NeuroStructural Decoding: Neural Text Generation with Structural Constraints

Mohaddeseh Bastan
Stony Brook University
mbastan@cs.stonybrook.edu

Mihai Surdeanu
University of Arizona
msurdeanu@email.arizona.edu

Niranjan Balasubramanian
Stony Brook University
niranjan@cs.stonybrook.edu

Abstract

Text generation often involves producing texts that also satisfy a given set of semantic constraints. While most approaches for conditional text generation have primarily focused on lexical constraints, they often struggle to effectively incorporate syntactic constraints, which provide a richer language for approximating semantic constraints. We address this gap by introducing NeuroStructural Decoding, a new decoding algorithm that incorporates syntactic constraints to further improve the quality of the generated text. We build NeuroStructural Decoding on the NeuroLogic Decoding (Lu et al., 2021b) algorithm, which enables language generation models to produce fluent text while satisfying complex lexical constraints. Our algorithm is powerful and scalable. It tracks lexico-syntactic constraints (e.g., we need to observe *dog* as subject and *ball* as object) during decoding by parsing the partial generations at each step. To this end, we adapt a dependency parser to generate parses for incomplete sentences. Our approach is evaluated on three different language generation tasks, and the results show improved performance in both lexical and syntactic metrics compared to previous methods. The results suggest this is a promising solution for integrating fine-grained controllable text generation into the conventional beam search decoding[1].

1 Introduction

Controllable text generation uses decoding algorithms that support lexico-semantic constraints to control what is included in the output. Checking for the presence or absence of certain words has been shown to improve applications such as recipe generation (Lu et al., 2021b), sentiment analysis (Howard et al., 2022), and predicting implicit consequences of beliefs (Alexeeva et al., 2022).

[1] Code and data is available at https://stonybrooknlp.github.io/NeuroStructuralDecoding

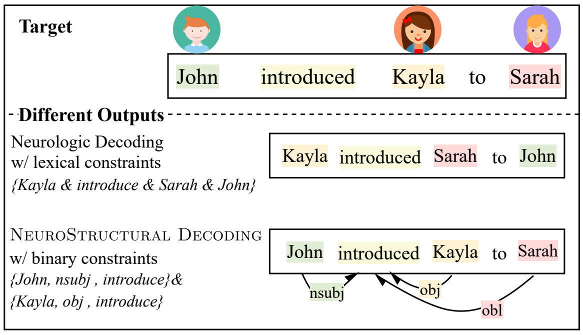

Figure 1: An example that compares the output produced by Neurologic Decoding with lexical constraints alone vs. the output generated by NeuroStructural Decoding with lexico-syntactic constraints.

Most of the current work in constrained generation focuses on handling *lexical* constraints (Miao et al., 2019; Lu et al., 2021b; Guo and Roth, 2021; Turcan et al., 2022). However, lexical constraints alone cannot directly support more complex semantic constraints such as the presence/absence of relations or events between multiple concepts. Consider the simple example in Figure 1, in which target sentence requires that different people have specific roles in an event. Lexical constraints can only check presence or absence of words (ala bag-of-words) and thus clearly cannot address such role requirements. This work targets methods that can handle these requirements. Such complex syntactico-semantic constraints exist in many other tasks, e.g., biomedical mechanism generation, where certain signaling pathways must be present in the output (Bastan et al., 2022), goal-oriented generation, where the output must include the correct syntactic representation of the semantic goal (Baumler and Ray, 2022), and machine translation, where syntactic differences induce semantic failures (Wang et al., 2018).

We introduce a new decoding algorithm called NeuroStructural Decoding, which supports *structural lexico-syntactic constraints* during infer-

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 9496–9510
July 9-14, 2023 ©2023 Association for Computational Linguistics

ence. Specifically, it supports *unary* constraints that verify syntactic roles (e.g., the word *John* appears as a subject), *binary* constraints that verify a single dependency (e.g., the word *John* is the subject of *introduced*), and *triplet* constraints that verify a two-way relation (e.g., *John* is the subject of *introduced* whose object is *Kayla*).

To efficiently track whether these structural constraints are satisfied, we extend the NeuroLogic Decoding (Lu et al., 2021b) algorithm, which assigns constraint scores to indicate the degree to which the constraints are satisfied. We address two specific challenges here. First, unlike their lexical counterparts, syntactic constraints do not lend themselves to hard definitions of reversible and irreversible constraints. Therefore, we use a soft interpretation that allows for satisfiability at later stages. Second, enforcing structural constraints requires tracking syntactic relations at every step of the decoding process, which in turn requires parsing the partially generated texts. Since parsers are typically trained to handle complete sentences, we adapt a dependency parser to handle incomplete sentences.

