and vice versa."
11. *Unknown Words* (Huang et al., 2020): "words or expressions [...] for which the translation engine could not find any translation candidate and for that reason were kept in the source language and copied to the translation output.
12. *Incorrect Semantic Information*: (Mahmud et al., 2021b):
    (a) *Partial Incorrect Information*: "Semantically meaningful, with a few errors."
    (b) *Semantically Unrelated to Code*: "Does not capture code context whatsoever."
    (c) *Algorithmically Incorrect*: "Conveys a different algorithmic meaning as compared to the code."

13. *Over-Generalization*: (Mahmud et al., 2021b):

    (a) *Different Meaning*: "Comment over-generalizes on the meaning of the code functionality."

    (b) *Algorithmically Incorrect*: "Overgeneralizes to the point of incorrectness."

    (c) *Missing Attribute Specification*: "Uses generic names such as var."

14. *Not translated word* (Caseli and Inácio, 2020).

15. *Incorrectly translated word* (Caseli and Inácio, 2020).

16. *Not translated n-gram* (Caseli and Inácio, 2020).

17. *Incorrectly translated n-gram* (Caseli and Inácio, 2020).

18. *Reordering* (Caseli and Inácio, 2020).

19. *Reordering errors* (Federico et al., 2014).

20. *Lexicon errors (including wrong lexical choices and extra words)* (Federico et al., 2014).

21. *Missing words* (Federico et al., 2014).

22. *Untranslated* Tezcan et al. (2017): "refer[s] to words that are not translated in the target but are copied instead, when they should have been translated."

23. *Do-not-translate* Tezcan et al. (2017): "refer[s] to source words that have been unnecessarily translated into the target."

24. *Mistranslation* Tezcan et al. (2017).

    (a) *Multi-word expressions*: "The translation is incorrect (and often too literal) because the source sentence contains multi-word expression such as an idiom, a proverb, a collocation, a compound or a phrasal verb."

    (b) *Part of speech*: change in part of speech between source and target text.

    (c) *Word sense disambiguation*: "The target text fragment refers to different (and a wrong) sense of the corresponding source text fragment."

i. *Content Word*.

ii. *Function Word*.

(d) *Partial Translation*: "The incorrect and partial translation of Dutch separable verbs."

(e) *Other*.

25. *Bilingual Terminology* Tezcan et al. (2017).

26. *Source Errors* Tezcan et al. (2017): MT errors that do not originate from the MT system.

27. *Mistranslation* (Specia et al., 2021): "critical content is translated incorrectly into a different meaning, or not translated (i.e. it remains in the source language) or translated into gibberish."

(a) *TOX* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in toxicity (hate, violence or profanity)."

(b) *SAF* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in health or safety risks."

(c) *NAM* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in named entities."

(d) *SEN* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in sentiment polarity or negation."

(e) *NUM* (Specia et al., 2021; Al Sharou and Specia, 2022): "Deviation in units/time/date/numbers."

(f) *INS* (Al Sharou and Specia, 2022): "Deviation in instructions."

(g) *OTH* (Al Sharou and Specia, 2022): "Other critical meaning deviation."

28. *Mistranslation* (Specia et al., 2017).

29. *Untranslated* (Specia et al., 2017).

30. *Incorrect function words* (Specia et al., 2017).

31. *Unintelligible* (Specia et al., 2017).

32. *Not Checkable* (Thomson and Reiter, 2020): "A statement which can not be checked; either the information is not available or it is too time-consuming to check."
33. *Incorrect Number* (Thomson and Reiter, 2020): "This includes numbers which are spelled out as well as digits."
34. *Incorrect Named Entity* (Thomson and Reiter, 2020): "This includes people, places, organisations, and days of the week."
35. *Incorrect word* (Thomson and Reiter, 2020): "A word which is not [a number or noun phrase] and is incorrect."
36. *Other* (Thomson and Reiter, 2020): "Any other type of mistake."
37. *Incorrect fact merging* (Kasner and Dusek, 2022).
38. *Intrinsic Noun-Phrase* (Tang et al., 2022a): "A model misrepresents word(s) from the source text that function(s) in a summary as subject, object, or prepositional object."
    (a) *Named Entity* (Tang et al., 2022a).
    (b) *Quantity* (Tang et al., 2022a).
    (c) *Negation* (Tang et al., 2022a).

A.2 Systematic Survey of Error Annotation Schemes and Error Taxonomies

A.2.1 Error Annotation Schemes for Machine Generated Text (MGT)

In this section, we present entire taxonomies and examples of individual errors for error annotation schemes developed for HGT and extracted from the 22 surveyed papers briefly summarised in Section **??**.

Costa et al. (2015) present the taxonomy found in Figure A.1. Below are some error examples extracted from the paper:

Example 1: Spelling error in Orthography level

Example: Spelling error

EN: Basilica of the Martyrs

EP: Basílica dos *Mátires

Correct translation: Basílica dos Mártires

Example 2: Omission error (content word) in Lexis level

Example: Omission error (content word)

EN: In his inaugural address, Barack Obama

EP: No seu * inaugural, Barack Obama

Correct translation: No seu discurso inaugural, Barack Obama

Example 3: Addition error (content word) in Lexis level

Example: Addition error (content word)

EN: This time I'm not going to miss

EP: Desta vez *correr não vou perder

Correct translation: Desta vez não vou perder

Al Sharou and Specia (2022) define seven main categories as mentioned in Section **??**. Here are some error examples extracted from the paper:

Example 1: Deviation in toxicity (TOX)

ST: Your killing the fucking planet.

MT-ed text: May the damn planet kill you.

Translation into Arabic by Systran

Example 2: Deviation in health/safety risks (SAF)

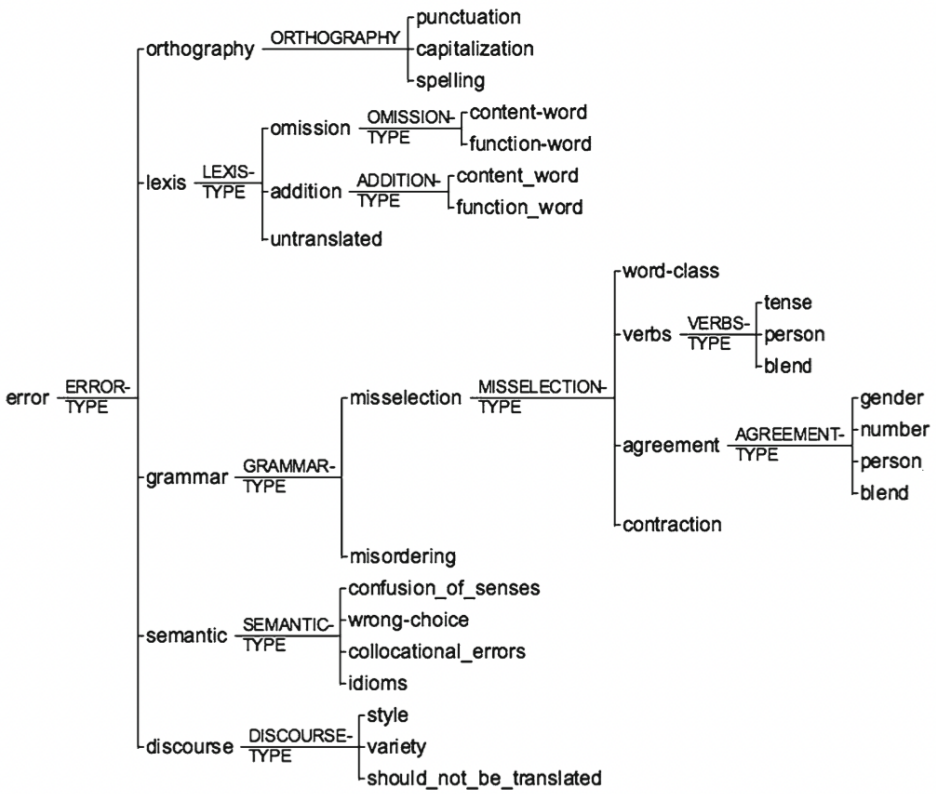

Figure A.1: Figure of taxonomy extracted from (Costa et al., 2015).

ST: I Know two teenagers that suffer from gerd it is a big problem for these people!

MT-ed text: I Know two teenagers that suffer from root disease it is a big problem for these people!

Translation into Chinese by GT.

Example 3: Deviation in named entities (NAM)

ST: Your fucking ass doesn't know shit about it AT ALL.Rocky.

MT-ed text: Your fucking ass doesn't know shit about it AT ALL.rock.

Translation into Italian by Bing

Caseli and Inácio (2020) presents a taxonomy found below:

1. Syntactic errors

    - Number agreement

- Gender agreement
- Verb inflection
- PoS

—item Lexical errors

- Extra word
- Absent word
- Not translated word
- Incorrectly translated word

2. N-gram

- Absent n-gram
- Not translated n-gram
- Incorrectly translated n-gram

3. Reordering

- Order

He et al. (2021) present a taxonomy found in Figure A.2.

Belkebir and Habash (2021) present a taxonomy found in Table Figure A.3.

Huang et al. (2020) show PolyTope with each error types on a three-coordinates for syntactic and semantic roles which is found in Figure A.4 and some examples extracted from the paper are found below.

Example 1: Inaccuracy Intrinsic

> "Pittsburgh Union Station is 10 kilometers from Exhibition Center and 3 kilometers from the University of Pittsburgh" in the source but "Pittsburgh Union Station is 3 kilometers from Exhibition Center" in the output.

