# Is Shortest Always Best?
# The Role of Brevity in Logic-to-Text Generation

**Eduardo Calò**[α] **Jordi Levy**[ω] **Albert Gatt**[α] **Kees van Deemter**[α]
[α]Utrecht University [ω]IIIA, CSIC
{e.calo,a.gatt,c.j.vandeemter}@uu.nl levy@iiia.csic.es

## Abstract

Some applications of artificial intelligence make it desirable that logical formulae be converted computationally to comprehensible natural language sentences. As there are many logical equivalents to a given formula, finding the most suitable equivalent to be used as input for such a "logic-to-text" generation system is a difficult challenge. In this paper, we focus on the role of brevity: Are the shortest formulae the most suitable? We focus on propositional logic (PL), framing formula minimization (i.e., the problem of finding the shortest equivalent of a given formula) as a Quantified Boolean Formulae (QBFs) satisfiability problem. We experiment with several generators and selection strategies to prune the resulting candidates. We conduct exhaustive automatic and human evaluations of the comprehensibility and fluency of the generated texts. The results suggest that while, in many cases, minimization has a positive impact on the quality of the sentences generated, formula minimization may ultimately not be the best strategy.

https://gitlab.nl4xai.eu/eduardo.calo/brevity-PL

## 1 Introduction

Logical formulae (LFs) are essential for scholars in many scientific fields, such as artificial intelligence and linguistics (e.g., formal semantics). For instance, some explainable artificial intelligence (XAI) methods (e.g., Guidotti et al., 2018) use LFs to provide interpretable and faithful explanations to black-box models. However, one of the drawbacks of these XAI methods is that their output formulae might be complex, hindering their understandability. Grasping the meaning of formulae is also hard for students of logic, especially when they are exposed to formalisms they are not yet accustomed to (Rector et al., 2004). Natural language generation (NLG) methods can be employed to simplify and translate LFs into understandable text in natural languages (NLs), effectively providing explanations for them.

Recently, NLG has made remarkable progress. In particular, in the context of data-to-text generation, good results have been achieved in different domains and datasets, such as sport (e.g., the ROTOWIRE dataset (Wiseman et al., 2017; Thomson et al., 2020)), restaurant (e.g., the E2E Challenge (Smiley et al., 2018; Dušek et al., 2020)), or WebNLG (Gardent et al., 2017). However, the texts produced in these contexts are often relatively poor in logical and rhetorical structure. Moreover, neural language models still fail to encode the semantics of logical formulae (Traylor et al., 2021b) and acquire analytical and deductive logical reasoning capabilities (Ryb et al., 2022). In particular, they struggle with logical connectives, where they fail to differentiate between conjunction and disjunction (Traylor et al., 2021a). Logic-to-text generation thus addresses an area of natural language processing where further progress is much needed.

One way in which the task of generating NL from complex LFs could be facilitated is by simplifying the input. We are interested in understanding the factors that make a formula more or less suitable as an input for a generator. In our work, we focus on brevity. The concept of brevity has long been a topic of linguistic discussion, dating back to at least Grice (1975), where Grice's submaxim of brevity states that shorter utterances should be favored over longer ones, avoiding unnecessary verbosity. Brevity has loomed large in computational accounts of language use as well, especially in the modeling of the human production of referring expressions (see §6 for discussion), a research area known as referring expressions generation (REG). Brevity could also be useful in our situation, in which case a shorter formula, instead of a lengthier logical equivalent, once verbalized using an NLG



*Proceedings of the The 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 180–192

algorithm, might lead to NL sentences that are more fluent and easier to understand.

In this paper, we study the role of brevity in logic-to-text generation, focusing on propositional logic (PL), a formalism for which logical equivalence is decidable. We formulate propositional logic formula minimization (i.e., finding the shortest logical equivalent of a given formula) as a Quantified Boolean Formulae (QBFs) satisfiability problem and employ the algorithm introduced in Calò and Levy (2023) that consistently identifies the shortest equivalents for a given formula.

It is not a foregone conclusion that the shortest formula must always lead to the best verbalization in English. To see this, suppose the input to the generator is of the form $\neg p \vee \neg q$. If the Sheffer stroke, $|$, (i.e., the NAND operator) is a permitted symbol, then the same information can be written more briefly as $p|q$, yet a "direct" verbalization of the former formula (e.g., *Not p or not q*) could well be more comprehensible and fluent than a direct verbalization of the latter (e.g., *It is not the case that p and q*), because the Sheffer stroke does not have a convenient shorthand in English.

Following this line of thought, several questions arise: When verbalizing an input logical formula into English, is it useful to start by finding the shortest formula that is equivalent to the input? Is the resulting text comprehensible to humans? How do we choose among the pool of potential shortest equivalent candidates?

The last question specifically opens up the issue of selecting the optimal translation, given that all potential shortest candidates are exactly of the same length. In our work, we experiment with several deterministic rule-based generators (thus controlling for faithfulness) and a number of selection strategies based on linguistic criteria, ranging from heuristics to neural metrics, to prune the resulting NL candidates. Finally, we conduct comprehensive automatic and human evaluations to assess comprehensibility and fluency of the generated texts.

## 2 Background

**Logical Optimization** Extensive research has been conducted on optimizing complex Boolean expressions, particularly in the field of electronic circuits, where practical considerations (i.e., a more complicated circuit with more logic gates takes up more physical space and produces more heat) make it paramount to find the smallest possible circuit, and hence the shortest possible (i.e., "minimal") formula representing its content. Popular methods for minimization include the Quine-McCluskey algorithm (Quine, 1952, 1955; McCluskey, 1956), Karnaugh maps (Karnaugh, 1953), the Petrick's method (Petrick, 1956), and the Espresso heuristic logic minimizer (Brayton et al., 1982). However, most work has focused on a limited set of canonical forms, such as conjunctive normal form (CNF) or disjunctive normal form (DNF). For our purposes (i.e., studying the interactions between logic and language), we need a general approach where a larger set of connectives and a wider variety of logical structures can be taken into account.