We demonstrate the utility of NEUROSTRUCTURAL DECODING on three different tasks and on multiple language generation models. Evaluations show that automatically-derived structural constraints enforced by NEUROSTRUCTURAL DECODING improve generation performance on three distinct tasks: (i) a constrained text generation challenge, COMMONGEN (Lin et al., 2019); (ii) a mechanism summarization task, SuMe (Bastan et al., 2022), (iii) a machine translation task from German to English (Barrault et al., 2019).

2 Related Work

Constrained generation methods aim to produce outputs that satisfy specific constraints. There is a vast body of work spanning methods that operate at inference-time (Post and Vilar, 2018; Miao et al., 2019; Lu et al., 2021b; Kumar et al., 2022; Yang and Klein, 2021; Turcan et al., 2022), or training-time (Krishna et al., 2020; Lample et al., 2018; Kobayashi et al., 2022; Kumar et al., 2021).

The inference-time methods, which include the work proposed here, can be divided into two broad categories based on the types of constraints considered. The first category focuses on simple lexical constraints, which specify the presence or absence of certain words or phrases in the generated text. The second focuses on satisfying more sophisticated non-lexical constraints and enforcing rules on the overall organization of the generated text. These two categories are discussed in detail below.

2.1 Lexical Constraints

Anderson et al. (2017) introduced constrained beam search by using a finite-state machine to track constraints. Their method forces the inclusion of selected tag words in the output and fixed, pre-trained word embeddings to facilitate vocabulary expansion to previously unseen tag words.

Hokamp and Liu (2017) developed grid beam search as an extension of traditional beam search that incorporates pre-specified lexical constraints and groups together hypotheses based on the number of constraints satisfied. This method is biased towards fulfilling constraints greedily, which can lead to sub-optimal solutions in many cases.

Post and Vilar (2018) improved the previous method with a dynamic beam allocation strategy, which can only consider a small number of hypotheses at each step. Later, Miao et al. (2019) developed the Constrained Generative Model which allows multiple keywords in the generated sentences. It involves inserting all positive constraint keywords in random order and then randomly sampling edits to improve the fluency of the sentence by replacing, inserting, or deleting words.

Sha (2020) presented an unsupervised optimization method for addressing lexically-constrained generation problems. Using gradients, they determine the appropriate location in a sequence for inserting, replacing, or deleting words.

Lu et al. (2021b) proposed NeuroLogic Decoding which uses logical constraints to include or exclude lexical constraints, i.e., words or simple phrases. The method proposed here is similar to this approach, but without limiting the constraints to lexical ones. Instead, our method also employs structural constraints that contain syntactic relations between words. This approach enables more flexible and effective constrained generation, allowing for the consideration of both lexical and structural constraints simultaneously.

2.2 Non-Lexical Constraints

Kumar et al. (2021) introduced a decoding algorithm for controlled text generation which allows for the simultaneous consideration of multiple differentiable sentence-level constraints to control the desired attributes of the generated text. However, the inclusion of complex structure based attributes can slow down the decoding process. Wang et al. (2021) proposed a method that can be incorporated to leverage multiple rules simultaneously. These rules go beyond simple lexical constraints, but they still only consider semantics (content control) and do not take into account syntactic structures. Nye et al. (2021) introduced a method that incorporates logical constraints to improve the coherence and consistency of the generated text. This approach examines the generated sentences for semantic consistency and operates only at the sentence level. It only considers semantic consistency with previously known conditions rather than syntactic constraints.

The incorporation of structure into machine translation has been the subject of several studies. Chen et al. (2017) proposed a syntax-aware encoder-decoder approach. These studies have aimed to train models that incorporate structure into the generated output. Fei et al. (2020) introduced a structure-aware transformer model for machine translation. Bastan and Khadivi (2022) presented a reordering layer to the BiLSTM architecture with the aim of incorporating structural information during the training process of MT in low resource settings. Yang et al. (2022) proposed a tree-based machine translation approach that utilizes lexical dependency parsing.

NEUROSTRUCTURAL DECODING is different from previous research in that it aims to incorporate syntax in the form of structural constraints into the beam search decoding algorithm for constrained generation. Our approach allows for the generation of output that satisfies both lexical and syntactic constraints without modifying the decoding algorithm or operating only at the semantic level. All is done at the inference stage without necessitating any additional model training.

3 NEUROSTRUCTURAL DECODING

Our goal is to support structural constraints that require certain relationships to hold between the lexical items in the generated sentence. In this work, we target structural constraints based on syntactic dependencies due to their ubiquity. Our method can potentially accommodate other constraints such as semantic roles or other domain-specific relations.

We expand on the NeuroLogic Decoding (Lu et al., 2021b) framework, which only handles lexical constraints. The fundamental idea is to assign a score to each hypothesis in the beam based on how many lexical constraints have been satisfied. The algorithm uses pruning, grouping, and selection strategies to efficiently search for the highest scoring hypotheses that satisfy as many of the constraints as possible. Our work modifies this framework to support structural syntactic constraints and their logical combinations.