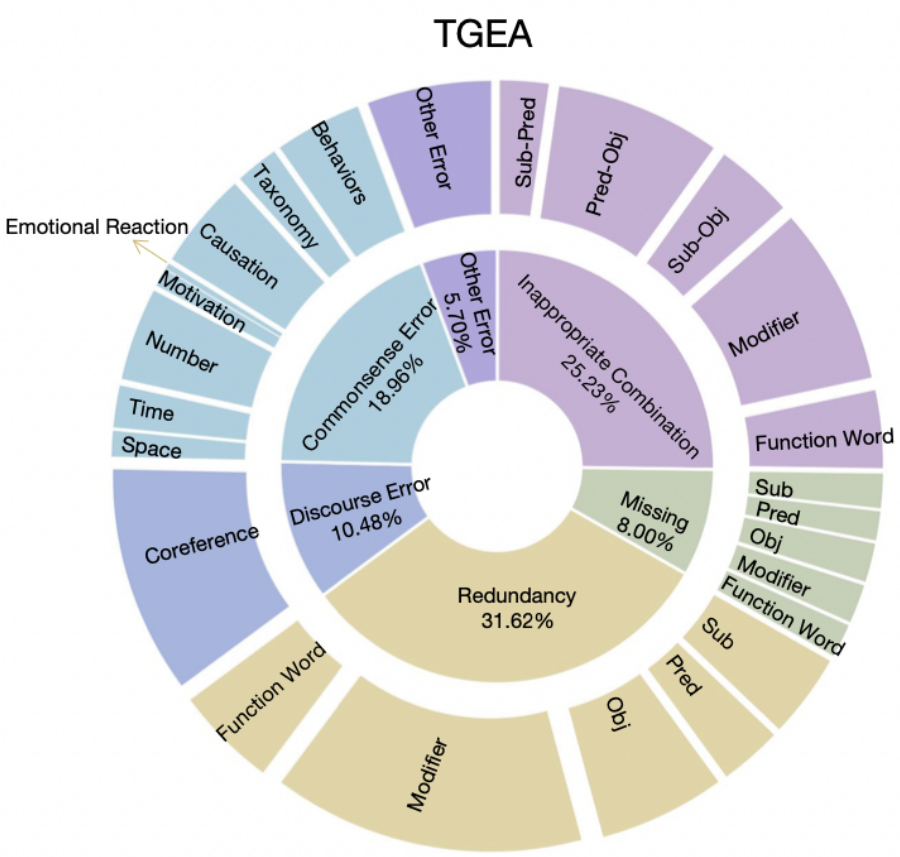

Figure A.2: Figure extracted from (He et al., 2021) of level-1 and level-2 error types in TGEA which is an error annotated dataset.

Example 2: Inaccuracy Extrinsic

> it is described as “Pittsburgh Union Station, also known as Pittsburgh South Station” in the output but “Pittsburgh South Station” is neither mentioned in the source text nor exists in the real world.

Example 3: Positive-Negative Aspect

> “push a button” summarized as “don’t push a button”, “non-slip” summarized as “slip”. This category applies only to actions and modifiers and refers to omitted or added negative particles.

Di et al. (2019) presents an error taxonomy found below.

1. Target errors: Target word errors are ’due to errors in the Wiktionary source data and incorrect extraction of paradigm tables.’

Class	Err Tag	Description	Arabic Example	Transliteration
Orthography	**OA**	Confusion in Alif, Ya and Alif-Maqsura	علي ← على	ϛly → ϛlý
	OC	Wrong order of word characters	تبرينا ← تربينا	tbrynA → trbynA
	OD	Additional character(s)	يعدوم ← يدوم	yϛdwm → ydwm
	OG	Lengthening short vowels	نقيمو ← نقيم	nqymw → nqym
	OH	Hamza errors	اكثر ← أكثر	Akθr → Âkθr
	OM	Missing character(s)	سالين ← سائلين	sAlyn → sAŷlyn
	ON	Confusion between Nun and Tanwin	ثوبن ← ثوبٌ	θwbn → θwbũ
	OR	Replacement in word character(s)	مصلنا ← وصلنا	mSlnA → wSlnA
	OS	Shortening long vowels	أوقت ← أوقات	Âwqt → ÂwqAt
	OT	Confusion in Ha, Ta and Ta-Marbuta	مشاركه ← مشاركة	mšArkh → mšArkẖ
	OW	Confusion in Alif Fariqa	وكانو ← وكانوا	wkAnw→ wkAnwA
	OO	*Other orthographic errors*	-	-
Morphology	**MI**	Word inflection	معروف ← عارف	mϛrwf → ϛArf
	MT	Verb tense	تفرحني ← أفرحتني	tfrHny → ÂfrHtny
	MO	*Other morphological errors*	-	-
Syntax	**XC**	Case	رائع ← رائعاً	rAŷϛ → rAŷϛAã
	XF	Definiteness	المن ← سن	Alsn → sn
	XG	Gender	الغربي ← الغربية	Alγrby → Alγrbyẖ
	XM	Missing word	على ← Null	*Null* → ϛlý
	XN	Number	فكرتي ← أفكاري	fkrty → ÂfkAry
	XT	Unnecessary word	Null← على	ϛlý → *Null*
	XO	*Other syntactic errors*	-	-
Semantics	**SF**	Conjunction error	سبحان ← فسبحان	sbHAn → fsbHAn
	SW	Word selection error	من ← عن	mn → ϛn
	SO	*Other semantic errors*	-	-
Punctuation	**PC**	Punctuation confusion	المتوسط ← المتوسط،	AlmtwsT. → AlmtwsT,
	PM	Missing punctuation	العظيم ← العظيم،	AlϛĎym → AlϛĎym,
	PT	Unnecessary punctuation	العام، ← العام	AlϛAm, → AlϛAm
	PO	*Other errors in punctuation*	-	-
Merge	**MG**	Words are merged	ذهبتالبارحة ← ذهبت البارحة	ðhbtAlbArHẖ → ðhbt AlbArHẖ
Split	**SP**	Words are split	المحا دثات ← المحادثات	AlmHA dθAt → AlmHAdθAt

Figure A.3: Figure of ARETA which is an error annotation system extracted from (Belkebir and Habash, 2021).

2. Prediction errors

   - nonce-word errors: Nonce word errors are 'due to illegal words, i.e. situations when the string generated by a system does not exist in Tibetan.'
   - allomorphy errors: Allomorphy errors are considered for verb inflection and diacritics for Tibetan language.

Mahmud et al. (2021a) present a taxonomy found in Figure A.5.

Costa et al. (2012) present a taxonomy and examples below:

The error taxonomy is as follows:

1. Missing words (when one or more words are missing in the translation)

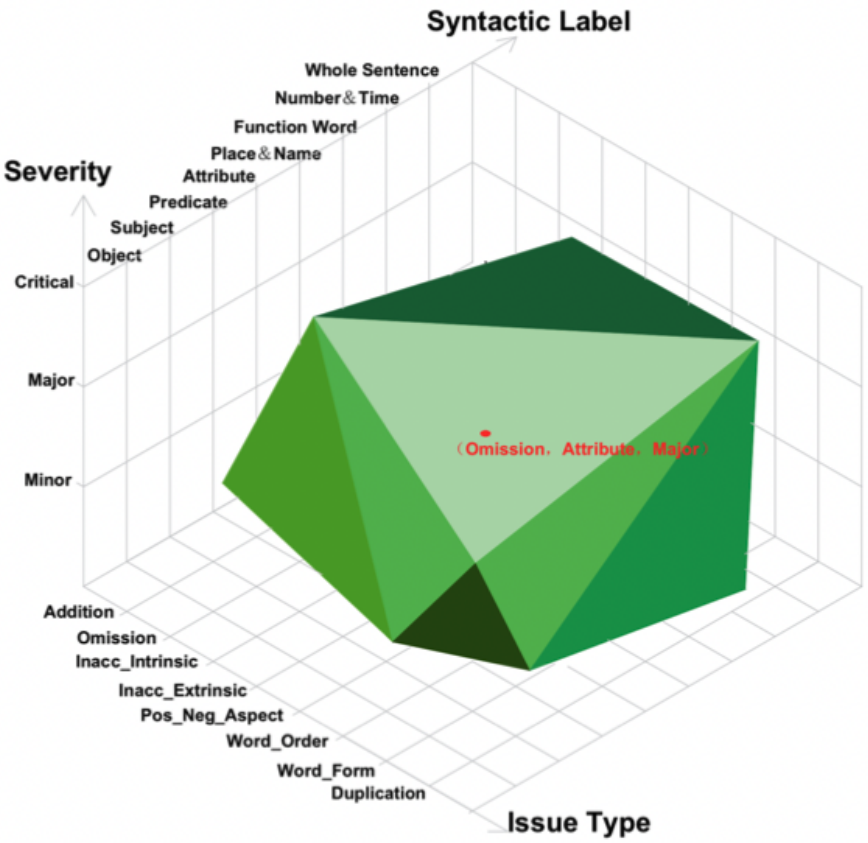

Figure A.4: Figure of PolyTope with each error types on a three-coordinates for syntactic and semantic roles extracted from (Huang et al., 2020).

- Missing word fillers
- Missing content words

2. Word order (when reordering model is unable to produce a reordering of the sentence)

3. Incorrect words (when a translation engine is unable to produce a correct translation of a word or expression)

   - Lexical choice
   - Disambiguation
   - Incorrect form
   - Extra words
   - Idiomatic Expressions

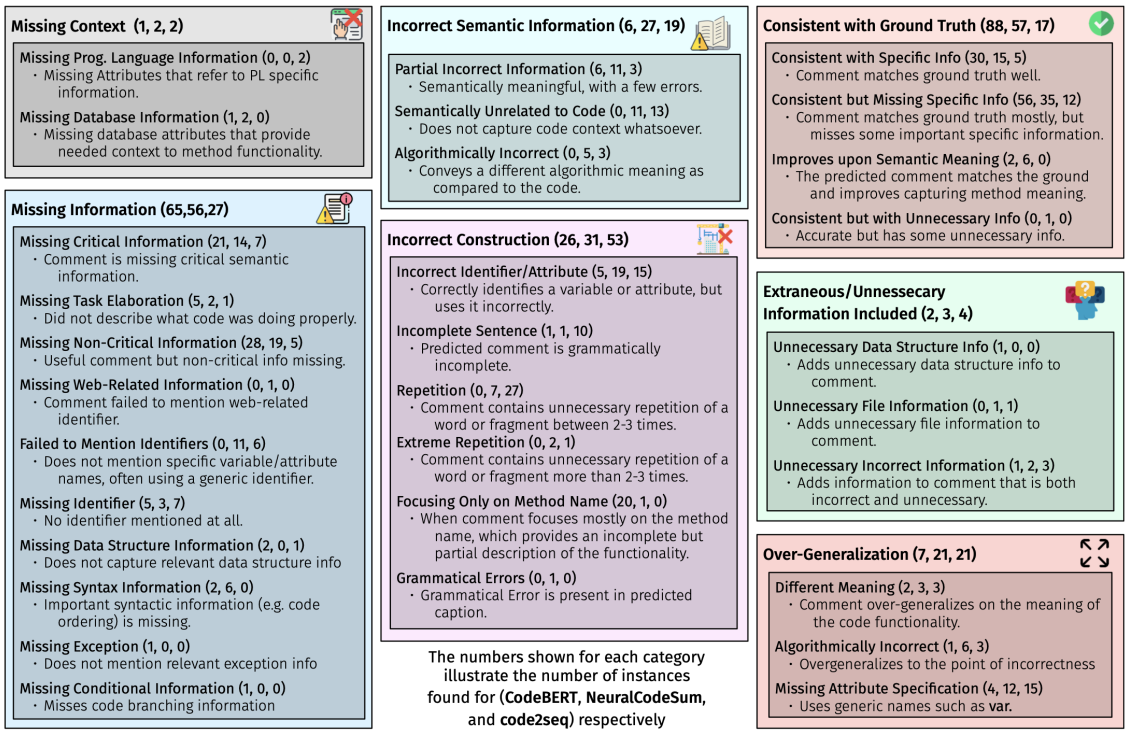

Figure A.5: Figure of taxonomy of errors between the generated summaries and ground truth extracted from (Mahmud et al., 2021a).