**Quantified Boolean Formulae** Quantified Boolean Formulae (QBFs) are an extension of propositional logic, where universal and existential quantifications over Boolean variables are allowed (Kleine Büning and Bubeck, 2009). Any QBF $\phi$ can be rewritten in a canonical prenex conjunctive normal form (PCNF) without any loss in expressivity, as follows. Let $\mathcal{B}$ be a finite set of Boolean variables, and $\mathcal{Q} = \{\forall, \exists\}$. A QBF $\phi$ over $\mathcal{B}$ in PCNF is given by $\phi := Q_1 B_1 . Q_2 B_2 \ldots Q_n B_n . \psi$, where $Q_i \in \mathcal{Q}$, $B_i \subseteq \mathcal{B}$, and $\psi$ is a Boolean formula over $\mathcal{B}$ in CNF. The part including only quantifiers and bound variables $Q_1 B_1 . Q_2 B_2 \ldots Q_n B_n$ is called the *prefix*, and $\psi$ is called the *matrix*.

The QBF satisfiability problem (Giunchiglia et al., 2009) involves determining the truth of a given QBF $\phi$. For example, given the QBF $\phi := \exists x_1, \ldots, x_n . \forall y_1, \ldots, y_m . \exists z_1, \ldots, z_t . \psi$, $\phi$ is true iff, there exists a truth assignment to $x_1, \ldots, x_n$, such that, for all truth assignments to $y_1, \ldots, y_m$, there exists a truth assignment to $z_1, \ldots, z_t$ such that $\psi$ is true. To solve this problem, several QBF solvers have been developed.[1] Practical applications of QBFs include AI, logic, planning, and games (Cashmore and Fox, 2010; Diptarama and Shinohara, 2016; Shukla et al., 2019). In our study, we utilize QBFs to encode and solve PL formula minimization.

**Logic-to-Text Generation** Logic-to-text generation is the task of generating NL text, starting from a logical formalism (e.g., propositional logic, description logic, or first-order logic). Although the bulk of recent work on NLG (see e.g., Gatt and Krahmer (2018) for a survey) has focused on other areas, generating text from logic nonetheless

[1] http://www.qbflib.org

has a long tradition, with approaches ranging from rule-based methodologies (Wang, 1980; De Roeck and Lowden, 1986; Calder et al., 1989; Shieber et al., 1989; Shemtov, 1996; Carroll and Oepen, 2005; Mpagouli and Hatzilygeroudis, 2009; Coppock and Baxter, 2010; Butler, 2016; Flickinger, 2016; Kasenberg et al., 2019) to statistical (Wong and Mooney, 2007; Lu and Ng, 2011; Basile, 2015) and neural models (Manome et al., 2018; Hajdik et al., 2019; Chen et al., 2020; Liu et al., 2021; Wang et al., 2021; Lu et al., 2022).

One of the complicating factors for this task is the problem of *logical-form equivalence* (Appelt, 1987; Shieber, 1988, 1993), which implies that every logical formula is equivalent to infinitely many other formulae, where the question of whether two formulae are logically equivalent is, in many formalisms (e.g., first-order logic), undecidable. In the present paper, we circumvent this problem by focusing on a decidable fragment of logic, as did, e.g., van Deemter and Halldórsson (2001) and Minock (2014) before us in different ways.

In a closely related work, Calò et al. (2022) manipulate a given first-order formula to obtain logically equivalent simplified versions via logical equivalence laws, yet their algorithm is not guaranteed to return the shortest formula.

## 3 Algorithm

To solve our PL minimization problem, we leverage the QBF-based algorithm presented in Calò and Levy (2023). We define formula length as the number of symbols (i.e., predicates and connectives, parentheses excluded) contained in a formula.

In outline, given (i) a PL formula $\psi$ and (ii) a functionally complete set $\mathcal{C}$ of PL connectives, the algorithm produces the set $\mathcal{P} = \{\psi'_1, \ldots, \psi'_n\}$ of all those PL formulae such that (a) $\psi$ and $\psi'_i$ are logically equivalent, (b) $\psi'_i$ does not contain any connectives that are not members of $\mathcal{C}$, and (c) there does not exist any strictly shorter sentence $\chi$ satisfying (a) and (b).

The strength of the QBF-based algorithm is that it computes a *scheme* $T_n$ of all candidates $\psi'_i$ of length $n$, instead of checking each one of $\psi'_i$ for equivalence with $\psi$. Tseitin transformation (Tseitin, 1983) is used to encode the equivalence of $T_n$ and $\psi$ as a QBF formula, which is checked for satisfiability (see Section 2) by a QBF solver (Tentrup, 2019). The algorithm can find all $\psi'_i \in \mathcal{P}$ of a certain length $n$. By making several calls to the QBF solver, increasing $n$, we make sure that the first found solution is a minimal solution.

The fact that the algorithm computes a unique *scheme* for all candidates of length $n$ makes it very efficient, compared with other straightforward approaches. We refer the reader to Calò and Levy (2023) for details on the implementation.

## 4 Experiments

Our experimentation strategy can be summarized as follows: (i) we simplify the input formulae using the algorithm described in §3, (ii) we realize all the outputs using different generators, and (iii) we prune the resulting candidate realizations using a number of selection strategies.

We use three rule-based generators: (i) a BASELINE, (ii) the system presented in Ranta (2011), and (iii) LOLA (Calò et al., 2022). BASELINE is a system that generates near-literal translations of the formulae. Ranta performs some syntactic optimization (e.g., flattening, aggregation, etc.) to improve fluency. LOLA is an extension of Ranta that performs heuristic logical optimization based on standard equivalence laws to the input formula before verbalizing it. The generators were evaluated for faithfulness (i.e., whether the generated text conveys all and only the information of the input formula) in Calò et al. (2022) and shown to guarantee faithful translations. We refer the reader to Ranta (2011) and Calò et al. (2022) for more details on the systems.