3.1 Structural Constraints

Formally, the input to NEUROSTRUCTURAL DECODING consists of a conjunctive normal form (CNF): $\{C_1 \wedge C_2 \wedge \cdots \wedge C_k\}$ where each clause $C_i = (D_{i1} \vee ... \vee D_{im})$ is a disjunction of some m literals. Each literal D_{ij} is a structural constraint expressing a logical condition for whether a specific syntactic structure should be present or absent in the generated text. To assess if a hypothesis in the beam (i.e., a partial generation) satisfies such a constraint, we check if the dependency tree of the hypothesis contains the syntactic structure. We support three types of structural constraints:

- **Unary Constraint:** A unary constraint asserts a dependency role for a single word. For example the unary constraint $D = (ball, obj)$ specifies that the word *ball* should appear in the generated text as an object of some verb.

- **Binary Constraint:** A binary constraint asserts a dependency relation between two words. The binary constraint $D = (team, subj, run)$, for example, asserts that the word *team* should appear as the subject of the word *run*.

- **Triplet Constraint:** A triplet constraint asserts a syntactic condition over three words specifying two dependency relations. For example, the constraint $D = (team, run, field)$ specifies two connected dependency relations. The word *run* should appear as the verb that connects the subject *team* and the object *field*. The triplet constraints allow for more fine-grained control for approximating semantic relations often expressed via predicate-argument structures.

3.2 Decoding

The goal of decoding is to find sequences that are highly probable under the decoding model and ideally satisfy the constraints. In practice, the problem is framed as an optimization problem that balances both aspects by using a scoring function that penalizes for the number of clauses that are not satisfied.

For a CNF with k clauses this can be stated as the following maximization objective for decoding:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} P_\theta(y|x) - \lambda \sum_{i=1}^{k} (1 - C_i)$$

where we overload C_i to denote a function that returns 1 if the underlying clause is true, i.e., if at least one of the literals in its disjunction is satisfied, and zero otherwise.

This objective is then used with a beam search where at each step we score each hypothesis in the beam based on the language model probability and the penalty for structural constraint satisfaction. The overall process can be summarized in the following steps:

1. Use the decoder to generate the top n probable words to extend each hypothesis in the beam.

2. For each extended hypothesis l, produce a parse tree P_l using a dependency parser.

3. Use pruning to remove hypotheses that have irreversibly violated any of the clauses; then group hypotheses based on shared irreversibly satisfied clauses. Within each group, use a selection strategy to maximize the chances of finding hypotheses that meet the constraints.

4. Compute the penalty for each clause C_i based on whether any of its individual structural constraint D_{ij} is satisfied in P_l.

5. Use the λ-weighted combination of the total penalties for each hypothesis and its model probability as the final score for each group.

We refer the reader to Section 2.3 in (Lu et al., 2021b) for details on the pruning, grouping, and selection strategies. Here we detail the key changes in how we determine reversible and irreversible satisfaction.

3.3 Constraint States for Pruning

Some hypotheses can be effectively pruned from the beam if we know that they have violated a constraint in an irreversible manner. The NeuroLogic Decoding framework tracks the following states for each clause:

Satisfied-Reversible: A constraint that is satisfied but can be unsatisfied later on.

Satisfied-Irreversible: A constraint that is satisfied and can not later be changed to the unsatisfied.

Unsatisfied-Reversible: A constraint that is yet unsatisfied but can be reversed later on.

Unsatisfied-Irreversible: A constraint that is unsatisfied and can not be satisfied later.

Assigning these states is more complicated for NEUROSTRUCTURAL DECODING because the words mentioned in a structural constraint can appear in a hypothesis but violate the expected structural relationship. In such cases, we need to determine if this violation is an irreversible one. Note that unlike lexical constraints, here we cannot make *hard* guarantees for the irreversible determinations because the syntactic structure of a sentence may change as more tokens are generated. The heuristic we apply in this paper is two fold: (a) All appearance of binary and triplet constraints are irreversible because, in our experience, larger syntactic structures are less likely to change during decoding, and (b) All appearance of unary constraints are reversible. More details below.

For binary and triplet constraints, if a constraint is satisfied, it will be considered irreversible. For instance, if a word *boy* is seen as the subject of verb *plays*, we assume it can not be reversed later[2]. We also assume that if a structure is seen with different syntactic condition other than the given constraint, this is an Unsatisfied-Irreversible condition. For instance, consider the triplet constraint *(Kathy, plays, guitar)*. If the noun *John* appears as the subject of the verb *plays* rather than *Kathy*, and the word *ball* appears as its object rather than *guitar*, we deem the constraint as Unsatisfied-Irreversible.