4. Unknown words (when the translation engine could not find the translation in the target language and keeps the words or expressions in the source language).

Example extracted from the paper in Figure A.6:

Original: *Who was the first American to walk in space?*
Translation: *Quem foi o primeiro a andar* ***americano*** *no espaço?*
Correct Translation: *Quem foi o primeiro* ***americano*** *a andar no espaço?*

Figure A.6: Examples of word order error extracted from (Costa et al., 2012)

Macklovitch (1991) presents an error tabulation for its taxonomy in Figure A.7.

I.	Morphology, graphology & layout
I.1.	Number, inflection & agreement
I.2	Upper/lower case
I.3	Hyphens, slashes & quotes
I.4	Layout; underlining
I.5	References; place names
I.6	Unknown words/symbols
II.	Analysis
II.1	Categorial homography
II.2	'-ing forms
II.3	'-ed forms & passives
II.4	Coordinate structures
II.5	Stacked nominals
II.6	Anaphora & ellipsis
II.7	Articles
II.8	Gibberish
III.	Transfer and generation
III.1	Incorrect/incomplete TL equiv.
III.2	Polysemy
III.3	Restructuring
III.4	Inappropriate/incorrect form generated
III.5	Stylistic changes

Figure A.7: Figure of taxonomy extracted from (Macklovitch, 1991)

A.2.2 Error Annotation Schemes for Human Generated Text (HGT)

In this section, we present entire taxonomies and examples of individual errors for error annotation schemes developed for HGT and extracted from the 22 surveyed papers briefly summarised in Section 4.3.

Lin et al. (2022) present a taxonomy based on error correction and each edit belongs to one of the three types listed below. Figure A.8 show error types and examples extracted from the paper.

The error taxonomy in Lin et al. (2022) is

1. Monolingual edits (identifiable from the target side of the text)
    - typos (spelling, punctuation, spacing and orthographic issues)
    - grammar
    - fluency
2. Bilingual edits (mismatch between source and target text) Eg, under-translation, mis-translation.
3. Preferential edits (based on the preference of the customer) Eg, terminology, style preference.

van der Goot et al. (2018) present a taxonomy; below are a few examples of errors.

Example 1: Typographical error

spirite—→spirit, complaing—→complaining, throwg—→throw

Example 2: Repetition

soooo—→so, weiiiiird—→weird

Example 3: Unknown

Error Type	Example Text
Monolingual: typos	s: Do your feet roll inwards when running? t: KIppen deine Füße beim Laufen nach innen? t': Kippen deine Füße beim Laufen nach innen?
Monolingual: grammar	s: Own tough winter runs in the ... t: Bei harten Winterläufe sorgt der ... t': Bei harten Winterläufen sorgt der ...
Monolingual: fluency	s: The traffic emerges from the VPN server and ... t: Der Verkehr wird vom VPN-Server ausgegeben und ... t': Der Datenverkehr wird vom VPN-Server ausgegeben und ...
Bilingual	s: Quad Core XEON E3-1501M, 2.9GHz t: Quad Core XEON 2,9 GHz t': Quad Core XEON E3-1501M, 2,9 GHz
Preferential	s: VersaMax I / O auxiliary spring clamp style t: VersaMax Zusatz-E / A Federklemmenart t': VersaMax Zusatz-E / A Federklemmenbauform

Figure A.8: Figure of error taxonomy for ACED corpus with examples extracted from (Lin et al., 2022).

Category	Examples
1. Typographical error	spirite↦spirit, complaing↦complaining, throwg↦throw
2. Missing apostrophe	im↦i'm, yall↦y'all, microsofts↦microsoft's
3. Spelling error	favourite↦favorite, dieing↦dying, theirselves↦themselves
4. Split	pre order↦preorder, screen shot↦screenshot
5. Merge	alot↦a lot, nomore↦no more, appstore↦app store
6. Phrasal abbreviation	lol↦laughing out loud, pmsl↦pissing myself laughing
7. Repetition	soooo↦so, weiiiiird↦weird
8. Shortening vowels	pls↦please, wrked↦worked, rmx↦remix
9. Shortening end	gon↦gonna, congrats↦congratulations, g↦girl
10. Shortening other	cause↦because, smth↦something, tl↦timeline,
11. Phonetic transformation	hackd↦hacked, gentille↦gentle, rizky↦risky
12. Regular transformation	foolin↦fooling, wateva↦whatever, droppin↦dropping
13. Slang	cuz↦because, fina↦going to, plz↦please
14. Unknown	skepta↦sunglasses, putos↦photos

Figure A.9: Figure of taxonomy extracted from (van der Goot et al., 2018).

skepta—→sunglasses, putos—→photos

Ng et al. (2014) present an error annotation scheme and a few examples below.

Figure A.10 includes extracted error categories and its examples from the paper. Here are some examples extracted from the paper:

Example 1: Verb tense (Vt)

Medical technology during that time [is → was] not advanced enough to cure him.

Example 2: Word form (Wform)

The sense of [guilty → guilt] can be more than expected.

Example 3: Unclear meaning (Um)

Genetic disease has a close relationship with the born gene. (i.e., no correction possible without further clarification.)

Dickinson and Herring (2008) presents a taxonomy for Russian verbal morphology:

1. Inappropriate verb stem
    - Always inappropriate
    - Inappropriate for this context
2. Inappropriate verb affix
    - Always inappropriate
    - Always inappropriate for verbs
    - Inappropriate for this verb
3. Inappropriate combination of stem and affix
4. Well-formed word in inappropriate context
    - Inappropriate agreement features
    - Inappropriate verb form (tense, perfective/imperfective, etc.)

Dickinson (2010) present an error taxonomy in Figure A.11.

Nagata et al. (2018) present a spelling error annotation scheme in Figure A.12.

Barbagli et al. (2016) present error annotations and examples in Figure A.13.

Himoro and Pareja-Lora (2020) present an error taxonomy and examples in Figure A.14.

Type	Description	Example
Vt	Verb tense	Medical technology during that time [**is** → was] not advanced enough to cure him.
Vm	Verb modal	Although the problem [**would** → may] not be serious, people [**would** → might] still be afraid.
V0	Missing verb	However, there are also a great number of people [**who** → who are] against this technology.
Vform	Verb form	A study in 2010 [**shown** → showed] that patients recover faster when surrounded by family members.
SVA	Subject-verb agreement	The benefits of disclosing genetic risk information [**outweighs** → outweigh] the costs.
ArtOrDet	Article or determiner	It is obvious to see that [**internet** → the internet] saves people time and also connects people globally.
Nn	Noun number	A carrier may consider not having any [**child** → children] after getting married.
Npos	Noun possessive	Someone should tell the [**carriers** → carrier's] relatives about the genetic problem.
Pform	Pronoun form	A couple should run a few tests to see if [**their** → they] have any genetic diseases beforehand.
Pref	Pronoun reference	It is everyone's duty to ensure that [**he or she** → they] undergo regular health checks.
Prep	Preposition	This essay will [**discuss about** → discuss] whether a carrier should tell his relatives or not.
Wci	Wrong collocation/idiom	Early examination is [**healthy** → advisable] and will cast away unwanted doubts.
Wa	Acronyms	After [**WOWII** → World War II], the population of China decreased rapidly.
Wform	Word form	The sense of [**guilty** → guilt] can be more than expected.
Wtone	Tone (formal/informal)	[**It's** → It is] our family and relatives that bring us up.
Srun	Run-on sentences, comma splices	The issue is highly [**debatable, a** → debatable. A] genetic risk could come from either side of the family.
Smod	Dangling modifiers	[**Undeniable,** → It is undeniable that] it becomes addictive when we spend more time socializing virtually.
Spar	Parallelism	We must pay attention to this information and [**assisting** → assist] those who are at risk.
Sfrag	Sentence fragment	**However, from the ethical point of view.**
Ssub	Subordinate clause	This is an issue [**needs** → that needs] to be addressed.
WOinc	Incorrect word order	[**Someone having what kind of disease** → What kind of disease someone has] is a matter of their own privacy.
WOadv	Incorrect adjective/ adverb order	In conclusion, [**personally I** → I personally] feel that it is important to tell one's family members.
Trans	Linking words/phrases	It is sometimes hard to find [**out** → out if] one has this disease.
Mec	Spelling, punctuation, capitalization, etc.	This knowledge [**maybe relavant** → may be relevant] to them.
Rloc−	Redundancy	It is up to the [**patient's own choice** → patient] to disclose information.
Cit	Citation	Poor citation practice.
Others	Other errors	An error that does not fit into any other category but can still be corrected.
Um	Unclear meaning	Genetic disease has a close relationship with the **born gene.** (i.e., no correction possible without further clarification.)

Figure A.10: Figure of taxonomy extracted from (Ng et al., 2014).

1. Abbreviation (ABR) in Intentional errors are due to omission of letters or use of homophones letters and/or numbers to replace syllables. Example, kme-¿kame.

2. Omission (OMS) in unintentional errors are due to deletion of letter from a word without an explanation. Example, Chaacano -¿ Chavacano.

Caines et al. (2020) present the error types determined by grammatical error correction of texts in TSCC in Figure A.15.

0. Correct: The word is well-formed.
1. Stem errors:
   (a) Stem spelling error
   (b) Semantic error
2. Suffix errors:
   (a) Suffix spelling error
   (b) Lexicon error:
      i. Derivation error: The wrong POS is used (e.g., a noun as a verb).
      ii. Inherency error: The ending is for a different subclass (e.g., inanimate as an animate noun).
   (c) Paradigm error: The ending is from the wrong paradigm.
3. Formation errors: The stem does not follow appropriate spelling/sound change rules.
4. Syntactic errors: The form is correct, but used in an in appropriate syntactic context (e.g., nominative case in a dative context)

- Lexicon incompleteness: The form may be possible, but is not attested.