For pruning, we experiment with the following five selection strategies: (i) length in number of words, (ii) pseudo-perplexity using BERT (Devlin et al., 2019), (iii) pseudo-SLOR using BERT, (iv) perplexity (PPL) using GPT-2 (Radford et al., 2019), (v) SLOR using GPT-2.[2] SLOR (Syntactic Log-Odds Ratio; Pauls and Klein, 2012; Kann et al., 2018) is a metric based on negative log-likelihood that penalizes highly probable unigrams. In detail, the score given by SLOR consists of the log probability of a sentence under a given language model, normalized by unigram log probability and sentence length. The intuition behind the normalizations is that a rare token should not bring down the sentence's score and shorter sentences should not be preferred over equally fluent longer ones. In our case, this should help us make fairer comparisons, as the length of the sentences generated by the re-

[2]We use `bert-large-cased` and `gpt-2-large`, respectively.

alizers varies considerably, and logical variables and constants (e.g., $x$, $y$, etc., which a language model treats as unigrams), which appear regularly in our sentences, have a unigram probability much higher than the other tokens in the lexicon. We compute PPL and SLOR with BERT, following the methodologies described in Salazar et al. (2020) and Lau et al. (2020) for masked language models.

For the experiments, we consider the Grade Grinder Corpus (GGC; Barker-Plummer et al., 2011), a parallel corpus where each NL sentence is paired with multiple logically equivalent formulae. We retrieve all PL formulae that are parsable by the generators we use. We first simplify the formulae using the algorithm described in §3 and obtain, for each formula, a set of logical equivalents, maximally reduced in terms of length. Out of 1092 PL formulae, 680 got simplified; the others were already in their shortest form. Table 1 shows some descriptive statistics. As a concrete example, starting from the following GGC formula containing 10 symbols:

$$(Tet(a) \wedge Tet(c)) \rightarrow \neg(\neg Large(a) \wedge \neg Large(c))$$

we end up with these shortest equivalents, with the number of symbols reduced to 7:[3]

$$Large(a) \vee ((Tet(c) \wedge Tet(a)) \rightarrow Large(c))$$
$$(Tet(c) \rightarrow Large(c)) \vee (Tet(a) \rightarrow Large(a))$$
$$(Tet(c) \wedge Tet(a)) \rightarrow (Large(c) \vee Large(a))$$
$$\ldots$$

| | $\mu$ | $\sigma$ | Min. | Max. |
|---|---|---|---|---|
| Original | 7.12 | 2.62 | 1 | 18 |
| Minimized | 5.52 | 1.97 | 1 | 11 |

Table 1: Statistics on the length of the GGC formulae before and after minimization.

We proceed with translating the resulting formulae into English using the three rule-based generators. Additionally, we also translate the original GGC formula with the three generators.

At the logic level, all potential candidates are exactly of the same length. Therefore, once NL sentences are generated, we prune the candidates by (i) scoring them using the five selection strategies mentioned above, (ii) selecting the one with the lowest score for each strategy. After this process, for each GGC input formula, we end up with 18 realizations: 15 after the pruning process (3 realizers $\times$ 5 selection strategies), plus 3 from translating the original GGC formula with the three realizers. Table 2 presents some examples.

## 5 Evaluation

### 5.1 Automatic Evaluation

We set up an automatic evaluation comparing the translations by the 18 systems presented in §4 vs. the ground truth NL references associated with the original input formulae in the GGC. We use six automatic metrics, three of which are based on $n-$gram overlap, namely, BLEU (Papineni et al., 2002),[4] METEOR (Banerjee and Lavie, 2005), and ROUGE-L (Lin, 2004), and three on BERT, namely, BERTScore (Zhang et al., 2020),[5] BLEURT (Sellam et al., 2020), a learned metric based on human ratings,[6] and SBERT (Reimers and Gurevych, 2019).[7] For all the metrics except SBERT, we use the implementations provided by `HuggingFace` (Wolf et al., 2020).[8] Table 3 summarizes the results obtained.

Several trends emerge from analyzing the table. The results hint that formula minimization generally improves the translations, as the scores (particularly $n-$gram-based metrics) for the systems that get the minimized versions of the formulae as input are generally higher than the others. Different selection strategies score very similarly, sometimes with negligible differences. The difference in behavior between semantics-based and $n-$gram-based metrics corroborates the findings in Calò et al. (2022). Excluding BLEURT, whose low results are probably due to the nature of the data on which it was pre-trained and the lack of fine-tuning on our side, the results of semantics-based metrics are comparable across the systems, especially when it comes to BERTScore. This can be seen as a confirmation that the generated texts are paraphrases of the GGC ground truth references. However, BERTScore's results need to be taken with a grain of salt, since BERT-like models are known for missing semantic

[3] We list just some of the equivalents, as the algorithm returns many more formulae of length 7 in the actual output.

[4] We adopt the SacreBLEU (Post, 2018) implementation for improved reproducibility.

[5] We use the model `roberta-large_L17_no-idf`.

[6] We use the model `bleurt-base-128` without fine-tuning.

[7] We compute cosine similarity after obtaining sentence embeddings with the model `all-distilroberta-v1`.