For unary constraints, we are more cautious with irreversibility. That is, even if the word mentioned in a constraint appears in a different syntactic role (thus violating the constraint), we allow it to remain reversible. For example, for the constraint *(John, subj)*, if the word *John* is generated as the subject violating its required syntactic role as an object, we keep the constraint reversible, so that the word *John* can reappear later in the decoding process.

3.4 Parsing Incomplete Sentences

Handling syntactic constraints requires parsing incomplete sentences that are generated at each step of the decoding process. Since dependency parsers are typically trained on complete sentences, we introduce a simple adaptation process to extend their capabilities to handle incomplete sentences.

We use a two-stage process, where we first train the parser on the standard dependency training us-

[2]We acknowledge that there is no guarantee that this assumption will hold true under all circumstances.

ing Penn Treebank WSJ dataset (Marcus et al., 1993). Then, we adapt it to handle incomplete sentences by continuing the training on a modified dataset containing incomplete sentences. We construct this new dataset as follows. First, for each sentence in the original WSJ dataset, we extract multiple sentence fragments containing words at positions $[0, k]$, for every $k \in [1, n]$, where n is the sentence length. Second, we extract the constituency edges for the words included in these fragments. Third, We convert each constituency tree to the corresponding fragment's dependency parse tree. In this case we make sure we don't have any missing dependency edges. As we show in the next section, this simple two-stage training process substantially improves the parser's performance on incomplete sentences.

4 Evaluation

We demonstrate the utility of NEUROSTRUCTURAL DECODINGon a variety of benchmark tasks related to the controllable text generation: COMMONGEN in Section 4.1, SuMe in Section 4.2, and Machine Translation in Section 4.3.

To prepare the training data for partial sentences, we convert constituency parse trees of partial sentences to dependency parse trees using Stanford CoreNLP library (Manning et al., 2014). To train the dependency parser for partial sentences, we use Stanza (Qi et al., 2020) model. We adapt it to parse incomplete sentences using the three-stage training described in Section 3.4.

Table 1 shows the improvements of the adapted parser over the default Stanza parser model on the test partition of the WSJ dataset, modified to contain both complete and incomplete sentences. For all tasks, when checking for syntactic constraints for subjects we use $nsubj$ dependencies and for objects relations use *obj* and *obl* dependencies.

4.1 Constrained Commonsense Generation

COMMONGEN (Lin et al., 2019) is a constrained text generation challenge designed as an assessment of generative commonsense reasoning. The task is to create a coherent sentence describing a scenario using a set of common concepts. A plausible sentence describing a real-life scenario requires a grammatically correct construction while adhering to and reasoning over commonsense relationships between the concepts.

	Default	Adapted
UAS	78.75	95.09
LAS	74.90	93.86
CLAS	72.76	93.00
MLAS	71.41	92.57

Table 1: The adapted parser model's performance on partial sentences was evaluated using different metrics. The results indicate that the default Stanza model struggles to parse partial sentences. However, by training it with partial sentences, it is able to accurately predict the labels of both partial and full sentences.

4.1.1 Problem Formulation

Given a set of input concepts the goal is to use them and construct a sentence y that is grammatically correct, makes sense, and describes a typical scenario, in which all concepts are used. The input concepts are a set of nouns $\{c_1, ...c_p\}$ and verbs $\{v_1, \cdots, v_q\}$. An example instance is the set of nouns {girl, hand} and a verb {wash} and a valid output sentence is: *The girl washed her hand.*

Given the noun and verb sets we define different constraints. The unary constraints assert that the nouns should appear in a subject or an object role (e.g., girl$\xleftarrow{subj}$*), and the verb must be the main verb of the sentence (i.e., its head is `root`). The binary constraints assert that the nouns should pair with some verb in a subject or object dependency relation (e.g., girl$\xleftarrow{subj}$wash). The triplet constraints assert that noun pair should appear in the subject and object dependency relation with a verb (e.g. girl$\xleftarrow{subj}$wash$\xrightarrow{obj}$ hand).