Figure A.11: Figure of taxonomy extracted from (Dickinson, 2010)

Korre et al. (2021) present an error annotation scheme in Figure A.16.

A.3 Differences in Semantic Errors Made by Different Types of Data-to-text Systems

A.3.1 Participants Recruitment Email

The following is our email template that we sent to recruit participants for our experiment.

Subject: Participants needed for data-to-text system evaluation (1 Hour, 25 Euros)

Error code	Explanation	Treatment
SP	Spelling that does not exist in English. e.g., I am a *sistem* engineer.	✓
PC	Inappropriate plural form conjugation. e.g., I didn't do *anythings*.	✓
OC	Over-regularized morphology. e.g., I *gived* her her hat.	✓
GC	Conjugation error other than the above two. e.g., I am *driveing*.	✓
NM	Spelling error in names. e.g., I went to *Desneyland*.	✓
RE	Real word spelling error (i.e., context sensitive error). e.g., *Their* is a house.	×
RO	Romanized Japanese. e.g., I ate an *omusubi*.	-
SR	Romanized Japanese that has no equivalent English expression. e.g., I went to *Hukuoka*. or that becomes proper English if transliterated. e.g., I ate *susi*.(susi → sushi).	-
CW	Coined word that is not used in English. e.g., I want to be a *nailist*.	-
FW	Foreign words other than Japanese. e.g., I have an *Arbeit*.	-
AL	Non-American (e.g. British English) spelling. e.g., It's my *favourite*.	-
AB	Improper abbreviation that is not used in English. e.g., I went to *USJ*.	-
O	Other than the above.	-

Figure A.12: Figure of spelling error and corresponding treatment extracted from (Nagata et al., 2018).

Dear all,

I hope this email finds you well. My name is [FirstName LastName], and I am currently working on a project focusing on the human evaluation of data-to-text system outputs as a part of my PhD thesis. I am reaching out to you to invite you to participate in this exciting research opportunity.

The aim of this project is to evaluate semantic errors (addition, omission, substitution, repetition) in the input (RDF triples) and data-to-text system outputs pairs from WebNLG 2020 Shared Task. We would need evaluations to be completed no

Class of Error	Type of Modification	Example
Verbs	Use of tense	[...] dopo aver fatto le squadre <M t="11" c="abbiamo">avevamo</M> subito iniziato a giocare
	Use of mood	[...] il pensiero che mi tormentava di più era che tra poco si <M t="12" c="sarebbe fatto">faceva</M> il campo scuola.
	Subject-Verb agreement	[...] la mia famiglia ed io <M t="13" c="stavamo">stavo</M> al mare a Torvajanica
Prepositions	Erroneous use	<M t="14" c="in">a</M> Romania sono andata <M t="14" c="in">a</M> agosto
Pronouns	Erroneous use	Proteggere i più deboli è molto coraggioso da parte di chi <M t="16" c="li">lo</M> protegge
	Redundancy	Alla nostra maestra <M t="18" c="canc">gli</M> piaceva tanto la storia
	Erroneous use of relative pronoun	La scienza non so perché mi fa pensare a un fenomeno costruito su un'altura <M t="19" c="per cui">che</M> ci vuole molto ingegno.
Articles	Erroneous use	<M t="111" c="gli">i</M> dei, sapendo che qualcuno aveva preso senza merito il sacro vaso della Giustizia, si rattristarono molto, [...]
Use of *h*	Omission	<M t="23" c="ho">o</M> visto uno spettacolo bellissimo con i raggi laser
Lexicon	Erroneous use	C'era molta ombra nel giardino e io mi ci <M t="31" c="addormentavo">addormivo</M> sempre.

Figure A.13: Figure of error annotations and examples extracted from (Barbagli et al., 2016)

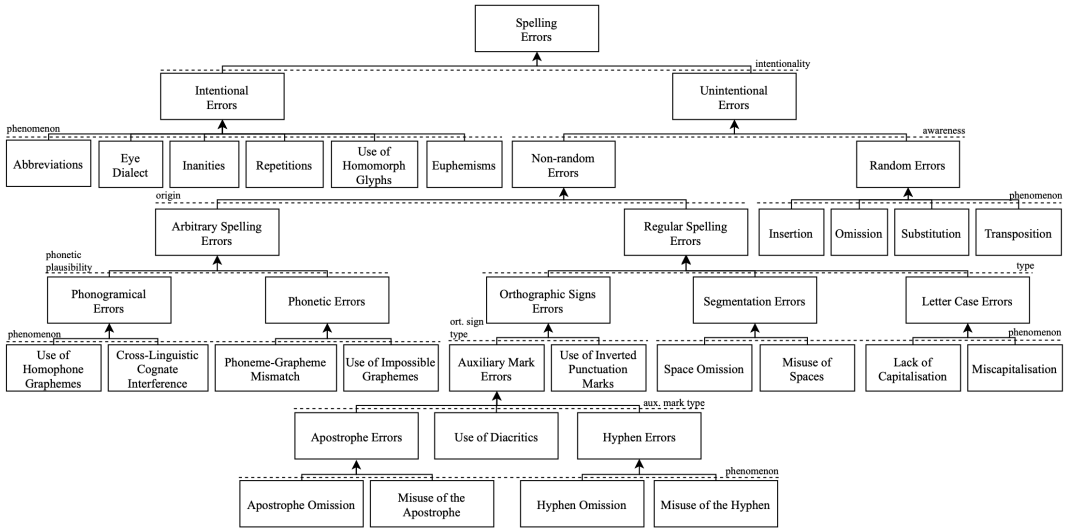

Figure A.14: Figure of spelling error taxonomy for ZC extracted from (Himoro and Pareja-Lora, 2020)

later than [DD MM YY].

Our pilot experiment showed that the evaluation should take about an hour and we are offering 25 Euro for this task.

Prior to the evaluation process, there will be a training session to familiarise the participants with the annotation tool we will be using and of course, provide clear guidelines on how to evaluate these system outputs. We will ensure that the par-

Edit type	Missing
	Replacement
	Unnecessary
Error type	Adjective
	Adjective:form
	Adverb
	Conjunction
	Contraction
	Determiner
	Morphology
	Noun
	Noun:inflection
	Noun:number
	Noun:possessive
	Orthography
	Other
	Particle
	Preposition
	Pronoun
	Punctuation
	Spelling
	Verb
	Verb:form
	Verb:inflection
	Verb:subj-verb-agr
	Verb:tense
	Word order

Figure A.15: Figure of the error types determined by grammatical error correction of texts in TSCC extracted from (Caines et al., 2020)

ticipants have all the necessary resources and support to carry out the evaluation effectively. To acknowledge the time and effort, we are offering compensation for your participation.

We believe that this research project makes a significant contribution to the scientific work in the field.

Error Type	Meaning	Description	Example
AD:FORM	Adverb Form	Errors concerning the form an adverb.	καλός → καλώς
ADJ:FORM*	Adjective Form	Errors concerning the form of an adjective	καλός → καλύτερος
NOUN:FORM	Noun Form	Errors concerning the number,the case or the suffix of a noun.	του νους → του νου
PRON:FORM	Pronoun Form	Errors concerning the number, the case or the suffix of a pronoun.	κάποια → κάποιας
VERB:FORM	Verb Form	Errors concerning the disposition, the voice, the inflection, the tense,the number or the person of a verb.	(εσείς) πηγαίνεται → (εσείς) πηγαίνετε
CONJ	Conjunction	Errors concerning conjunctions.	και → αλλά
PREP	Preposition	Errors concerning prepositions.	από → σε
DET*	Determiner	Errors concerning articles or determiners.	το → του τον → έναν
SPELL	Spelling	Spelling errors.	ευχέρια → ευχέρεια
FN	Final -ν/nu	Final -ν/nu addition or removal.	την → τη / μη → μην
PUNCT	Punctuation	Errors concerning the punctuation.	. → ;
OTHER	Other Errors	An error that does not fit into any other category but can still be corrected.	καμία → για κανένα
ACC	Accentuation	Accentuation addition or removal.	καθηκοντα → καθήκοντα
UNK	Unknown error type	An error that can be detected but not corrected.	usually long error spans
WO	Words Order	Error in words order.	όταν φεύγω έρθεις → όταν έρθεις φεύγω
ORTH*	Orthography	Spacing Errors	γιασένα → για σένα
PART:FORM	Participle Form	Errors concerning the number,the case or the person of a participle.	(πήγε) τρεχόμενος → (πήγε) τρέχοντας
VERB:SVA	Subject Verb Agreement	The subject and the verb to be in person agreement.	(εγώ θα) φύγει → (εγώ θα) φύγω

Figure A.16: Figure of ELERRANT and human error type annotation guide extracted from (Korre et al., 2021). The error types with (*) do not exist for human annotation scheme and the last two error types do not exist in the ELERRANT annotation scheme.

If you are interested in being a part of this research project and contributing to the field, please express your interest by filling out this Google Form.

Thank you for considering this opportunity. Your participation is highly valued, and I look forward to the possibility of working together on this important research project.

Best regards,

[Signature]

A.3.2 Pilot Participants Feedback

Nine out of 10 evaluators filled in the reflection form. The other evaluator gave their feedback via text communication. In this section, we report the feedback as received

from the reflection form. We summarise them below:

- Five out of nine evaluators expressed the overall pilot experiment was *neither easy nor difficult*, two of them marked as *easy* and the other two marked it as *difficult*.

- Six out of nine evaluators found the annotation guidelines *easy* to follow, two of them marked it as *neither easy nor difficult* and one of them found to be *difficult*.

- On an average, it took about 20 minutes for the evaluators to understand the annotation guidelines.

- On an average, it took about 25 minutes for the evaluators to complete the annotation task.

- All evaluators confirmed that they read the annotation guidelines before starting the annotations.

- Six out of nine evaluators found the brat annotation *difficult* to use. Meanwhile, three of them found it *easy*.

- Seven out of nine evaluators expressed their need on more training for using the brat annotation tool (apart from Section 3 "Instructions for using the brat annotation tool" in the instructions document) whereas two of them answered a no.

- Seven out of nine evaluators found the error type's definitions and examples easy to follow in the instructions document *easy* to follow. Meanwhile, two of them found it *difficult*.

- Eight out of nine participants expressed their interest in the main study, one of them expressed as a maybe.

A.3.3 Annotation Steps

We asked annotators to follow the following annotation steps, as part of the annotation guidelines:

1. In the first step, the evaluator should examine whether each element in the input triples is verbalised or not. If an element is not expressed in the verbalisation, mark the element as an omission error type in the triple.

