Interpretative Evaluation Metric for Factual Information Inconsistency

Mingwei Huang and **Heng Ji**
Computer Science Department
University of Illinois at Urbana-Champaign
{mingwei6, hengji}@illinois.edu

Abstract

Many recent automatic evaluation metrics are designed to target hallucination problems in model outputs which impede the progresses of many natural language generation tasks. However, most of the metrics are not interpretative and subject to specific tasks. In this paper, we 1) introduce a possible form of interpretative evaluation metric for semantic differences, and 2) propose an automatic interpretative evaluator, named FactDiff, which takes the advantage of the semantic parsing task, Abstract Meaning Representation (AMR), to generate concrete and comprehensive interpretation of factual information inconsistency between generated text and references[1].

Text	
$\mathcal{R}$: **Scott** has a dog. $\mathcal{H}$: **Scott** has a cat.	
Metric	Evaluation Output
BLEU_3	0.5109
FactDiff	Incorrect-entity hallucination error: 0.5 Incidence: The **possession** (**pet**) of **Scott**, which should be dog according to the reference, is misinformed to be cat. It may be a hallucination of wrong pet. [2] Incorrect-context hallucination error: 0 Missing/additional information error: 0

Table 1: Example output of 3-gram BLEU and FactDiff.

1 Introduction

Recent researches in natural language generation (NLG) problems have made remarkable progresses, and many NLG models are implemented to tackle a variety of applications, such as summarization (e.g. Gehrmann et al., 2018; Nayeem et al., 2018), translation (Koehn, 2009; Ma et al., 2020), paraphrasing (e.g. Shen et al., 2020), and dialog (e.g. Vinyals and Le, 2015; Xie et al., 2022). Nevertheless, the evaluation of model outputs has started to be one of the bottlenecks for NLG tasks (Wiseman et al., 2017; Tian et al., 2019).

Galliers and Jones (1993) categorized evaluation for generated text into two parts: 1) extrinsic approaches that measure how the system influence users' actions, and 2) intrinsic approaches which involves assessments of the quality of generated text for given tasks, and usually, the quality is evaluated in terms of correctness or usefulness. Majority of academic researches rely on some intrinsic approaches due to its feasibility (Gkatzia and Mahamood, 2015; Gehrmann et al., 2022). And among the intrinsic approaches, as a trade-off between assessment accuracy and cost, automatic evaluation metrics are more prevalent than human evaluations as a part of the model development pipeline in researches.

However, both human and automatic evaluation metrics do not directly reflect detailed features or disadvantages of generated text. Though some evaluation metrics take concrete semantic or syntactic features into account (Nenkova et al., 2007; Goyal and Durrett, 2020), the final form of the evaluation output is still one or multidimensional quantitative or categorical scores. The quantitative evaluation results, especially that are generated by automatic evaluation metrics which are not perfectly accurate, do not provide an interpretation on the meanings of the scores, let alone the characterized features of generated text. This problem impedes the understanding and comparisons of NLG models. Many researchers need to manually read system generated outputs to comprehend and report the detailed features that are not captured by evaluation metrics to improve and present their models (Gehrmann

[1]Code and outputs will be made publicly available at https://github.com/Scott-Huang/FactDiff

[2]Manually edited to be simpler and understandable.

et al., 2022). And as suggested by van Miltenburg et al. (2021) and Bender and Koller (2020), focusing on and understanding the limitations of NLG models are as important as improving aggregated scores.

Evaluation metrics should also generate qualitative analysis besides quantative scores to make the model development process more efficient and interpretable. To this end, we propose an interpretative evaluation model, named FACTDIFF, to generate detailed reasons for each factually inconsistent output text and references pairs and and an overall summary of the output. The format of output is presented with a comparison between a commonly used evaluation metric across various NLG tasks, BLEU (Papineni et al., 2002; Lin and Och, 2004) and ours in table 1. Our model produces a multi-dimensional measure on various types of possible hallucination errors and the detailed comprehensible explanation on where and how the errors are detected.

The details of our framework will be introduced in section 3 and 4. In addition, the experiment performance with qualitative analysis will be listed in section 5.