Formally, we can state the constraints as follows:
Unary:

$$([c_1 \xleftarrow{subj} *] \vee [c_1 \xleftarrow{obj} *]) \\ \wedge \cdots \wedge ([c_p \xleftarrow{subj} *] \vee [c_p \xleftarrow{obj} *]) \\ \wedge ([v_1 \xleftarrow{root} \texttt{root}] \vee \cdots \vee [v_q \xleftarrow{root} \texttt{root}])$$

Binary: For each v_i in the verbs we add:

$$([c_1 \xleftarrow{subj} v_i] \vee [c_1 \xleftarrow{obj} v_i]) \\ \wedge \cdots \wedge ([c_m \xleftarrow{subj} v_i] \vee [c_m \xleftarrow{obj} v_i])$$

Triplet: For each verb v_i we add the following constraints:

$$([c_1 \xleftarrow{subj} v_i \xrightarrow{obj} c_2] \vee [c_2 \xleftarrow{subj} v_i \xrightarrow{obj} c_1] \vee \cdots \vee \\ [c_{m-1} \xleftarrow{subj} v_i \xrightarrow{obj} c_m] \vee [c_m \xleftarrow{subj} v_i \xrightarrow{obj} c_{m-1}])$$

4.1.2 Evaluation Setup

We treat this problem as a conditional text generation, where we train a model on the training data and use the automatically derived constraints at test time to perform NEUROSTRUCTURAL DECODING. The COMMONGEN dataset consists of 32,651 instances in train, 993 in validation set, and 1,497 in test set. We benchmark state-of-the-art text to text generation models T5 (Raffel et al., 2020) and BART (Lewis et al., 2020) and evaluate the performance of different types of constraints based on these models. We use a finetuned models from the Huggingface library (Wolf et al., 2019).

For evaluation, we follow the evaluation metrics introduced in (Lin et al., 2019) and structural coverage which measure the percentage of structural constraints that are satisfied. We extract the gold structural constraints by using Stanza (Qi et al., 2020) parser on the gold outputs. We evaluate the performance of Vanilla Decoding (beam-based decoder), NeuroLogic Decoding, and NEUROSTRUCTURAL DECODING.

4.1.3 Results

Table 2 shows the results for vanilla decoding, NeuroLogic Decoding and NEUROSTRUCTURAL DECODING when using the three types of constraints. For both models, the lexical constraints in NeuroLogic Decoding improve over vanilla decoding. Unary constraints provide small gains over vanilla decoding but binary and triplet constraints show significant improvements over both vanilla and the lexical constraints. The lexical coverage numbers show that vanilla decoding satisfies most of the lexical constraints already, which means that the underlying models already produces sentences that contain the necessary words but with low structural coverage between the concepts. In general, the improvements in structural coverage corresponds to gains in the target quality measures, showing the benefit of structural constraints for this task.

Table 3 presents examples of the outputs generated by the COMMONGEN task, comparing the results obtained from the NeuroLogic Decoding and NEUROSTRUCTURAL DECODING. These examples highlight three primary advantages of NEUROSTRUCTURAL DECODING: (a) it generates full sentences whereas NeuroLogic Decoding may fail sometimes; (b) it can adhere to the intended syntactic structure, and (c) it can produce semantically and syntactically correct sentences that surpass the quality of those from NeuroLogic Decoding.

4.2 Biomedical Mechanism Generation

Summarizing biomedical mechanisms requires generating the relation underlying two biomedical entities and a succinct and informative sentence that effectively explains this relationship (Bastan et al., 2022). This task presents significant challenges, as it requires extracting and integrating information from different sentences in order to generate a mechanism sentence that explains *why* the relation exists or *how* the relation comes about.

4.2.1 Problem Formulation

Given a set of sentences from the abstracts of the biomedical literature $X = \{x_1, x_2, ..., x_m\}$ and two main entities e_1 the *regulator* and e_2 the *regulated* entity , the goal is to output the correct relation between two entities (*positive* or *negative* regulation) and generate a sentence Y summarizing the mechanism underlying this relation.

For the lexical constraints in NeuroLogic Decoding we define a multi-word constraint that encloses the entity in the correct tag for it. Note that the task provides the regulator and regulated entity information as part of the input. We use this to create the two multi-word lexical constraints.

$$[< re > e_1 < er >], [< el > e_2 < le >]$$

For the structural constraints, our goal is to nudge the model towards sentences that express the correct relation between the input entities. We add unary and binary constraints that target some lexico-syntactic connection between the entities as a proxy for their semantic connection as follow.

Unary: These specify that the regulator entity appears as the *subject* of some verb and the regulated entity as the *object* of some verb.

$$([e_1 \xleftarrow{subj} *]) \wedge ([e_2 \xleftarrow{obj} *])$$

Binary: As a binary constraint, we require that the regulator and the regulated entities appear as the *subject* and *object* of the *same* verb:

$$([e_1 \xleftarrow{subj} * \xrightarrow{obj} e_2])$$

Note that we don't specify which verb heads these relations; it just needs to be the same verb.