   If the whole triple is not expressed in the verbalisation, mark each element as an omission error type in the triple. For example, if the triple 'ENAIRE — city — Madrid' is not expressed in the verbalisation, then mark 'ENAIRE' as an omission, 'city' as an omission and 'Madrid' as an omission.

   If each element in the input triples is verbalised which means there is no omission error, then proceed to the second step.
2. In the second step, the evaluator should examine whether all the content words and phrases in the verbalisation render a corresponding element(s) in the triples.

   If a content word phrase does not render a corresponding element in the input triples, mark it as an addition error type.

   If all the content phrases in the verbalisation render a corresponding element in the input triples this means there is no addition error, so proceed to the third step.
3. In the third step, the evaluator should check if any part of the output is repeated, including close paraphrases. This is the case e.g. if an element in the triples is rendered more than once. If there is a content phrase that is repeated in this sense, mark it as a repetition error type.

   If all the content phrases in verbalisation include all the elements in the triples without an extra in the verbalisation that has no relation in the input triples, which means there is no repetition error, then proceed to the next pair of triple(s) and verbalisation.

A.3.4 Additional Notes Given to Evaluators

We provide the following notes below to the evaluators along with the annotation guidelines. More details in Appendix A.3.5.

- If there is more than one triple in the input, triples are enclosed within single quotes (‘ ’) and separated by commas. For example, ‘Joe_Biden — president — United_States’, ‘Joe_Biden — birthPlace — Pennsylvania’.

- The evaluator should be careful while selecting the word span when marking an error. The evaluator should select complete tokens, i.e., words in the text, that are delimited by whitespace.

  For example, the selection for ‘president’ in ‘Joe Biden is the president of the United States.’ is correct, but selecting just ‘pres’ is not correct, as in ‘Joe Biden is the president of the United States.’ Similarly, ‘Joe_Biden — president — United_States’ is correct, but ‘Joe_Biden — president — United_States’ is not.

- The evaluator should consider the inferred verbs and tenses correct in verbalisations as long as they are implied by the information in the input triple(s).

  For example, consider the input triple “Alessio_Romagnoli — youthclub — A.S._Roma” and the corresponding verbalisation “alessio romagnoli plays for the a . s . roma youth team.” Here ‘plays for’ can be inferred from the presence of ‘youthclub’ in the input triple. This is considered valid/correct and should not be marked as an error.

- However, cases such as, ‘youthclub’ being verbalised as ‘youthteam’ (‘youthclub’ is not rendered in the output and ‘youthteam’ is added in the output) or ‘AC_Hotel_Bella_Sky_Copenhagen’ verbalised as ‘hotel bella sky copenhagen’ (‘AC_Hotel_Bella_Sky_Copenhagen’ should be marked as an omission and ‘hotel bella sky copenhagen’ as addition) should be marked as errors.

- The evaluator should take extra care with units, dates and other numerical values and their conversions. For example, if '1234 m' is verbalised as '1.234 km' then it should not be considered an error. If '2006-12-31' is verbalised as '31st July 2016' then it should be marked as an omission ('2006-12-31' is not rendered in the output), and addition ('31st July 2016' is added in the output). If '610.0' is verbalised as '610 metres' then it should be considered an error where 'metres' will be an addition error.

A.3.5 Other Supplementary Materials

We have also included our participation selection form, participation reflection form and annotation guidelines as a part of the supplementary materials for this paper. We share all data and other resources on our GitHub link here: RHuidrom96/Differences-in-Semantic-Errors-Made-by-Different-Types-of-Data-to-text-Systems.

A.3.6 Tables

	System	#With Error	#Error Free
Rule-based	Baseline-FORGE2020	15	15
	DANGNT-SGU	12	18
	RALI	13	17
Non-LLM neural	Amazon-AI-Shanghai	13	17
	NUIG-DSI	10	20
	NILC	21	9
	TGEN	14	16
	CycleGT	14	16
	FBConvAI	15	15
	OSU-Neural-NLG	10	20
	cuni-ufal	13	17
	bt5	14	16
	Huawei-Noah's-Ark-Lab	22	8
LLM	GPT-3.5	16	14
	LLAMA-2 70bchat	18	12

Table A.1: Counts of each *with error* and *error free* sample for each system.

Table A.1 summarises the performance of various systems in terms of the number of *with error* and *error free* samples. Each system has a total of 30 samples. The distribution of *with error* versus *error free* samples varies across the systems, with no system being completely error-free.

System Type	Average Error Rate per System
Rule-based	0.44
Non-LLM neural	0.49
LLM	0.57

Table A.2: Average Error Rates per System Type for samples with errors

Table A.2 presents average error rates for samples containing errors across different system types. The formulas for calculating these average error rates per system type are detailed in equations A.1 and A.2. Rule-based systems exhibit the lowest average error rate of 0.44. In comparison, Non-LLM neural systems have an average error rate of 0.49. LLM systems, on the other hand, demonstrate the highest average error rate of 0.57. This summary highlights how different system types perform in terms of error rates, providing insight into their relative effectiveness.

$$\text{Average Error Rate per System Type} = \frac{\sum(\text{Error Rate per System})}{\text{Number of Systems in the System Type}} \tag{A.1}$$

$$\text{Error Rate per System} = \frac{\text{Number of Samples with Errors}}{\text{Total Number of Samples}} \tag{A.2}$$

A.4 Details of Meta-Evaluation for LLM-as-Judge Metrics

A.4.1 WebNLG 2020 Dataset

The WebNLG+ 2020 Challenge focused on (i) mapping RDF triples to generate English or Russian texts (generation) and (ii) converting English or Russian texts into RDF triples (semantic parsing). Our work addresses the generation task for English. The English WebNLG 2020 dataset (version 3.0) comprises 13,211/1,667/1,779 triple sets in the train, dev, and test splits, respectively, with triple sizes ranging from one to seven and 19 DBpedia categories, three of which are unseen in the training set. The challenge involved 15 teams submitting 48 system runs, with 14 teams focusing

on English data and six on Russian data.

For the human evaluation, 10% of the test dataset was sampled (178 samples) and evaluated on each team's primary system submission. (Castro Ferreira et al., 2020) recruited 109 annotators via Amazon Mechanical Turk, providing them with instructions (criteria on a 0–100 slider scale), RDF triples, and system outputs. Each sample received three annotations.

A.4.2 Rotowire Dataset

The ReproHum initiative (Belz et al., 2021, 2022, 2023; Balloccu et al., 2024) curated two reproductions Arvan and Parde (2023); van Miltenburg et al. (2023a), of the human evaluation in Puduppully and Lapata (2021) which uses the ROTOWIRE dataset. Five systems were evaluated over three criteria on 200 instances per criteria. In total, there are 600 instances across all criteria. There are three ratings per item and the participants can only respond using the characters 'A' or 'B' to indicate their preference over the summaries. There were a total of 216 participants in the first reproductions and 262 participants in the second reproductions. The original study does not provide raw human evaluation scores, which is why we used the reproduced scores for comparison in our work.

A.4.3 Experimental Grid

For WebNLG 2020: {English}x{Llama3-8B-Instruct, Mistral-7B-Instruct-v0.2, command-r-plus-4bit}x{zero-shot}x{seeds: 42, 1738, 1234}x{Evaluator(s) set-up: one LLM as one evaluator on (a) same instructions as the human evaluation, (b) custom minimal zero-shot prompt with original definitions included, (c) custom minimal zero-shot prompt without original definitions included}.

For Rotowire: {English}x{Llama3-8B-Instruct, Mistral-7B-Instruct-v0.2, command-r-plus-4bit, Qwen2.5-7B-Instruct-1M}x{zero-shot}x{seeds: 42, 1738, 1234}x{Evaluator(s) set-up: one LLM as one evaluator on same instructions as the human evaluation across (a) models of varying sizes, (b) models of comparable sizes}.

A.4.4 Prompts

We present the prompt used in our experiments in this section. In particular, we outline the general instruction used for all LLMs, we present the prompt template for each LLM. All of this can be found in Tables A.3–A.5.

Common Template for All Prompts for J_H	
{task_desc}	Please (i) follow the instructions, (ii) be honest and fair in your judgements, (iii) try to be as correct as possible in your conclusions. For example, the text would generally get a score higher than 0 for Correctness if at least some objects in it are introduced correctly. Similarly, the text would not be rated with 100 for Correctness if at least one object is not introduced correctly.
{task_instr}	Task Instructions: You are given a piece of data and a text that describes data. Below you will find statements that relate to the text. Please rate each of these statements by moving the slider along the scale where 0 stands for 'I do not agree', and 100 stands for 'I fully agree'.
{data}	DATA:
{desc}	DESCRIPTION:
{statement}	How well do you agree with the following statements?
{datacoverage_criteria}	Data Coverage: The text contains all predicates from the data and does not miss any predicates shown in the data.
{relevance_criteria}	Relevance: The text contains only known/relevant predicates, which are found in the data. The text does not contain any unknown/irrelevant/unrecognizable predicates.
{correctness_criteria}	Correctness: When describing information about relevant predicates (those which are in both data and text), the text depicts them with correct/proper objects. Also, the text correctly introduces the subject.
{textstr_criteria}	Text Structure: The text is written in good English, i.e., it is free from grammatical errors and well-structured.
{fluency_criteria	Fluency: The text sounds logically correct and forms a coherent whole. There are no parts of the text you would change to make it sound better. The text forms a nice narrative.
{feedback}	Write your feedback in the field below if you have any (not necessary):
Llama3-8B-Instruct Prompt	

Special tokens	{llama3_bos}: ⟨|begin_of_text|⟩ ; {llama3_eos}: ⟨|end_of_text|⟩ ; {llama3_sot}: {{; {llama3_eot}: }}
Template	{llama3_bos} {llama3_sot}{task_description}{task_instruction}{data}{triples} {description}{verb} {statement}{datacoverage}{relevance}{correctness} {textstructure}{fluency}{feedback} {llama3_eot}{llama3_eos} Data Coverage: Relevance: Correctness: Text Structure: Fluency:
Mistral-7B-Instruct-v0.2 Prompt	
Special tokens	{mistral_bos}: ⟨s⟩ ; {mistral_eos}: ⟨/s⟩ ; {mistral_sot}: [INST] ; {mistral_eot}: [/INST]
Template	{mistral_bos}{mistral_sot} {task_description}{task_instruction}{data}{triples} {description}{verb} {statement}{datacoverage}{relevance}{correctness} {textstructure}{fluency}{feedback}{mistral_eot}{mistral_eos} Data Coverage: Relevance: Correctness: Text Structure: Fluency:
Command-r-plus-4bit Prompt	
Special tokens	{commandrplus_instruction}: ## Instructions\n; {commandrplus_input}: ## Input\n; {commandrplus_output}: ## Output\n; {commandrplus_criterion}: ## Criterion\n

Template	{commandrplus_instruction}{task_description} {task_instruction}{commandrplus_input}{data}{triples} {commandrplus_output}{description}{verb}{commandrplus_criterion} {statement}{datacoverage}{relevance}{correctness} {textstructure}{fluency}{feedback}Output: Data Coverage: Relevance: Correctness: Text Structure: Fluency:

Table A.3: Human Evaluation Guidelines from WebNLG 2020 given to the LLMs.