2 Abstract Meaning Representation

The implementation of our model make use of the task, **Abstract Meaning Representation (AMR)** which provides robust and semantically rich graph representation of text (Banarescu et al., 2013). The example sentence, "Scott has a dog" can be parsed into the following AMR graph. The AMR graph

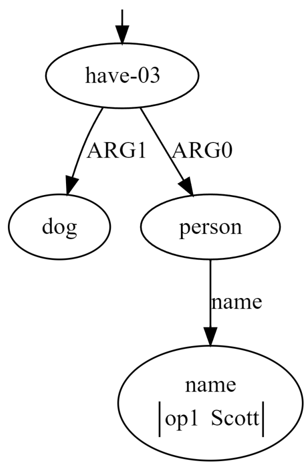

Figure 1: AMR graph for sentence, “Scott has a dog." Generated by an online parser (Lindemann et al., 2019).

captures “who is doing what to whom” in a sentence. The text is parsed at sentence level with cross-sentence entity co-reference, and each sentence is parsed into a directed, acyclic graph [3] where vertices are entities/concepts and edges are relations. In this example, “person" is an entity vertex with a name that has a value being “Scott" and is an argument, “ARG-0", of the action “have-03". The details and usage of AMR parsing will be introduced later in section 4.

3 Task Formulation

Given an input sentence S and a list of references $\mathcal{R} = \{R_1, \ldots, R_n\}$, assume that we have an almost perfect parser that can parse a sentence into the form $G = (V, E)$ where V is the set of explicit or implicit concepts in the sentence which can be named entities, nominal mentions, or any common nouns; let K be the set of all possible relations between two concepts, and $E \subseteq V \times V \times K$ is the set of all relations amongst the parsed concepts. Denote the parsed semantic representation of the input sentence as G_S and the representation of references as $\{G_{R_1}, \ldots, G_{R_n}\}$.

Let $p(v)$ for concept $v \in V$ be the set of atomic facts that v refers to (Wittgenstein, 1922). And we use $\mathbb{E}_{e \in p(v)}[e \in p(v')]$ to measure the overlap between two concepts v and v'. Obviously, there may not exists a clear boundary of $p(v)$ (Zaefferer, 2019), but practically, we can approach the concept similarity with some empirical method $s_{concept} : V \times V \rightarrow [0, 1]$, such as cosine similarity of word embedding, so that:

$$s_{concept}(v, v') \approx \mathbb{E}_{e \in p(v)}[e \in p(v')]. \quad (1)$$

Sticking with Wittgenstein's notion of atomic facts, let $c(v, E) := \bigcap_{(v,v',r)\in E} p(v', r)$ be the context of the mentioned concept v which represents the range of all possibilities of v. For example, the entity, “Scott" in the sentence “Scott has a dog and walks his dog", has a context of “has a dog" and “walk the dog", meaning “Scott" must be something that is possible to have a state of owning a dog and being able to walk and is walking his dog. Despite its infeasibility to be directly approximated, let's assume there exists a function $s_{context} : V \times E \times V \times E \rightarrow [0, 1]$ that can calculate the context overlap with an acceptable accuracy

[3] Many AMR parsers may generate graphs with cycles. And it is a problem for evaluating and using parsed AMR graphs.

such that

$$s_{context}((v, E), (v', E')) \approx \mathbb{E}_{e \in c(v,E)}[e \in c(v', E')]. \quad (2)$$

The detailed implementations of $s_{concept}$ and $s_{context}$ are available at section 4.2.

Denote p^{-1} as the reverse function referring atomic facts with words, our task is to report any difference between input sentence semantics $G_S = (V_S, E_S)$ and references $\{(V_{R_1}, E_{R_1}), \ldots, (V_{R_n}, E_{R_n})\}$ with the following purpose: for any $v \in V_S$, report

$$p^{-1}\left((p(v') \cap c(v', E'))\Delta(p(v) \cap c(v, E))\right),$$

which is a description in natural language of the symmetric difference between all relevant information of v and of v'.

We can simplify the report without much information loss using the following tricks:

1) if $\max_{i,v' \in V_{R_i}} s_{concept}(v, v')$ is large with maxima argument i, v', meaning that v and v' are likely refer to the same concept, and $s_{context}((v, E), (v', E_{R_i}))$ is large, meaning they have similar contexts too, then don't report since both the concepts and the contexts are similar.

2) If both $\max_{i,v' \in V_{R_i}} s_{concept}(v, v')$ and $\max_{i,v' \in V_{R_i}} s_{context}((v, E), (v', E_{R_i}))$ are small, meaning that there exists no similar concept or context in references, then report (v, E) as make-up information since no supporting evidence can be found in references.