4.2.2 Evaluation Setup

The SuMe (Bastan et al., 2022) dataset consists of 20,765 instances in the training set, 1,000 instances in the development set, and 1,000 instances

Model			Structural Coverage	Lexical Coverage	ROUGE-L	METEOR	CIDEr	SPICE
T5	Vanilla Decoding		41.1	89.7	41.8	21.0	12.1	43.1
	Neurologic Decoding		59.3	97.7	42.7	22.2	12.6	44.3
	NEURO STRUCTURAL DECODING	Unary	59.5	98.1	42.8	22.1	13.2	44.9
		Binary	52.2	98.0	43.5	23.4	13.3	45.6
		Triplet	**61.7**	**98.3**	**44.1**	**23.4**	**13.9**	**45.9**
BART	Vanilla Decoding		41.5	95.5	41.3	21.1	12.8	43.5
	Neurologic Decoding		58.2	98.1	42.6	22.4	13.0	44.4
	NEURO STRUCTURAL DECODING	Unary	62.1	98.2	42.8	22.9	14.4	44.7
		Binary	65.6	98.5	43.1	23.7	15.1	45.6
		Triplet	**72.2**	**99.1**	**44.5**	**24.2**	**15.5**	**46.1**

Table 2: Comparison of the different decoding strategies on the COMMONGEN task. We use two pretrained models (T5 and BART) for generation, and couple them with different decoding algorithms. Vanilla denotes the basic beam search with a beam size of 50.

Neurologic Decoding		NEUROSTRUCTURAL DECODING	
Constraints	Output	Constraints	Output
["girl","girls","Girl","Girls"], ["wash","washed","washing","washes"], ["hand","hands","Hand","Hands"]	Hand washing soap in a sink.	$([\text{girl}\xleftarrow{subj}*]\vee[\text{girl}\xleftarrow{obj}*])\wedge$ $([\text{hand}\xleftarrow{subj}*]\vee[\text{hand}\xleftarrow{obj}*])$ $\wedge$ $([\text{wash}\xleftarrow{root}\texttt{root}])$	A girl is washing her hand.
["jump","jumping","jumps","jumped"], ["backs", "back", "Back", "Backs"], ["dog","dogs","Dog","Dogs"]	A dog on the back of a boat jumping into the water.	$([\text{dog}\xleftarrow{subj}*]\vee[\text{dog}\xleftarrow{obj}*])\wedge$ $([\text{back}\xleftarrow{subj}*]\vee[\text{back}\xleftarrow{obj}*])$ $\wedge$ $([\text{jump}\xleftarrow{root}\texttt{root}])$	A dog is jumping in the water with his back to the boat.
["game", "games", "Game", "Games"], ["Side", "side", "Sides", "sides"], ["watched", "watch", "watches", "watching"]	the side continue to watch the game .	$([\text{game}\xleftarrow{subj}*]\vee[\text{game}\xleftarrow{obj}*])\wedge$ $([\text{side}\xleftarrow{subj}*]\vee[\text{side}\xleftarrow{obj}*])$ $\wedge$ $([\text{watch}\xleftarrow{root}\texttt{root}])$	soccer player watches from the side as the game continues.

Table 3: Examples of the outputs for the COMMONGEN task generated by NEUROSTRUCTURAL DECODING and Neurologic Decoding.

in the test set. We only use the test set to evaluate the performance of different constrained generation methods. We use a fine-tune model on top of a pre-trained SciFive (Phan et al., 2021) for this task and compare the three decoding strategies: vanilla beam decoding, NeuroLogic Decoding, and NEUROSTRUCTURAL DECODING. For evaluation, we follow the metrics introduced in (Bastan et al., 2022). We also report the structural coverage for all decoding algorithms.

4.2.3 Results

Table 4 shows the comparison between different types of decoding for SuMe task. While lexical constraints in NeuroLogic Decoding provide small gains in the generation quality measures, using structural constraints with NEUROSTRUCTURAL DECODING yields larger gains. Even the unary constraint provides improvements over the lexical constraints, and binary improves even further.

Lastly, we find that the structural constraints result in improvements in the relation generation performance as well. More analysis on the outputs of this task available in Appendix B.

Model		Structural Coverage	ROUGE-L	RG
Vanilla Decoding		38.2	43.3	79
Neurologic Decoding		41.5	43.7	80
NEURO STRUCTURAL DECODING	Unary	54.1	43.8	80
	Binary	**55.0**	**44.1**	**81**
	Triplet	-	-	-

Table 4: Comparison of decoding strategies on the SuMe mechanism generation task. The generation model is SciFive fine-tuned on the SuMe dataset.

4.3 Machine Translation

Lexical constraints have been previously used in machine translation (MT) for controlling specific terminology (Lu et al., 2021a), inferring the correct word inflection given lemmatized constraints (Jon et al., 2021), reducing gender bias problems (Lu et al., 2021b), and improving the positive lexical constraints satisfaction by vectorized data augmentation (Hu et al., 2019).