Common Template for All Prompts for J_{C+D} & J_{C-D}	
{our_task_desc}	You are an evaluator. Please read the instructions carefully and provide your judgements honestly and accurately.
{zs_minimal}	Rate the following input triple(s) and text that describes the input triple(s) on a scale from 0 to 100 based on the following criteria:
{input_triples}	Input Triple(s):
{text}	Text:
{datacoverage_criteria}	Data Coverage: The text contains all predicates from the data and does not miss any predicates shown in the data.
{relevance_criteria}	Relevance: The text contains only known/relevant predicates, which are found in the data. The text does not contain any unknown/irrelevant/unrecognizable predicates.
{correctness_criteria}	Correctness: When describing information about relevant predicates (those which are in both data and text), the text depicts them with correct/proper objects. Also, the text correctly introduces the subject.
{textstr_criteria}	Text Structure: The text is written in good English, i.e., it is free from grammatical errors and well-structured.
{fluency_criteria	Fluency: The text sounds logically correct and forms a coherent whole. There are no parts of the text you would change to make it sound better. The text forms a nice narrative.

Llama3-8B-Instruct Prompt	
Special tokens	{llama3_bos}: ⟨|begin_of_text|⟩ ; {llama3_eos}: ⟨|end_of_text|⟩ ; {llama3_sot}: {{; {llama3_eot}: }}
Template	{llama3_bos} {llama3_sot}{our_task_desc}{zs_minimal} {datacoverage}{relevance}{correctness}{textstructure}{fluency} {input_triples}{triples} {text}{verb}{llama3_eot}{llama3_eos} Output: Data Coverage: Relevance: Correctness: Text Structure: Fluency:
Mistral-7B-Instruct-v0.2 Prompt	
Special tokens	{mistral_bos}: ⟨s⟩ ; {mistral_eos}: ⟨/s⟩ ; {mistral_sot}: [INST] ; {mistral_eot}: [/INST]
Template	{mistral_bos}{mistral_sot}{our_task_desc}{zs_minimal} {datacoverage}{relevance}{correctness}{textstructure}{fluency} {input_triples}{triples} {text}{verb}{mistral_eot}{mistral_eos} Output: Data Coverage: Relevance: Correctness: Text Structure: Fluency:
Command-r-plus-4bit Prompt	
Special tokens	{commandrplus_instruction}: ## Instructions\n; {commandrplus_criterion}: ## Criterion\n {commandrplus_input}: ## Input\n; {commandrplus_output}: ## Output\n

Template	{commandrplus_instruction}{our_task_desc}{zs_minimal} {commandrplus_criterion}{datacoverage}{relevance}{correctness}{textstructure {commandrplus_input}{input_triples}{triples} {commandrplus_output}{text}{verb} Output: Data Coverage: Relevance: Correctness: Text Structure: Fluency:

Table A.4: Custom zero-shot instructions given to the LLMs. {datacoverage}{relevance}{correctness}{textstructure}{fluency} is used only for instructions with definitions.

Common Template for All Prompts for J_{H_V} & J_{H_C}	
{summaries}	Summaries
{sys_summaries}	System Summaries
{A}	A:
{B}	B:
{rank_criteria}	Ranking Criteria
{Criteria}	Coherence or Grammaticality or Repetition
{answer}	Answers
{best}	Best:
{worst}	Worst:
{analysis}	Analysis
System-level Prompt	
{gen_instr_rotowire}	You are a native speaker of English or a near-native speaker who can comfortably comprehend summary of NBA basketball games written in English.
{task_head_rotowire}	Evaluate Sports Summaries of (NBA) basketball games.

{task_instr_rotowire}	Your task is to read two short texts which have been produced by different automatic systems. These systems typically take a large table as input which contains statistics of a basketball game and produce a document which summarizes the table in natural langauge (e.g., talks about what happened in the game, who scored, who won and so on). Please read the two summaries carefully and judge how good each is according to the following criterion:
{task_desc_rotowire}	This task contains validation instances (for which answers are known) that will be used for an automatic quality assessment of submissions. Therefore, please read the summaries carefully.
System Prompt:	{gen_instr_rotowire} {task_head_rotowire} {task_instr_rotowire} {task_desc_rotowire}
Llama3-8B-Instruct Prompt	
Special tokens	{llama3_bos}: ⟨|begin_of_text|⟩ ; {llama3_eos}: ⟨|end_of_text|⟩ ; {llama3_sot}: {{; {llama3_eot}: }}
Template	{llama3_bos}{llama3_sot}{summaries}{sys_summaries}{A}{a} {B}{b} {rank_criteria}{Criteria}{answer}{best} {worst} {analysis}{llama3_eot}{llama3_eos}
Mistral-7B-Instruct-v0.2 Prompt	
Special tokens	{mistral_bos}: ⟨s⟩ ; {mistral_eos}: ⟨/s⟩ ; {mistral_sot}: [INST] ; {mistral_eot}: [/INST]
Template	{mistral_bos}{summaries}{sys_summaries}{A}{a} {B}{b} {rank_criteria}{Criteria}{answer}{best} {worst} {analysis}{mistral_eot}{mistral_eos}
Command-r-plus-4bit Prompt	
Special tokens	{commandrplus_instruction}: ## Instructions\n; {commandrplus_criterion}: ## Criterion\n {commandrplus_input}: ## Input\n; {commandrplus_output}: ## Output\n

Template	{commandrplus_instruction}{summaries}{sys_summaries} {commandrplus_input}{A}{a} {B}{b} {commandrplus_criterion}{rank_criteria}{Criteria} {commandrplus_output}{answer}{best} {worst} {analysis} Output: Best: Worst:
Qwen2.5-7B-Instruct-1M Prompt	
Special tokens	-
Template	{summaries}{sys_summaries}{A}{a} {B}{b} {rank_criteria}{Criteria}{answer}{best} {worst} {analysis} Output: Best: Worst:

Table A.5: Human Evaluation Guidelines from Puduppully and Lapata (2021) given to the LLMs.

A.5 Reproduction Study of An Automatic Error Type Analysis Method

We carried out two reproductions of the human evaluation originally reported by Dušek and Kasner (2020a). In the first, we invited the same evaluators who are the authors of the original study to reassess the generated outputs, thereby testing the study's repeatability under identical conditions. In the second, we assembled a

completely new panel of judges to examine reproducibility when one critical variable i.e., the evaluator cohort is altered. We then compare three classes of evaluation outcomes, employing tailored similarity metrics for each, and close with a discussion of the factors likely driving the observed decline in reproducibility, alongside strategies that might help improve these issues.

Dušek and Kasner (2020a) proposed an automatic method for semantic error detection (SED) in data-to-text generation (see Figure A.17 for example MR–text pairs) based on bidirectional textual entailment. Each triple in the input meaning representation (top of each example in Figure A.17) is converted via simple templates into a short textual form; entailment is then checked both from input to output and from output to input. If the input does not entail the output, a hallucination error is diagnosed (some content in the output is not present in the input); if the output does not entail the input, it is taken to mean an omission error (some content in the input is not present in the output). If input and output do entail each other, then the output is taken to be error-free.

For the entailment checking, the method used a pretrained RoBERTa model (Liu et al., 2019) using the MultiNLI dataset (Williams et al., 2018) via the Transformers library (Wolf et al., 2020). The model (referred to as the NLI-SED system below) produces probability estimates for the three possible outputs: contradiction, neutral and entailment, and a relation is accepted if its probability exceeds the others. To detect omissions, all template-generated texts of the input triples are concatenated and tested against the output in a single entailment check; for hallucinations, each triple's text is paired individually with the output. The system's final judgment for each generated sentence is one of: *OK, omission, hallucination, hallucination+omission*.

A.5.1 Manual evaluation of the SED method

The original study that is the subject of reproduction is a manual evaluation in which Dušek and Kasner compared the SED labels obtained from their NLI-SED system

MR: Atlantic City, New Jersey | country | United States
United States | capital | Washington, D.C.

NLG system output: atlantic city, new jersey comes from the united states where the capital is washington, d.c.

SED label: Reference label (derived from human rating): *not OK*; NLI-SED label: *OK*

Error analysis annotation: *other* (both system and reference are incorrect), *bad sentence*

MR: FC Dinamo Batumi | manager | Levan Khomeriki
Aleksandre Guruli | club | FC Dinamo Batumi

NLG system output: fc dinamo batumi was at levan khomeriki and manages aleksandre guruli.

SED label: Reference label (derived from human rating): *not OK*; NLI-SED: *OK*

Error analysis annotation: *reference correct, unjustified OK*

Figure A.17: Two examples each consisting of a meaning representation (MR); an NLG system output (from WebNLG 2017); two SED labels (the reference error label derived from the WebNLG 2017 human ratings, and the output from the NLI-SED system); and correctness and error label annotations as produced in one of our reproductions. Figure taken verbatim from Huidrom et al. (2022), page 2.

for data from the E2E Dušek et al. (2020) and the WebNLG Gardent et al. (2017b) shared tasks with reference labels derived from the human rating. They performed an error analysis on a sample of 100 cases where NLI-SED system generated label and reference label disagreed. In each case, they decided which was right and which was wrong, and additionally selected labels indicating the likely source(s) of any error(s), from among six different error labels for E2E, and five for WebNLG (labels as described for each dataset below).[3] Finally, in each case, the authors also provided unstructured notes which explain their annotations.

For E2E, reference labels (*OK, omission, hallucination, hallucination+omission*) were available from the E2E shared task where they were generated by the organisers with what they termed a slot-error script based on regular expression matching, with patterns informed by a subset of the E2E development set.