[8] https://huggingface.co/evaluate-metric

| System + Selection Strategy | Translation |
|---|---|
| Orig. BASELINE | *If f is large, then f is a cube or if f is large, then f is a dodecahedron.* |
| Orig. Ranta | *At least one of these holds: - if f is large, then f is a cube - if f is large, then f is a dodecahedron.* |
| Orig. LOLA | *f is not large, f is a cube, f is not large or f is a dodecahedron.* |
| Minim. BASELINE BERT SLOR | *f is a dodecahedron or if f is large, then f is a cube.* |
| Minim. Ranta GPT PPL | *If f is large, then f is a cube or f is a dodecahedron.* |
| Minim. LOLA Length | *f is not large, f is a cube or f is a dodecahedron.* |

Table 2: Some translations from the original formula $(Large(f) \rightarrow Cube(f)) \vee (Large(f) \rightarrow Dodec(f))$.

| System + Selection Strategy | $n-$gram-based Metrics | | | Semantics-based Metrics | | |
|---|---|---|---|---|---|---|
| | METEOR | ROUGE-L | SacreBLEU | BERTScore | BLEURT | SBERT |
| Orig. BASELINE | 0.5514 | 0.4386 | 10.8638 | 0.9051 | −0.0916 | 0.7819 |
| Orig. Ranta | 0.5654 | 0.4697 | 12.1577 | 0.9082 | −0.1099 | 0.7464 |
| Orig. LOLA | 0.5655 | 0.5012 | 14.0046 | 0.9115 | −0.0492 | 0.7672 |
| Minim. BASELINE Length | 0.5639 | 0.4955 | 14.7115 | 0.9129 | −0.0131 | **0.7935** |
| Minim. BASELINE BERT PPL | 0.5677 | 0.4977 | 14.8686 | 0.9123 | 0.0028 | 0.7928 |
| Minim. BASELINE BERT SLOR | 0.5652 | 0.4980 | 14.8386 | 0.9129 | −0.0151 | 0.7926 |
| Minim. BASELINE GPT PPL | 0.5752 | 0.4999 | 14.9506 | 0.9122 | −0.0017 | 0.7916 |
| Minim. BASELINE GPT SLOR | 0.5717 | 0.5077 | 14.5815 | **0.9136** | 0.0008 | **0.7935** |
| Minim. Ranta Length | 0.5802 | 0.5037 | 15.5895 | 0.9130 | 0.0126 | 0.7794 |
| Minim. Ranta BERT PPL | 0.5807 | **0.5143** | 15.1632 | 0.9123 | 0.0077 | 0.7720 |
| Minim. Ranta BERT SLOR | **0.5865** | 0.5099 | **15.6208** | 0.9132 | 0.0117 | 0.7797 |
| Minim. Ranta GPT PPL | 0.5759 | 0.5020 | 15.3613 | 0.9120 | 0.0092 | 0.7720 |
| Minim. Ranta GPT SLOR | 0.5780 | 0.5005 | 15.3833 | 0.9132 | **0.0141** | 0.7792 |
| Minim. LOLA Length | 0.5722 | 0.5050 | 15.5769 | 0.9133 | −0.0005 | 0.7805 |
| Minim. LOLA BERT PPL | 0.5771 | 0.5137 | 15.2999 | 0.9134 | −0.0028 | 0.7769 |
| Minim. LOLA BERT SLOR | 0.5811 | 0.5131 | 15.3199 | 0.9133 | −0.0006 | 0.7800 |
| Minim. LOLA GPT PPL | 0.5689 | 0.4995 | 15.0712 | 0.9132 | 0.0048 | 0.7756 |
| Minim. LOLA GPT SLOR | 0.5709 | 0.4967 | 15.3573 | 0.9132 | −0.0107 | 0.7799 |

Table 3: Performance of the 18 systems against the GGC ground truth references according to the automatic metrics.

nuances, such as negation (Ettinger, 2020), which is crucial for evaluating our task.

### 5.2 Human Evaluation

We conduct a human evaluation to understand the impact of formulae minimization on the translations. We recruit a group of 42 human evaluators and ask them to give feedback on (i) **comprehensibility** (i.e., whether the message conveyed by the sentence is understandable and not open to multiple interpretations), and (ii) **fluency** (i.e., whether the sentence sounds like a natural English sentence and is grammatically correct). These are central requirements to look for, as text generated from logic can be extremely disfluent and incomprehensible (e.g., a literal translation from a formula), while still being faithful to the input.

Evaluators are asked to rate the comprehensibility and fluency of each translation on a $7-$point Likert scale (Likert, 1932). If comprehensibility receives a score $< 4$, participants are asked to give the motivations for which the sentence is hard to understand (i.e., ambiguity, complexity, or length of the sentence, or other). See Appendix A for more information on how we conduct the evaluation and the instructions given to the evaluators.

We sample 48 references from the GGC and select translations of the corresponding formula by 6 systems. The systems we choose are Orig. BASELINE, Orig. Ranta, and Orig. LOLA and Minim. BASELINE BERT SLOR, Minim. Ranta BERT SLOR, and Minim. LOLA BERT SLOR (henceforth, Minim. BBS, Minim. RBS, and Minim. LBS, respectively; see §4). Among the minimized variants, we choose BERT SLOR for two reasons: (i) BERT-based scoring seems to perform slightly better than the other selection strategies (see §5.1), and (ii) given that SLOR and PPL scores are nearly identical across systems, we opt for SLOR for theoretical reasons (see §4). After the selection, we end up with a total of 48 (references) $\times$ 6 (systems) $=$ 288 experimental items.

Participants and experimental items are randomly assigned to one of six groups and rotated through a 6 (systems) $\times$ 6 (participant groups) Latin square (Fisher, 1925). This guarantees that every item is shown to approximately the same number of participants, that every participant is

shown the same number of items (48), and that participants only see one system translation per original formula.

### 5.3 Results

The overall comprehensibility and overall fluency of each translation are computed as the means of the ratings given by the evaluators on the two dimensions. The inter-annotator agreements for both dimensions are low (comprehensibility: Krippendorff's $\alpha = 0.329$; fluency: Krippendorff's $\alpha = 0.282$). We find a very strong positive correlation between the two dimensions (Pearson's $r = 0.89; p \ll 0.001$), indicating that more fluent translations are also more comprehensible.