3) If $\max_{i,v' \in V_{R_i}} s_{concept}(v, v')$ is large with maxima argument i, v' and $s_{context}((v, E), (v', E_{R_i}))$ is small, report

$$p^{-1}\left((c(v', E')\Delta c(v, E)) \cap p(v)\right), \quad (3)$$

which is the description of the difference between context $c(v, E)$ and $c(v', E')$ when applying to concept v.

4) Else if $\max_{i,v' \in V_{R_i}} s_{context}((v, E), (v', E_{R_i}))$ is large with maxima argument i, v' and $s_{concept}(v, v')$ is small, report

$$p^{-1}\left((p(v')\Delta p(v)) \cap c(v, E_S)\right), \quad (4)$$

which is the description of the difference between concepts v and v' under context $c(v, E_S)$.

4 Our Approach

We first parse input sentences with a pretrained stack-transformer-based AMR parser (Fernandez Astudillo et al., 2020) into AMR graphs with entity coreference. After that, we align vertices based on their concepts and contexts. We then report possible differences between aligned vertices pairs as well as the unmatched vertices which are considered to be additional or missing information, and analyze their differences and possible causes.

4.1 AMR Parsing

As discussed before, it is not trivial to come up $s_{context}$ in equation 2 to approximate context overlap, especially in an unsupervised setting. Instead of adapting machine learning approach, we decide to rely on a semantic representation of unification-based formalism (Kay, 1979, 1984), which provides a simplified, topological and combinable notation of context as attributes. And the abstract meaning representation further simplifies the notions of context attributes using standard feature structure with directed acyclic graph parsing of the attributes (Shieber, 1987; Banarescu et al., 2013). With the AMR graph of parsed text, we are then able to use the neighbor nodes of concept v to approach the context of it, $c(v, E)$.

We employ the stack-transformer-based AMR parser (Fernandez Astudillo et al., 2020), pretrained on AMR 3.0 annotations[4], to parse sentence S into $G_{AMR} = (V_{AMR}, E_{AMR})$. V_{AMR} includes all concepts and actions, such as "Scott", "dog", and "has". The parser further annotates the meaning of action "has" with PropBank frames (Kingsbury and Palmer, 2002; Palmer et al., 2005) with explanations shown below. For all parsed con-

Roleset id: have.03 , *own, possess*
Roles:

Arg0-PAG: *owner* (vnrole: 100.1-pivot, 39.4-agent)
Arg1-PPT: *possession* (vnrole: 100.1-theme, 39.4-patient)

Figure 2: A PropBank frameset of have-03

cept nodes that are not frames, we consider them as vertices of G_S with $V_S = \{v \textit{ is not frame} \mid v \in V_{AMR}\}$. For all frame nodes $v \in V_{AMR}$, we make use of the frame definition to convert v to (v_1, v_2, r) for all arguments $v_1 \neq v_2$ of v where $(v, v_1, \text{:ARGX}), (v, v_2, \text{:ARGX}) \in E_{AMR}$. In the example of "Scott has a dog", "has" is parsed into attribute relations ("Scott", "dog", owner) and ("dog", "Scott", possession). It is possible that a frame v only has one argument, such as "dog

[4] https://catalog.ldc.upenn.edu/LDC2020T02

is owned", we parse it with an empty vertex to represent all possible instances, (“dog", None, possession). The possible types of relations parsed in E_{AMR} are shown in table 2. We directly parse

Core roles	:ARG0, :ARG1, :ARG2, ...
Non-core roles	:age, :beneficiary, :condition :example, :extent, :li :part, :path, :time, ...
Unit entity	:century, :day, :decade, :season, :weekday, :year, ...
Special case	:prep-from, :prep-among, ...

Table 2: Examples of AMR roles. Full list of roles, definitions and examples is in AMR guidelines and the official website.

these relations from E_{AMR} into E_S without any modification.