In the sample Dušek and Kasner annotated in their error analysis, the counts for

[3]The error classes and raw counts from the annotations we use in this section were not reported in the original publication, but were instead mapped to less fine-grained findings.

E2E		
Counts of	Slot-error Script	NLI-SED
OK	33	54
omission	42	32
hallucination	17	7
omiss+halluc	8	7
WebNLG		
Counts of	Human Ratings	NLI-SED
OK	45	54
not OK	55	46

Table A.6: Counts for different SED labels as per the reference labels (produced by the slot-error script, which is based on regular expressions, in the case of E2E, and derived from human ratings in the case of WebNLG), and the NLI-SED system.

Counts of	E2E					Counts of	WebNLG				
	Dušek & Kasner 2020	Repeat. Test (A1+A2)	CV^*	Reprod. Test (A3+A4)	CV^*		Dušek & Kasner 2020	Repeat. Test (A1+A2)	CV^*	Reprod. Test (A3+A4)	CV^*
ref correct	34	36	5.697	41	18.611	ref correct	51	38	29.126	59	14.502
SED correct	45	48	6.432	44	2.240	SED correct	42	40	4.863	35	18.127
other	18	16	11.730	15	18.127	other	7	15	72.510	6	15.339
[eatType]	5	6	18.127	6	18.127	[bias-templ]	22	16	31.484	5	125.549
[priceRange]	30	33	9.495	28	6.876	[val-format]	7	3	79.760	10	35.188
[famFriend]	10	13	26.019	8	22.156	[bad-sent]	14	27	63.225	10	33.234
[f-halluc]	8	5	46.016	22	93.054	[unj-OK]	8	25	102.722	28	110.778
[f+omiss]	16	11	36.926	24	39.880	[unj-notOK]	15	19	23.460	12	22.156
[f+halluc]	17	20	16.168	8	71.784						

Table A.7: QRA assessment of correctness and error label counts (type I results), on the ***combined*** annotations (in Repeatability Test, half randomly taken from each original annotator; in Reproducibility Test, half randomly taken from each of the new annotators).

reference labels produced by the slot-error script and for the NLI-SED system generated labels look as shown in Table A.6. In addition, there was partial agreement between the reference labels from the slot-error script and NLI-SED system in 12 cases, where both detected an omission (and one additionally detected a hallucination). There was no partial agreement on hallucinations.

In Dušek and Kasner's annotations, the script-generated labels were deemed to be correct (and the NLI-SED system's prediction wrong) in 34 out of 100 cases, and the NLI-SED system's predictions were deemed correct (and the script wrong) in 45 cases. In 18 cases, either both the script-generated labels and the NLI-SED system's

prediction were wrong or the evaluators were unable to decide. These numbers are also included in the upper part of the first Dusek & Kasner column in Table A.7.[4]

The six error type classes for the E2E error annotations were as listed below. Note that the descriptions and definitions given here were created as part of our reproductions. The implications of creating new instructions for a reproduction are discussed in Section A.5.4.

Each error class, derived from (Dušek and Kasner, 2020a), represents a different possible source of an error made by Dušek and Kasner's NLI-SED system or the slot-error script, and as many error classes were selected as applied in each case, in some cases none were selected (frequencies are shown in the lower part of the first Dusek & Kasner column in Table A.7). These error classes tend to apply predominantly to either the NLI-SED system or the slot-error script, indicated by underlines below. The short labels in square brackets are used to refer to each class in the results tables below.

1. **Error related to *eatType=restaurant* slot value [eatType]:** The incorrect SED label (produced either by the NLI-SED system or the slot error script) is likely caused by something involving the slot/value pair eatType=restaurant, e.g. not detecting a hallucination when the eatType slot is not in the input, but the output mentions a restaurant.
2. **Error related to *priceRange* slot [priceRange]**: The incorrect SED label (produced by either the NLI-SED system or the slot error script) makes an error related to the priceRange slot, e.g. incorrectly identifying a hallucination in the priceRange slot, when the price range information has in fact been correctly verbalised.
3. **Error related to *familyFriendly attribute* [famFriend]**: The incorrect SED label (produced by either the NLI-SED system or the slot error script) makes an error related to the familyFriendly slot, e.g. incorrectly identifying an omission when the information has in fact been correctly verbalised.

[4]Numbers don't add up to 100 because of missing annotations.

4. **Other false negative hallucination ('off-topic blabber') [f-halluc]**: The incorrect SED label (produced by either the NLI-SED system or the slot error script) fails to detect a hallucination (unrelated to E2E slots) present in the verbalisation.

5. **Other false positive omission ('unjustified omission') [f+omiss]:** The incorrect SED label (produced by either the NLI-SED system or the slot error script) wrongly detects an omission in the verbalisation.

6. **Other false positive hallucination ('unjustified hallucination') [f+halluc]:** The incorrect SED label (produced by either the NLI-SED system or the slot error script) wrongly detects a hallucination in the verbalisation.

For the WebNLG sample, Dušek and Kasner (2020a) created reference SED labels by mapping human quality judgements on a 1–3 scale (crowdsourced for WebNLG 2017) to *OK* (>=2.5) and *not OK* (<2.5). The crowdscourced quality judgements exist for a subset of 223 inputs from the WebNLG 2017 test set each paired with 10 different NLG outputs from participating systems. SED labels were taken to differ unless they were both *OK*, or one was *not OK* and the other was one of *omission, hallucination, omission+hallucination*).[5]

In the sample Dušek and Kasner annotated in their error analysis, the counts for reference labels derived from human ratings and for the NLI-SED system generated labels look as shown in the lower half of Table A.6. The *not OK* label count of 46 shown for the NLI-SED system breaks down into 29 cases of omission, 13 cases of hallucination, and 4 cases of combined omission+hallucination).

In Dušek and Kasner's annotations, the reference label (the mapped human rating) was deemed to be correct (and the NLI-SED system prediction wrong) in 51 out of 100 cases, and the NLI-SED system prediction was deemed correct (and the reference label wrong) in 42 cases. In 7 cases either both reference label and NLI-SED system prediction were deemed wrong or the evaluators were unable to

[5]One case of agreement, where the mapped human label was *Not OK* and the NLI-SED system produced *omission*, was included by mistake.

decide. These numbers are also shown in the top part of the second Dusek & Kasner column in Table A.7)

The five error class labels, derived from (Dušek and Kasner, 2020a), for the WebNLG error annotations were as shown below. Each item may have more than one or none of these. The first three classes indicate, where possible, the likely source of the error in the SED label that was deemed wrong (produced by either the NLI-SED system or the mapped human ratings). Otherwise one of the last two will apply. Label frequencies are shown in lower part of the second Dusek & Kasner column in Table A.7. Please note that each error class predominantly applies to the underlined method (NLI-SED system, reference).

1. **NLI-SED system error due to poor triple-to-text input mapping ('biased template')** [**bias-templ**]: Incorrect NLI-SED system label due to an inappropriate template being used in mapping the input triples to text (templates tend to work better for certain subject/object values, but the same template is used for all cases with a given predicate), resulting in ungrammatical sentences or even shift in meaning.

2. **NLI-SED system failure to recognise subject or object semantic equivalence ('value format')** [**val-format**]: In the verbalisation, the formatting of a subject or object differs from the input to the extent where the NLI check in the NLI-SED system failed to recognise them as equivalent in meaning (e.g. metres vs. kilometres).

3. **Incorrect reference SED label due to disfluent verbalisation ('bad sentence')** [**bad-sent**]: The human reference label, mapped to *not OK*, is incorrect, and this is likely because the human rating was affected by the disfluency/ungrammaticality of the verbalisation.

4. **Other cases of incorrect *OK* label ('unjustified OK')** [**unj-OK**]: The incorrect label (from either the human references or the NLI-SED system) is OK, and none of the above apply.

5. **Other cases of incorrectly identifying a semantic error ('unjustified not OK') [unj-notOK]**: the incorrect label (from either the human references or the NLI-SED system) either literally a *not OK* label, or one of *omission, hallucination, omission+hallucination*, *not OK*, and none of the above apply.

A.5.2 Reproducing human evaluation of the SED method

A.5.2.1 Reproduction targets

In the present context, there are four types of results that are candidates for reproduction; the first three are taken from QRA++ (Belz, 2025): (I) *single numeric values* for the same measure (e.g. the overall number of times the SED label produced by the NLI-SED system was correct); (II) *sets of numeric values for a set of related measures* (e.g. the numbers of input/output pairs annotated with each error label); (III) *sets of discrete labels* from the same task (e.g. the correct/incorrect labels assigned to the NLI-SED system labels and the reference SED labels); and (IV) *findings relating to differences between systems* on the same evaluation method and test metric.

In order to draw conclusions regarding repeatability and reproducibility, results from original and reproduction studies need to be compared, and how they're compared depends on which type (*I*, *II*, *III*, or *IV* above) a result is. We pick this up again in Section A.5.2.2; here we list the results of types (I)–(III) from Dušek and Kasner that we attempt to reproduce in our two reproduction studies (relating to differences between systems (type *IV*) is not performed):

I. Single numeric values (overall counts):

   a. Count of reference correct;

   b. Count of NLI-SED system correct;

   c. Count of both reference and NLI-SED system incorrect or evaluators couldn't decide;

d. Count of individual error labels, six different labels for E2E, five for WebNLG (see Tables A.7 and A.8 for short-form labels).

II. Sets of related numeric values:

a. Set of counts of *Correctness* labels (i.a–i.c above);

b. Set of counts of SED *Error-class* labels (i.d above).

III. Sets of categorical values:

a. Set of *Correctness* labels (one of {*NLI-SED, reference, neither*}; exactly one label per evaluation item);

b. Set of SED *Error-class* labels; multiple labels per evaluation item).

A.5.2.2 Approach to Reproduction

For results of type *I* above (where we have single measured quantity values to compare), we follow the quantified reproducibility assessment (QRA++) approach Belz (2025) which means (a) identifying and documenting the properties of evaluation experiments as standardised attribute-value pairs (*conditions of measurement* in QRA++ terms); and (b) computing the small-sample coefficient of variation (CV^*) over compared quantity values, as the measure of degree of reproducibility. QRA++ assessment results are shown in Tables A.7 and A.8 and discussed in Section A.5.3.1.

For type *II* results (Table A.9, Section A.5.3.2) we compute Pearson's r for pairwise correlation.

For results of type *III*, we compute Fleiss's kappa (the multi-evaluator generalisation of Scott's pi) on aligned sets of categorical values where we have exactly one label per item (which is the case for the correctness labels), and Krippendorff's alpha where we have multiple labels per item (which is the case for the error labels). Results are shown in Table A.10 and discussed in Section A.5.3.3.