Figure 1 shows the boxplot with the distribution of the ratings on comprehensibility and fluency for all the systems. The translations from Minim. RBS receive the highest mean on both comprehensibility ($\mu = 5.19$) and fluency ($\mu = 4.89$). One-way ANOVA analyses reveal that for both comprehensibility and fluency, the differences between systems are statistically significant (comprehensibility: $F(5, 282) = 21.72; p \ll 0.001$; fluency: $F(5, 282) = 13.39; p \ll 0.001$).

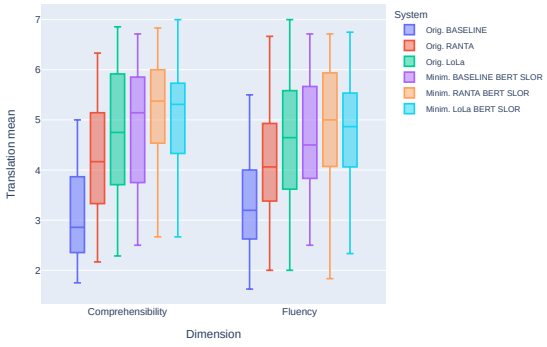


Figure 1: Boxplot with the distribution of translations' mean ratings across systems, for both comprehensibility and fluency.

Tukey's HSD tests for multiple comparisons show comparable results on the two dimensions. In general, Orig. LoLa is not significantly different from all the minimized variants. This suggests that human evaluators did not perceive a difference between settings where the input was manipulated using equivalence laws (*à la* LoLa) and settings where QBF minimization was used. Moreover, the tests show that all the minimized variants do not significantly differ from each other. This may be an indication that formula minimization plays an important role beforehand and the choice of the realizer used for translation does not matter much. Lastly, we notice less variance when BERT SLOR variants are involved, especially in comprehensibility with Ranta and LoLa.

We compute correlations between the human ratings and the score assigned by BERTScore, ROUGE-L, and SBERT to the questions rated by the evaluators, for both comprehensibility and fluency. Figure 2 shows the scatterplots and Table 4 the numerical results. The results are comparable across the two dimensions and we find low, but statistically significant positive correlations with human judgments on both comprehensibility and fluency.

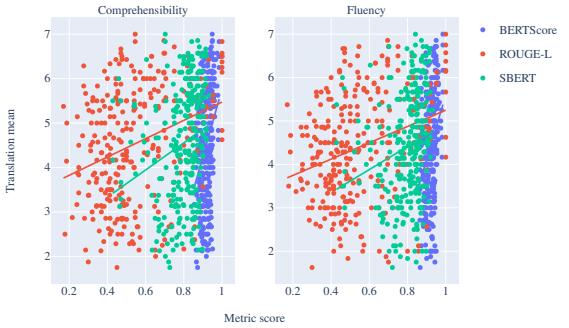


Figure 2: Scatterplots with the correlations between translations' mean ratings on comprehensibility and fluency and scores assigned by the automatic metrics.

| | BERTScore | ROUGE-L | SBERT |
|---|---|---|---|
| Comprehensibility | 0.331 | 0.315 | 0.223 |
| Fluency | 0.342 | 0.302 | 0.205 |

Table 4: Correlations between human ratings and automatic metrics on comprehensibility and fluency. All results are computed using Pearson's $r$ and are statistically significant ($p < 0.001$).

We shed some light on the reasons why certain translations achieve low comprehensibility by inspecting the responses to the follow-up questions that were presented when comprehensibility is rated poorly (see §5.2). In most cases, low intelligibility corresponds to ambiguities detected in the translation (selected 330 times). Next comes the complexity of the linguistic structure (306), and finally the length of the translation (110). Other reasons are also chosen (118). We break down in Table 5 the detailed figures per system. We clearly

| | Orig. BASELINE | Orig. Ranta | Orig. LOLA | Minim. BBS | Minim. RBS | Minim. LBS |
|---|---|---|---|---|---|---|
| Ambiguity | 119 | 56 | 48 | 39 | 33 | 35 |
| Complexity | 110 | 58 | 46 | 38 | 28 | 26 |
| Length | 46 | 13 | 11 | 14 | 7 | 9 |
| Other | 33 | 26 | 14 | 15 | 12 | 18 |

Table 5: Figures for the reasons of translations' low comprehensibility per system.

| | |
|---|---|
| Sentence | *If b is a tetrahedron, then b is a tetrahedron and it is not the case that c is a tetrahedron.* |
| Interpretation 1 | *(If b is a tetrahedron, then b is a tetrahedron) and (it is not the case that c is a tetrahedron).* |
| Interpretation 2 | *If (b is a tetrahedron), then (b is a tetrahedron and it is not the case that c is a tetrahedron).* |
| Original Formula | $Tet(b) \rightarrow (Tet(b) \wedge \neg Tet(c))$ |

Table 6: An ambiguous translation and its possible interpretations.

notice that manipulating the input formula helps improve the comprehensibility of the sentences, as the number of problematic cases decreases with LOLA and the minimized variants. We proceed with a manual check of the translations and report some interesting cases.

A noteworthy example of ambiguity is presented in Table 6. The sentence can have (at least) two interpretations. We need to resort to the original formula in the GGC to disambiguate the sentence and retrieve the intended meaning, which corresponds to the second interpretation. The sentence is generated by Orig. BASELINE. Other systems greatly improve the translation's comprehensibility, e.g., the corresponding translation by Minim. LBS *b is not a tetrahedron or c is not a tetrahedron* is rated much higher by the evaluators ($\mu = 4.43$ vs. $\mu = 2.86$).