Moreover, we perform post-processing of named entity co-reference in V_{AMR} before generating V_S. If any two named entities have the same or equivalent names, we combine the two nodes as one.[5]

4.2 Node Alignment

Given the semantic representation of input sentence G_S and a reference G_R, for a concept $v_s \in V_S$, we want to align the node v_s with $v_r = \arg\max_{v_r \in V_R} s_{context}((v_s, E_S), (v_r, E_R))$. We define $s_{context}$ in equation 1 with a decaying weight factor $0 \leqslant \alpha \leqslant 1$ and a coefficient $0 \leqslant \beta \leqslant 1$ recursively:

$$
\begin{aligned}
& s_{context}((v_s, E_S), (v_r, E_R); \alpha, \beta) \\
= & \sum_{(v_s, v_s', r_s) \in V_S} \max_{(v_r, v_r', r_r) \in V_R} \Big(\\
& \beta(s_{concept}(v_s', v_r') + \mathbb{1}[r_s \cap r_r \neq \varnothing]^6) \\
& + \alpha(1-\beta) s_{context}((v_s', E_S), (v_r', E_R); \alpha, \beta)\Big).
\end{aligned}
$$

Note that the function will not terminate if either G_S or G_R contains cycles. Although theoretical AMR graphs are acyclic, it is possible to have circles in parsed graphs and we need to break the circle if found. In this way, we can align all nodes V_S with V_R and calculate their context overlaps.

We adapt a modified Hill-climbing method by Cai and Knight (2013) to optimize the algorithm to be efficient but still effective. And we define $s_{concept;t}(v, v')$ in equation 2 by hard classification with a threshold t: if v is a named entity, return $\mathbb{1}[v$ and v' have equivalent names]; else return $\mathbb{1}[\text{similarity}(v, v') > t]$ where the similarity is calculated through their word embedding trained on wiki (Mikolov et al., 2013; Bojanowski et al., 2017).

4.3 Inconsistency Interpretation

Once nodes are aligned and we can easily see which pairs of nodes have similar concepts but different contexts or vice versa, we then report the descriptive difference between the paired nodes by directly stating the nodes and contexts.

If two nodes v_s, v_r are similar but in different contexts, according to equation 3, we state the node v_s and different $(v_s', v_r'), (r_s, r_r)$ pairs for $(v_s', v_s, r_s) \in E_S, (v_r', v_r, r_r) \in E_R$ or $(v_s, v_s', r_s) \in E_S, (v_r, v_r', r_r) \in E_R$ through a manually designed template.

If two nodes v_s, v_r are different but in similar contexts, according to equation 4, we state the context $\{(v_1, v_2, r_s) \mid v_1 = v_s \vee v2 = v_s \wedge (v_1, v_2, r_s) \in E_S\}$ and two nodes v_s, v_r through a manually designed template.

After that, we calculate and report the percentage of incorrect nodes, contexts, and unmatched nodes which are considered missing or additional information.

In addition, we incorporate some external knowledge to postulate and analyze the cause of inconsistency. Currently, we only implement a hyponym and antonym check between two different nodes through WordNet (Fellbaum, 2005). And we are working on using knowledge bases to reason about inconsistent context of a node.

5 Experiments

Since most evaluation studies are focusing on summarization and machine translation (Reiter and Belz, 2009), we decide to conduct an experiment on evaluating summarization which is a semantic-based task.

5.1 Data

The experiment is run on FRANK dataset (Pagnoni et al., 2021), which composes around 500 documents and 2250 summaries in CNN/DM (Hermann et al., 2015) and XSum (Narayan et al., 2018). The dataset also includes a typology of factual errors that categorize semantic inconsistency into 9 types.

[5] This should be the job completed by the AMR parser, but we find that the parser often fails in co-reference.

[6] There can be multiple relations on one edge. Also, if an argument of a frame has multiple definitions, it will parsed into multiple relation too.