A.5.2.3 Two Reproductions

Our two reproduction studies repeated the Dušek and Kasner evaluations as closely as possible, the first using the same evaluators, the second using different evaluators. There were two complicating factors, necessitating the use of (i) new evaluator instructions, and (ii) a different way of allocating evaluators to evaluation items.

The reason for the difference in evaluator instructions is that in the original work, instructions were not written down, a shared understanding being developed in the course of the work instead. In order to repeat the evaluations with new evaluators less familiar with the work, instructions had to be written down and shared which were then used in both reproductions. The instructions are included verbatim in the Appendices A.3.3 and A.3.4.

Regarding evaluator allocation, in the original work, the work was shared between the two authors who each did about half of E2E and half of WebNLG, but it was not recorded who did which ones. For that reason, we decided to get the evaluators in the first reproduction (the original two authors) to each annotate all 100 E2E items and all 100 WebNLG items, and then we randomly selected half from each evaluator pair for a like-for-like comparison (in the tables below we call this the *combined* set of annotations). Assessing the similarity between these combined results and the original results forms the main body of our reproduction study: type *I* results are shown in Table A.7, type *II* in Table A.9, and type *III* in Table A.10.

Additionally, we compare the four complete sets of annotations with the original annotations individually, for the single numeric values (type *I* results) from E2E and WebNLG only (Table A.8).

Each evaluator worked on a separate Google spreadsheet in the exact same format as in the original study,[6] except that in the repeatability test which involved the original annotators, we shuffled the order of evaluation items to avoid inadvertent recall of original annotations.

[6] A blank copy of the evaluation sheet can be found here: `https://docs.google.com/spreadsheets/d/1_4DZVu6Ow-9kZOjQJCjg2qZCLUt435Og`

Counts of	E2E						Counts of	WebNLG					
	D&K	A1	A2	A3	A4	*CV**		D&K	A1	A2	A3	A4	*CV**
ref correct	34	41	31	37	50	21.325	ref correct	51	43	34	55	48	19.598
SED correct	45	45	53	41	47	10.594	SED correct	42	44	30	37	48	19.291
other	18	14	15	22	3	55.016	other	7	12	13	8	4	46.984
[eatType]	5	10	5	2	8	57.382	[bias-templ]	22	18	16	7	2	70.856
[priceRange]	30	31	39	42	9	47.756	[val-format]	7	1	3	26	0	162.088
[famFriend]	10	11	10	8	1	56.718	[bad-sent]	14	27	15	9	6	63.275
[f-halluc]	8	8	3	38	0	149.505	[unj-OK]	8	31	17	48	0	102.418
[f+omiss]	16	10	14	42	6	89.937	[unj-notOK]	15	16	25	26	1	67.727
[f+halluc]	17	15	24	19	4	52.288							

Table A.8: QRA assessment of individual numeric results (type *I*), using the 4 sets of ***individual*** annotations.

A.5.3 Results and analysis

A.5.3.1 Comparison of type *I* results

The results from the QRA test on label counts (type *I* results, i.e. single numeric values) for the *combined* annotations are shown in Table A.7. The counts from the original study are in the Dusek & Kasner column in the left half of the table for E2E,[7] and in the right half for WebNLG. Counts from the repeatability (original annotators) and reproducibility (new annotators) tests and the corresponding CV* scores are shown in columns labeled as such in each half.

Looking at correctness label counts for E2E (rows 1–3, left half), the original annotators (A1+A2) are on the whole better able to reproduce their own results than the new annotators (A3+A4), which is as expected. However, if we look at the corresponding figures for WebNLG (right half) it turns out that here, the *new* annotators reproduce the original counts more closely. In terms of differences between correctness labels, the 'SED correct' counts are overall easiest to reproduce.

Moving on to error class counts, for E2E, CV* is broadly the same for original/new annotators for error classes relating to specific slots (eatType, priceRange, famFriend), but considerably worse for the new annotators for the remaining, more generic, error classes. For WebNLG, it is a more mixed picture: the new annotators (who are the thesis author and another colleague from the same lab) reproduce the

[7] Counts for *ref correct*, *SED correct*, and *other* do not add up to 100 because of 3 missing annotations.

original counts better than the original annotators for error classes val-format and bad-sent, worse for error classes bias-templ and unj-OK, and equally well for error class unj-notOK.

Table A.8 sheds additional light on the reproducibility of individual category counts by providing a larger sample, with four sets of new annotations compared to the original one set of annotation, for each of E2E and WebNLG, thereby giving more reliable estimates of CV*. The two halves of the table are structured as in Table A.7.

The results in Table A.8 provide overall estimates across all five sets of annotations of the degree of reproducibility of the individual types of counts. For both E2E and WebNLG, correctness label counts are far easier to reproduce than error class counts which is as expected. Beyond that, again the 'SED correct' count is the most reproducible for both E2E and WebNLG. For E2E, counts for errors related to the priceRange slot (priceRange) are easiest to reproduce, whereas false negative hallucinations (f-halluc) are by far the hardest. For WebNLG, counts for bad-sent (bad grammar/fluency likely leading to 'not OK' label) are easiest to reproduce, and counts for val-format (phrases that are semantically equivalent not being recognised as such) are by far the hardest.

To put these CV* numbers into perspective, in the first ReproGen Shared Task, all except one (an outlier above 70) of the CV* scores for human evaluations were below 39 Belz et al. (2020).

A.5.3.2 Comparison of type *II* results

The results in the preceding section showed how reproducible correctness and error label *counts* were, for each count type independently, and regardless of whether labels were attached to the same items. In this section, we look at sets of counts in conjunction, and in the next section we look at labels as attached to evaluation items. Table A.9 presents results from correlation tests on the set of all three correctness labels (top half), and on the set of all five (WebNLG) or six (E2E) error labels (sets

	Pearson's r	E2E	Web-NLG
Correctness	Orig vs. A1+A2	0.999	0.965
	Orig vs. A3+A4	0.948	0.963
	A1+A2 vs. A3+A4	0.959	0.857
Error classes	Orig vs. A1+A2	0.947	0.209
	Orig vs. A3+A4	0.620	-0.630
	A1+A2 vs. A3+A4	0.373	0.414

Table A.9: Pearson's r for counts of correctness and error-class labels (type *II*), using the ***combined*** annotations (see Table A.7 caption and Section A.5.2.3).

of related numeric values, i.e. type *II* results). Here too we are using the *combined* annotations.

We can see from the Pearson's r values that for both E2E and WebNLG all correlations are strong for the sets of correctness label counts. For the error class label count sets, on the other hand, while the original annotators achieve high correlation with their own earlier label counts for E2E, they do not for WebNLG, where the correlation is weak. The correlation between the A1+A2 and A3+A4 error label counts is weak to medium for both E2E and WebNLG. The new annotators do a reasonable job reproducing the original labels for E2E (r=0.62), but worst by far is the pronounced negative correlation between the original and the new annotators for the WebNLG error labels.

A.5.3.3 Comparison of type *III* results

The results from the agreement tests with Fleiss's kappa and Krippendorff's alpha on both label types as attached to evaluation items (type *III* results, i.e. related sets of categorical values), again on the *combined* annotations, are shown in Table A.10. For E2E and correctness labels, a similar picture emerges as previously in that agreement is similarly good across all comparisons, reflected also in the '%=' column which shows the percentage of times there was perfect agreement across all labels and all annotators in a given comparison. For E2E and error class labels, the original

			E2E	% =	Web-NLG	% =
Correctness	Fleiss's κ	All	0.674	71%	0.269	40%
		Orig vs. A1+A2	0.676	81%	0.140	50%
		Orig vs. A3+A4	0.677	81%	0.527	73%
		A1+A2 vs. A3+A4	0.643	78%	0.112	48%
Error classes	Kripp.'s α	All	0.467	12%	0.165	3%
		Orig vs. A1+A2	0.735	60%	0.207	21%
		Orig vs. A3+A4	0.347	15%	0.114	7%
		A1+A2 vs. A3+A4	0.330	18%	0.166	12%

Table A.10: Fleiss's kappa for correctness and Krippendorff's alpha for error-class labels (type *III*), using the ***combined*** annotations (see Table A.7 caption and in text). '% =' = percentage of items with identical labels.

annotators have strong agreement with their own original annotations, and the rest of the comparisons show medium agreement.

Again the picture is more mixed for WebNLG, where the new annotators have medium label-level agreement with the original labels for correctness, but for the other seven comparisons, label-level agreement is quite startlingly low (0 being chance). Meanwhile, original annotations and A3+A4 for WebNLG have the highest percentage with identical labels (=73%) highlighting that the second pair of annotations closely mirrors the original correctness judgments compared to the first pair (Orig. vs. A1+A2).

A.5.4 Discussion

The error-analysis based evaluation method in this work compares system outputs with reference outputs, but rather than just counting it against the system if there is disagreement between the two, it examines which is actually right in each case, also identifying the types of errors made by each. For E2E, 4 out of 5 sets of annotations (Table A.8) agreed that the NLI-SED system was more often correct than the (automatically generated) references using the script; for WebNLG the balance was slightly tipped in favour of the references (here derived from human

ratings). These were important findings in the original paper, and are confirmed in all reproductions.

Across all reproductions on the E2E dataset, errors involving the priceRange predicate are by far the most frequent, while those associated with eatType and famFriend occur least often. By contrast, no error category in the WebNLG annotations shows a consistent pattern across every annotation set.

The degree to which the different types of results were reproducible varied. The more high-level correctness labels saw far better agreement than the more fine-grained error labels which also involve greater cognitive load. Moreover, the different backgrounds of the annotators and their degree of familiarity with the system and data may also have contributed to variation.

Differences may be due to unclear instructions in the original study and not enough hands-on training for our annotators. To address this, similar future work should give annotators comprehensive, category-specific information, adopt an protocol where one can refine guidelines through pilot tests and detailed feedback and, once annotators reach a shared understanding, formalise and document the finalised instructions.

A.5.4.1 Conclusion

In conclusion, we carried out two reproductions of a manual error analysis of the outputs from a semantic error detection (SED) system based on two-way entailment detection by an NLI model. We selected three types of results for reproduction, namely single numeric values, sets of numeric values, and sets of discrete labels, each of which requires different methods of comparison. While overarching findings held, finer-grained agreement varied, particularly for WebNLG and detailed error categories. Overall, E2E results were more reproducible than WebNLG, and simple correctness labels proved more stable than nuanced error classes.