Problematic cases pertaining to the complexity and length of the sentence include those presenting bulleted lists. Evaluators are ambivalent about their use: Some systematically give high scores to sentences containing bulleted lists, while others severely criticize them. One example that particularly baffled the evaluators is the following, as it contains nested levels of indentation:

*At least one of these holds:*
- *d is a dodecahedron and d is small*
- *all these hold:*
  - *d is not a dodecahedron or d is not small*
  - *a is small*

Further, we inspect in which circumstances formula minimization leads to better translations. We consider the scores on comprehensibility[9] of the translations by Orig. BASELINE vs. Minim. BBS, and Orig. Ranta vs. Minim. RBS. We select the top 10 instances where the score difference is the highest. We do not consider LOLA to keep the analysis controlled, as LOLA performs further logical manipulation before verbalization. We manually inspect the original and minimized formulae, and find out, unsurprisingly, that the outputs are improved mostly thanks to redundancy removal (e.g., repeated predicates and double negation) from the input. As an example, the GGC formula $\neg(BackOf(c, a) \rightarrow \neg FrontOf(c, e)) \wedge FrontOf(c, e)$ gets translated by Orig. Ranta as *It is not the case that if c is in back of a, then c is not in front of e and c is in front of e*. After minimization, the resulting formula $FrontOf(c, e) \wedge BackOf(c, a)$ gets translated by Minim. RBS as *c is in front of e and in back of a*, gaining 2.71 points in comprehensibility.

## 6 Conclusion

We have studied the role of brevity in logic-to-text generation. We employed a state-of-the-art (in terms of speed) QBF-based algorithm (Calò and Levy, 2023) that always finds the shortest equivalents to an input PL formula. We verbalized the outputs experimenting with several realizers and selection strategies to study whether the translations from shorter formulae are more comprehensible and fluent than those from their longer logically equivalent counterparts.

The results of our evaluations suggest that manipulating the original input formula (using logical equivalence laws as in LOLA or via minimization) improves the sentences generated. Our study taught us some other lessons as well. For example, the free text comments that our evaluators provided suggest

[9]We get similar results when we look at fluency.

that there is a need to (i) take measures to mitigate ambiguity in the generated sentences (see also Table 5 and Table 6), and (ii) further improve fluency, despite the fact that both Ranta and LOLA already take some measures to do it (the former performing syntactic optimizations and the latter performing both syntactic and logical optimizations).

In conclusion, is brevity valid as a principle that guides logic-to-text generation? A comparison with referring expressions generation (REG) might be helpful here. Researchers in REG build computational models of the choices that human speakers make when referring. Early REG algorithms (Dale, 1992) always generated the shortest expression that singles out the intended referent. However, such brevity-oriented REG algorithms have been found both computationally infeasible (Dale and Reiter, 1995) and dissimilar to the approaches followed by human speakers. Recent REG models all strike a compromise between brevity and a number of other factors (Van Deemter, 2016); they can be seen as approximating brevity to different degrees. It is conceivable, likewise, that future work on logic-to-text generation ends up following a similar pattern. For example, although the results reported in this paper suggest that brevity has a role to play, future logic-to-text algorithms might achieve even better performance by deviating from brevity to some extent. Perhaps brevity in logic-to-text generation should be weighed less heavily in some communicative situations, just as speakers are known to generate more elaborate referring expressions when referential situations are complex (Koolen et al., 2011; Paraboni and van Deemter, 2014).

We hope that future research, in which logical formulae and their natural language "translations" are embedded in well-understood practical tasks, for example in logic teaching or XAI, may shed further light on these questions.

## Limitations

In the present paper, we have concentrated on PL. The first natural extension of this work would be to see if the QBF-based algorithm (or similar methods) could scale up to other (more expressive) formalisms, e.g., first-order logic. This would open up a range of interesting research questions, as in first-order logic, equivalence is in general undecidable. As a first step, an approach based on the use of a first-order theorem prover (e.g., VAMPIRE (Riazanov and Voronkov, 2002)) to check logical equivalence could be explored. This would not guarantee total coverage but might handle the vast majority of cases.

Our work has focused on four common logical operators, i.e., negation, conjunction, disjunction, and implication. When including other operators, such as the biconditional or the Sheffer stroke, the results could differ. For example, given that the Sheffer stroke is functionally complete on its own, we could have very short formulae but that may result in incomprehensible or disfluent texts.

Our conclusions are drawn from a limited number of realizers sharing similar properties (i.e., all of them are rule-based and derived from the system originally presented in Ranta (2011)). On the other hand, because of this, we were able to perform controlled generation and zoom in on the impact of minimization, which would not be straightforward in other settings, e.g., neural.

Moreover, we have only tackled English as NL. Brevity is drastically language-dependent and experimenting with other (especially typologically diverse) languages could bring different results.

Finally, the evaluation process could be further refined, as hinted by some comments we received in the human evaluation. For instance, some suggest that working within practical domains, especially with the help of pictures, would have eased the work of the evaluators.

## Acknowledgments

We thank the anonymous reviewers for their careful reading of our manuscript and their insightful comments and suggestions.

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement № 860621.

## References

Douglas E. Appelt. 1987. Bidirectional grammars and the design of natural language generation systems. In *Theoretical Issues in Natural Language Processing 3*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Transla-*

*tion and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Dave Barker-Plummer, Richard Cox, and Robert Dale. 2011. Student translations of natural language into logic: the Grade Grinder Corpus release 1.0. In *Proceedings of the 4th International Conference on Educational Data Mining*, pages 51–60.

Valerio Basile. 2015. *From logic to language: Natural language generation from logical forms*. Ph.D. thesis, University of Groningen.

Robert K. Brayton, Gary D. Hachtel, Lane A. Hemachandra, A. Richard Newton, and Alberto Luigi M. Sangiovanni-Vincentelli. 1982. A comparison of logic minimization strategies using espresso: An apl program package for partitioned logic minimization. In *Proceedings of the International Symposium on Circuits and Systems*, pages 42–48.

Alastair Butler. 2016. Deterministic natural language generation from meaning representations for machine translation. In *Proceedings of the 2nd Workshop on Semantics-Driven Machine Translation (SedMT 2016)*, pages 1–9, San Diego, California. Association for Computational Linguistics.

Jonathan Calder, Mike Reape, and Henk Zeevat. 1989. An algorithm for generation in unification categorial grammar. In *Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, England. Association for Computational Linguistics.