<table>
<tr><th>Reference</th><th>BertSum output</th></tr>
<tr><td>gary locke will be confirmed as kilmarnock boss on friday . the club went unbeaten during Locke 's first six games in charge . the 39-year-old former killie defender has paid tribute to his players .</td><td>gary locke will be confirmed as kilmarnock 's permanent manager on friday . The 39-year-old has been given a three-year deal at kilmarnock . locke had been working as no 2 under allan johnston when the manager announced in february that he would be leaving the club at the end of the season</td></tr>
<tr><th colspan="2">Evaluator Output</th></tr>
<tr><td colspan="2">Incorrect-context hallucination error: 0.13
Incidence:
Gary Locke is benefactive, hearer(ARG2) of the action, confirm-01 Kilmarnock boss(ARG1), but the output shows thing confirmed(ARG1) of the action is Kilmarnock manager.
Gary Locke poss tribute, but the output shows Gary Locke is entity(ARG2) of the action give-01 deal(ARG1).
Incorrect-entity hallucination error: 0
Missing/additional information error: 0.56
Incidence: ...</td></tr>
<tr><th>Reference</th><th>PtGen Output</th></tr>
<tr><td>steph surry scored 36 points to lead the golden state warriors to a 96-88 victory over the oklahoma city thunder and into the nba finals .</td><td>golden state warriors beat golden state warriors 4-1 to reach the last eight of the women 's super league .</td></tr>
<tr><th colspan="2">Evaluator Output</th></tr>
<tr><td colspan="2">Incorrect-context hallucination error: 0
Incorrect-entity hallucination error: 0.33
Incidence:
Golden State Warriors should not be loser(ARG3) of the action, Golden State Warriors(ARG0) beat-03, and it should be Oklahoma City Thunder.
4 1 should not be score-entity of the action, Golden State Warriors beat-03, and it should be 96 88.
Women 's Super League should not be goal, end state, thing attained(ARG1) of the action Golden State Warriors reach-01, and it should be NBA.
Missing/additional information error: 0</td></tr>
</table>

Table 3: Example of FactDiff evaluation output on system generated text and references in FRANK dataset.

The dataset is not pre-processed and we feed the references and summaries into the AMR parser and get 2250 pairs of data.

5.2 Qualitative Analysis

Since our work only aims to generate interpretation on possible inconsistency rather than an accurate quantitative scores, we choose not and it is very difficult to run any automatic evaluation about its accuracy and informativeness. Instead, we report a qualitative analysis after manually reading the interpretation of references and system generated text. And the examples of evaluation output are shown in Table 3.

During the experiment and the development process, we found that the actual bottleneck is the AMR parser and the quality of generated text. According to our manual observation, more than half of generated text are parsed into nonsense which is unusable for evaluation. We contribute partial causes of parsing failure to the quality of generated text. Many of the system output have incomprehensible pieces or entire sentences which violate the assumption of stack-based syntactical parser that the sentence should be valid and correct. We also attempted to use other transformer-based AMR parser (Bai et al., 2022) and found it suffer severe hallucination. Furthermore, we calculated the sentence perplexities of references and generated text using language model, BERT (Devlin et al., 2018) and GPT-2 (Radford et al., 2019), but result shows that the perplexity is not highly correlated with either understandability or the quality of parsed graphs. So we conclude that we cannot use any supervised semantic parsing technique on model outputs, as there still exists an intrinsic difference between them and natural languages (Saggion et al., 2010; Gehrmann et al., 2019).

In conclusion, we think that it is not yet applicable for developing an interpretative evaluation metric that directly employs and and heavily relies on a supervised semantic parser, such as AMR parsers.

6 Related Work

There has been numerous studies on evaluation metrics for factual information inconsistency, especially in summarization task. Besides FRANK which is used in our study, Maynez et al. (2020), Huang et al. (2020), and Fabbri et al. (2021) all proposed benchmarks for evaluation metrics with different topology of hallucination errors. Laban et al. (2022) and Fabbri et al. (2022) conducted ablation studies on recent summarization evaluation models. But most of these works only focus on one-dimensional quantitative scores. Our model is an complementary work to fully interpret and analyze the semantic features of generated text.

The node alignment algorithm is inspired and modified based on AMR evaluation metrics, Smatch (Cai and Knight, 2013) and SEMA (Anchiêta et al., 2019) which measures the semantic overlap between of two AMR graphs parsed from the same source.

The Pyramid model (Nenkova et al., 2007) provides a complementary interpretative evaluation of semantic difference which is more robust and accurate, but it requires human evaluation. Our work attempts to achieve similar result through an automatic approach.

The works of Yu et al. (2019) and Goyal and Durrett (2020) encoded dependency parsing to output better evaluation scores. Our work can be used in similar ways by encoding the interpretation to generate more characterized for specific tasks.

Zhang and Ji (2021) used parsed AMR graphs in information extraction task through graph encoding and graph conditional decoding. They contributed an interesting and efficient use of semantic information in AMR graphs in a supervised setting.

7 Discussion

We have argued the necessariness of interpretative evaluation metrics for natural language generation tasks and even for any NLP tasks. Besides the model development process, we also think an interpretative evaluator can improve the model training process as an additional loss by pointing out which part of the output contains misinformation to help models have a correct entity awareness and context awareness.