In this work, we show how automatically derived structural constraints when used with NEUROSTRUCTURAL DECODING can help improve MT performance. We evaluate our system on DE-EN translation, which is known to suffer from non-verbal agreement, idioms, terminology and name-entity mismatch, and intricate German verbal grammar (Barrault et al., 2019; Macketanz et al., 2018; Avramidis et al., 2019).

4.3.1 Problem Formulation

The MT models take as input a source language sentence $X = \{x_1, ..., x_n\}$ and output a target language sentence $Y = \{y_1, ..., y_m\}$. We formulate the constrained decoding version of this by automatically deriving a set of constraints C from X and use them during decoding.

To this end, we first parse the source language sentence X to identify the main verb (`root`) x_v, its subject x_s, and its object x_o. We then use a word-to-word translation of these to obtain a *set* of candidate translations for the verb (Y_v), the subject (Y_s), and the object (Y_o). . Lastly, we add constraints that capture the subject-verb-object relationship between these candidate translation sets to varying degrees.

Unary: This requires the subject translations to appear as the subject of some verb and similar specifications for the object and the main verb translations.

$$([y_{s1} \xleftarrow{subj} *] \vee \cdots \vee [y_{sp} \xleftarrow{subj} *]) \wedge$$
$$([y_{o1} \xleftarrow{obj} *] \vee \cdots \vee [y_{oq} \xleftarrow{obj} *]) \wedge$$
$$([y_{v1} \xleftarrow{root} \texttt{root}] \vee \cdots \vee [y_{vk} \xleftarrow{root} \texttt{root}])$$

Binary: This requires dependency relations between the subject and verb translations and the object and verb translations. For each verb translation y_{vi}, we add the following:

$$([y_{s1} \xleftarrow{subj} y_{vi}] \vee \cdots \vee [y_{sp} \xleftarrow{subj} y_{vi}])$$
$$\wedge ([y_{o1} \xleftarrow{obj} y_{vi}] \vee \cdots \vee [y_{oq} \xleftarrow{obj} y_{vi}])$$

Triplet: For each translated verb v_i we add the following constraints to establish that a pair of subject and object translations share the corresponding dependency relations with a specific verb translation. For each verb translation y_{vi} we add the following:

$$([y_{s1} \xleftarrow{subj} y_{vi} \xrightarrow{obj} y_{o1}] \vee [y_{s2} \xleftarrow{subj} y_{vi} \xrightarrow{obj} y_{o1}]$$
$$\vee \cdots \vee [y_{sp} \xleftarrow{subj} y_{vi} \xrightarrow{obj} y_{oq}])$$

4.3.2 Evaluation Setup

We use the German to English test portion of the WMT19 dataset (Barrault et al., 2019). This dataset contains 2000 sentences with a total of 36,141/39,561 words, and 8,763/6,764 unique word types in German and English. We use the same model as described in (Lu et al., 2021a) and use thee `newstest19` set to evaluate.

For word to word translation process explained in previous section, we take advantage of free, open source Google Translate API[3].

Model		Structural Coverage	BLEU
Vanilla Decoding		52.20	41.16
Neurologic Decoding		52.12	39.62
NEURO STRUCTURAL DECODING	Unary	53.92	41.90
	Binary	55.50	42.12
	Triplet	**55.54**	**42.35**

Table 5: Comparison of decoding strategies for German to English translation on the WMT19 dataset.

4.3.3 Results

Table 5 shows that the automatically derived structural constraints produce improvements in translation quality as measured by BLEU. Using the word-to-word translations directly as lexical constraints for NeuroLogic Decoding leads to a drop in translation quality, which suggests that the candidate word translations produced through word-to-word translation (which ignores context) are imperfect. Note that this penalizes NEUROSTRUCTURAL DECODING as well, since it uses the same lexical items in its structural constraints. Further, the structural coverage of NeuroLogic Decoding is the same as vanilla decoding, which suggests that the model likely struggles to generate structural relations between the translated words on its own. On the other hand, enforcing structural constraints between the word-to-word translations provides consistent gains across all three types of constraints,

[3]https://github.com/ssut/py-googletrans

with improvements of 1.2 BLEU and 3.3 structural coverage over vanilla decoding. These results demonstrate the effectiveness of NEUROSTRUCTURAL DECODING considering that it is negatively impacted by the imperfect word-to-word translations that are used in its constraints. Overall, these findings highlight the potential of NEUROSTRUCTURAL DECODINGas a means to enhance the performance of existing MT models.

5 Conclusion

We introduced a novel constrained generation decoding algorithm, called NEUROSTRUCTURAL DECODING, which infuses structural constraints driven by syntactic dependencies in the generated output. We evaluated our method on three generation tasks: constrained commonsense generation, summarizing biomedical mechanisms, and MT. We showed that our method generates better texts according to the various task-specific metrics when compared against vanilla decoding and constrained decoding that relies solely on lexical constraints.