Eduardo Calò, Elze van der Werf, Albert Gatt, and Kees van Deemter. 2022. Enhancing and evaluating the grammatical framework approach to logic-to-text generation. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 148–171, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Eduardo Calò and Jordi Levy. 2023. General boolean formula minimization with QBF solvers.

John Carroll and Stephan Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *Second International Joint Conference on Natural Language Processing: Full Papers*.

Michael Cashmore and Maria Fox. 2010. Planning as qbf. In *International Conference on Automated Planning and Scheduling Doctoral Consortium (ICAPS 2010)*.

Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyou Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020. Logic2Text: High-fidelity natural language generation from logical forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2096–2111, Online. Association for Computational Linguistics.

Elizabeth Coppock and David Baxter. 2010. A translation from logic to english with dynamic semantics. In *New Frontiers in Artificial Intelligence*, pages 197–216, Berlin, Heidelberg. Springer Berlin Heidelberg.

Robert Dale. 1992. *Generating referring expressions: Constructing descriptions in a domain of objects and processes.* The MIT Press.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive science*, 19(2):233–263.

A.N. De Roeck and B.G.T. Lowden. 1986. Generating English paraphrases from formal relational calculus expressions. In *Coling 1986 Volume 1: The 11th International Conference on Computational Linguistics*.

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

Ryo Yoshinaka Diptarama and Ayumi Shinohara. 2016. Qbf encoding of generalized tic-tac-toe. In *4th International Workshop on Quantified Boolean Formulas (QBF) Co-located with 19th International Conference on Theory and Applications of Satisfiability Testing (SAT), Bordeaux, France*, pages 14–26.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.

Allyson Ettinger. 2020. What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Ronald Aylmer Fisher. 1925. *Statistical methods for research workers*. Edinburgh, Scotland: Oliver and Loyd.

Dan Flickinger. 2016. Generating English paraphrases from logic. *From Semantics to Dialectometry*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Enrico Giunchiglia, Paolo Marin, and Massimo Narizzano. 2009. Reasoning with quantified boolean formulas. In *Handbook of satisfiability*, pages 761–780. IOS Press.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local rule-based explanations of black box decision systems.

Valerie Hajdik, Jan Buys, Michael Wayne Goodman, and Emily M. Bender. 2019. Neural text generation from rich semantic representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2259–2266, Minneapolis, Minnesota. Association for Computational Linguistics.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323, Brussels, Belgium. Association for Computational Linguistics.

Maurice Karnaugh. 1953. The map method for synthesis of combinational logic circuits. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, 72(5):593–599.

Daniel Kasenberg, Antonio Roque, Ravenna Thielstrom, Meia Chita-Tegmark, and Matthias Scheutz. 2019. Generating justifications for norm-related agent decisions. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 484–493, Tokyo, Japan. Association for Computational Linguistics.

Hans Kleine Büning and Uwe Bubeck. 2009. Theory of quantified boolean formulas. In *Handbook of satisfiability*, pages 735–760. IOS Press.

Ruud Koolen, Albert Gatt, Martijn Goudbeek, and Emiel Krahmer. 2011. Factors causing referential overspecification in definite descriptions. *Journal of Pragmatics*, 43(13):3231–3250. Factors causing referential overspecification in definite descriptions Pagination: 20.

Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. 2020. How furiously can colorless green ideas sleep? sentence acceptability in context. *Transactions of the Association for Computational Linguistics*, 8:296–310.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

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

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

Wei Lu and Hwee Tou Ng. 2011. A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1611–1622, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Xuantao Lu, Jingping Liu, Zhouhong Gu, Hanwen Tong, Chenhao Xie, Junyang Huang, Yanghua Xiao, and Wenguang Wang. 2022. Parsing natural language into propositional and first-order logic with dual reinforcement learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5419–5431, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Kana Manome, Masashi Yoshikawa, Hitomi Yanaka, Pascual Martínez-Gómez, Koji Mineshima, and Daisuke Bekki. 2018. Neural sentence generation from formal semantics. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 408–414, Tilburg University, The Netherlands. Association for Computational Linguistics.

Edward Joseph McCluskey. 1956. Minimization of boolean functions. *The Bell System Technical Journal*, 35(6):1417–1444.

Michael Minock. 2014. In pursuit of decidable'logical form'. In *The Fifth Swedish Language Technology Conference, 13-14 November 2014, Uppsala*.

Aikaterini Mpagouli and Ioannis Hatzilygeroudis. 2009. A Knowledge-based System for Translating FOL Formulas into NL Sentences. In Iliadis, Maglogiann, Tsoumakasis, Vlahavas, and Bramer, editors, *Artificial Intelligence Applications and Innovations III*, volume 296, pages 157–163. Springer US, Boston, MA. Series Title: IFIP Advances in Information and Communication Technology.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ivandré Paraboni and Kees van Deemter. 2014. Reference and the facilitation of search in spatial domains. *Language, Cognition and Neuroscience*, 29(8):1002–1017.

Adam Pauls and Dan Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 959–968, Jeju Island, Korea. Association for Computational Linguistics.

Stanley R. Petrick. 1956. A direct determination of the irredundant forms of a boolean function from the set of prime implicants. *Air Force Cambridge Res. Center Tech. Report*, pages 56–110.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Willard V. Quine. 1952. The problem of simplifying truth functions. *The American Mathematical Monthly*, 59(8):521–531.

Willard V. Quine. 1955. A way to simplify truth functions. *The American Mathematical Monthly*, 62(9):627–631.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Aarne Ranta. 2011. Translating between language and logic: what is easy and what is difficult. In *Proceedings of the International Conference on Automated Deduction*, pages 5–25. Springer.