8 Future Work

Reporting inconsistency at sentence level cannot be directly used in any part of model development process, because human can manually read sentences pairs much more efficiently and accurately. Therefore, additional to the interpretation of inconsistencies, we also want to summarize them into some understandable features. This requires annotations about features of NLG models and extensive study on the format of annotated features.

Since there are not many robust semantic parser for malformed system outputs, it is also important to explore other approaches to approximate concept and context overlap without relying on semantic representations.

9 Conclusion

We introduce a new form of interpretative evaluation metric for semantic differences, and propose an implementation as an automatic interpretative evaluator, named FACTDIFF, that based on abstract meaning representation parsing, to generate detailed explanation about where and how system outputs and references are different semantically.

References

Rafael Torres Anchiêta, Marco Antonio Sobrevilla Cabezudo, and Thiago Alexandre Salgueiro Pardo. 2019. Sema: an extended semantic evaluation for amr. In *(To appear) Proceedings of the 20th Computational Linguistics and Intelligent Text Processing*. Springer International Publishg.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page todo, Online. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Alexander R. Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Qafacteval: Improved qa-based factual consistency evaluation for summarization.

Christiane Fellbaum. 2005. Wordnet and wordnets. In Alex Barber, editor, *Encyclopedia of Language and Linguistics*, pages 2–665. Elsevier.

Ramón Fernandez Astudillo, Miguel Ballesteros, Tahira Naseem, Austin Blodgett, and Radu Florian. 2020. Transition-based parsing with stack-transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1001–1007, Online. Association for Computational Linguistics.

J.R. Galliers and K.S. Jones. 1993. *Evaluating Natural Language Processing Systems*. Computer Laboratory Cambridge: Technical report. University of Cambridge, Computer Laboratory.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Dimitra Gkatzia and Saad Mahamood. 2015. A snapshot of NLG evaluation practices 2005 - 2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 57–60, Brighton, UK. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. What have we achieved on text summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, Online. Association for Computational Linguistics.

Martin Kay. 1979. Functional grammar. *Annual Meeting of the Berkeley Linguistics Society*.

Martin Kay. 1984. Functional unification grammar: A formalism for machine translation. pages 75–78.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Philipp Koehn. 2009. *Statistical Machine Translation*. Cambridge University Press.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Chin-Yew Lin and Franz Josef Och. 2004. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 501–507, Geneva, Switzerland. COLING.

Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2019. Compositional semantic parsing across graphbanks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4576–4585, Florence, Italy. Association for Computational Linguistics.

Shuming Ma, Dongdong Zhang, and Ming Zhou. 2020. A simple and effective unified encoder for document-level machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3505–3511, Online. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. 2018. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1191–1204, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4(2):4–es.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. ACL '02, page 311–318, USA. Association for Computational Linguistics.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ehud Reiter and Anja Belz. 2009. An Investigation into the Validity of Some Metrics for Automatically Evaluating Natural Language Generation Systems. *Computational Linguistics*, 35(4):529–558.

Horacio Saggion, Juan-Manuel Torres-Moreno, Iria da Cunha, Eric SanJuan, and Patricia Velázquez-Morales. 2010. Multilingual summarization evaluation without human models. In *Coling 2010: Posters*, pages 1059–1067, Beijing, China. Coling 2010 Organizing Committee.

Xiaoyu Shen, Ernie Chang, Hui Su, Cheng Niu, and Dietrich Klakow. 2020. Neural data-to-text generation via jointly learning the segmentation and correspondence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7155–7165, Online. Association for Computational Linguistics.

Stuart M. Shieber. 1987. An introduction to unification-based approaches to grammar. *Journal of Symbolic Logic*, 52(4):1052–1054.

Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P. Parikh. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *ArXiv*, abs/1910.08684.

Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, Stephanie Schoch, Craig Thomson, and Luou Wen. 2021. Underreporting of errors in NLG output, and what to do about it. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 140–153, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *ICML Deep Learning Workshop, 2015*.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

L. Wittgenstein. 1922. Tractatus logico-philosophicus. *London: Routledge, 1981*.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *arXiv preprint arXiv:2201.05966*.

Hui Yu, Weizhi Xu, Shouxun Lin, and Qun Liu. 2019. Machine translation evaluation metric based on dependency parsing model. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 18(4).

D. Zaefferer. 2019. *Semantic Universals and Universal Semantics*. Groningen-Amsterdam Studies in Semantics. De Gruyter.

Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online. Association for Computational Linguistics.