Alan Rector, Nick Drummond, Matthew Horridge, Jeremy Rogers, Holger Knublauch, Robert Stevens, Hai Wang, and Chris Wroe. 2004. Owl pizzas: Practical experience of teaching owl-dl: Common errors & common patterns. In *Engineering Knowledge in the Age of the Semantic Web: 14th International Conference, EKAW 2004, Whittlebury Hall, UK, October 5-8, 2004. Proceedings 14*, pages 63–81. Springer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Alexandre Riazanov and Andrei Voronkov. 2002. The design and implementation of vampire. *AI communications*, 15(2-3):91–110.

Samuel Ryb, Mario Giulianelli, Arabella Sinclair, and Raquel Fernández. 2022. AnaLog: Testing analytical and deductive logic learnability in language models. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 55–68, Seattle, Washington. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Hadar Shemtov. 1996. Generation of paraphrases from ambiguous logical forms. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Stuart M. Shieber. 1988. A uniform architecture for parsing and generation. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*.

Stuart M. Shieber. 1993. The problem of logical form equivalence. *Computational Linguistics*, 19(1):179–190.

Stuart M. Shieber, Gertjan van Noord, Robert C. Moore, and Fernando C. N. Pereira. 1989. A semantic-head-driven generation algorithm for unification-based formalisms. In *27th Annual Meeting of the Association for Computational Linguistics*, pages 7–17, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Ankit Shukla, Armin Biere, Luca Pulina, and Martina Seidl. 2019. A Survey on Applications of Quantified Boolean Formulas. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 78–84, Portland, OR, USA. IEEE.

Charese Smiley, Elnaz Davoodi, Dezhao Song, and Frank Schilder. 2018. The E2E NLG challenge: A tale of two systems. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 472–477, Tilburg University, The Netherlands. Association for Computational Linguistics.

Leander Tentrup. 2019. CAQE and QuAbS: Abstraction Based QBF Solvers. *Journal on Satisfiability, Boolean Modeling and Computation*, 11:155–210.

Craig Thomson, Ehud Reiter, and Somayajulu Sripada. 2020. SportSett:basketball - a robust and maintainable data-set for natural language generation. In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*, pages 32–40, Santiago de Compostela, Spain. Association for Computational Lingustics.

Aaron Traylor, Roman Feiman, and Ellie Pavlick. 2021a. AND does not mean OR: Using formal languages to study language models' representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International*

*Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 158–167, Online. Association for Computational Linguistics.

Aaron Traylor, Ellie Pavlick, and Roman Feiman. 2021b. Transferring representations of logical connectives. In *Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA)*, pages 22–25, Groningen, the Netherlands (online). Association for Computational Linguistics.

Grigori Samuilovitsch Tseitin. 1983. On the Complexity of Derivation in Propositional Calculus. In Jörg H. Siekmann and Graham Wrightson, editors, *Automation of Reasoning: 2: Classical Papers on Computational Logic 1967–1970*, pages 466–483. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kees Van Deemter. 2016. *Computational models of referring: a study in cognitive science*. MIT Press.

Kees van Deemter and Magnús M. Halldórsson. 2001. Logical form equivalence: the case of referring expressions generation. In *Proceedings of the ACL 2001 Eighth European Workshop on Natural Language Generation (EWNLG)*, Toulouse, France. Association for Computational Linguistics.

Chunliu Wang, Rik van Noord, Arianna Bisazza, and Johan Bos. 2021. Evaluating text generation from discourse representation structures. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 73–83, Online. Association for Computational Linguistics.

Juen-tin Wang. 1980. On computational sentence generation from logical form. In *COLING 1980 Volume 1: The 8th International Conference on Computational Linguistics*.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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.

Yuk Wah Wong and Raymond Mooney. 2007. Generation by inverting a semantic parser that uses statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 172–179, Rochester, New York. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

## A Details on Human Evaluation

We conduct the human evaluation via Prolific.[10] The 42 evaluators we recruit are all native speakers of English and completed at least high school. They are paid £3 for an estimated workload of 20 minutes. Figure 3 presents the instructions provided to the evaluators and an example sentence.

[10] https://www.prolific.co/

Thank you very much for participating in this experiment!

It will take approximately 20 minutes to fill in this survey. If you do wish to participate, your response will be handled anonymously: The information in this study will only be used in ways that will not reveal who you are. You will not be identified in any publication from this study or in any data files shared with other researchers. Your participation in this study is confidential. If at any point you would like to stop, you can close this form and your response will be deleted.

I have read the above information and understand the purpose of the research and that data will be collected from me. I agree that data gathered for the study may be published or made available, provided my name or other identifying information is not used.

◯ I confirm this.
◯ I do not confirm this and I want to withdraw from participation.

---

The purpose of the experiment is to assess the quality of some automatically generated English sentences concerning geometrical shapes and their properties. We are interested in receiving feedback on (i) comprehensibility (i.e., do you understand precisely the message conveyed by the sentence?), and (ii) fluency (i.e., does the sentence sound natural to you?).

We will present to you 48 sentences, and for each, we would like to know your feedback on the aforementioned aspects. In detail, you will have to answer the following questions:

1. How **comprehensible** is the sentence? By a comprehensible sentence, we mean that it is understandable and does not have multiple interpretations.
2. How **fluent** is the sentence? By a fluent sentence, we mean that it sounds like a natural English sentence and is grammatically correct.

Please, note down the definitions of comprehensibility and fluency, in case you want to refer to them later.

---

Here's an example:

Sentence:
*If it is not the case that c is a cube, then a is a tetrahedron, and if it is not the case that d is a cube, then b is a cube.*

Comprehensibility
①②③④⑤⑥⑦
Fluency
①②③④⑤⑥⑦

Why do you think that the sentence is hard to understand?
(In the real questionnaire, this appears only if comprehensibility $< 4$)
Note: with 'the sentence is ambiguous', we mean 'the sentence has multiple meanings'.

◯ The sentence is ambiguous
◯ The sentence is too long
◯ The language structure is too complex
◯ Other: ☐

Now it is your turn!

---

Figure 3: The instructions provided to the evaluators during our human evaluation.