Limitations

While we evaluate our method on three distinct generation tasks, we acknowledge that we rely on a single language (English) and a single type of structural constraints (on top of syntactic dependencies). Further work is required to verify if the proposed approach holds on other languages (e.g., it is unclear how much our method is impacted by low-resource languages where syntactic parsers may be of lower quality) and other types of structural constraints (e.g., semantic roles). This work focuses on relatively smaller langauge models and does not address the impact and modes of usage of structural constraints on larger language models such as GPT-3.

Ethics Statement

This work did not rely on any user studies or annotations.

Constrained generation may potentially be used for nefarious purposes such as infusing biases or misinformation in the generated texts. The authors do not condone any such usages of the proposed approach.

Acknowledgements

This material is based on research that is supported in part by the National Science Foundation under the award IIS #2007290, and in part by an award from the Stony Brook Trustees Faculty Awards Program.

References

Maria Alexeeva, Allegra A. Beal Cohen, and Mihai Surdeanu. 2022. Combining extraction and generation for constructing belief-consequence causal links. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 159–164, Dublin, Ireland. Association for Computational Linguistics.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, and Hans Uszkoreit. 2019. Linguistic evaluation of German-English machine translation using a test suite. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 445–454, Florence, Italy. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Mohaddeseh Bastan and Shahram Khadivi. 2022. A preordered RNN layer boosts neural machine translation in low resource settings. In *Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022)*, pages 93–98, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Mohaddeseh Bastan, Nishant Shankar, Mihai Surdeanu, and Niranjan Balasubramanian. 2022. SuMe: A dataset towards summarizing biomedical mechanisms. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6922–6931, Marseille, France. European Language Resources Association.

Connor Baumler and Soumya Ray. 2022. Hybrid semantics for goal-directed natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1946, Dublin, Ireland. Association for Computational Linguistics.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. *arXiv preprint arXiv:1707.05436*.

Hao Fei, Yafeng Ren, and Donghong Ji. 2020. Retrofitting structure-aware transformer language model for end tasks. *arXiv preprint arXiv:2009.07408*.

Ruohao Guo and Dan Roth. 2021. Constrained labeled data generation for low-resource named entity recognition. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4519–4533, Online. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Phillip Howard, Gadi Singer, Vasudev Lal, Yejin Choi, and Swabha Swayamdipta. 2022. Neurocounterfactuals: Beyond minimal-edit counterfactuals for richer data augmentation. *arXiv preprint arXiv:2210.12365*.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. 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 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.

Josef Jon, João Paulo Aires, Dusan Varis, and Ondřej Bojar. 2021. End-to-end lexically constrained machine translation for morphologically rich languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4019–4033, Online. Association for Computational Linguistics.

Hideo Kobayashi, Yufang Hou, and Vincent Ng. 2022. Constrained multi-task learning for bridging resolution. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 759–770, Dublin, Ireland. Association for Computational Linguistics.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. *arXiv preprint arXiv:2010.05700*.

Manoj Kumar, Yuval Merhav, Haidar Khan, Rahul Gupta, Anna Rumshisky, and Wael Hamza. 2022. Controlled data generation via insertion operations for NLU. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 54–61, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. 2021. Controlled text generation as continuous optimization with multiple constraints. *Advances in Neural Information Processing Systems*, 34:14542–14554.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2019. Commongen: A constrained text generation challenge for generative commonsense reasoning. *arXiv preprint arXiv:1911.03705*.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, et al. 2021a. Neurologic a* esque decoding: Constrained text generation with lookahead heuristics. *arXiv preprint arXiv:2112.08726*.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021b. NeuroLogic decoding: (un)supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299, Online. Association for Computational Linguistics.

Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, and Hans Uszkoreit. 2018. Fine-grained evaluation of German-English machine translation based on a test suite. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 578–587, Belgium, Brussels. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguis-*

tics: System Demonstrations, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6834–6842.

Maxwell Nye, Michael Tessler, Josh Tenenbaum, and Brenden M Lake. 2021. Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning. In *Advances in Neural Information Processing Systems*, volume 34, pages 25192–25204. Curran Associates, Inc.

Long N. Phan, James T. Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. Scifive: a text-to-text transformer model for biomedical literature. *CoRR*, abs/2106.03598.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Lei Sha. 2020. Gradient-guided unsupervised lexically constrained text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8692–8703, Online. Association for Computational Linguistics.

Elsbeth Turcan, David Wan, Faisal Ladhak, Petra Galuscakova, Sukanta Sen, Svetlana Tchistiakova, Weijia Xu, Marine Carpuat, Kenneth Heafield, Douglas Oard, and Kathleen McKeown. 2022. Constrained regeneration for cross-lingual query-focused extractive summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2668–2680, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xinyi Wang, Hieu Pham, Pengcheng Yin, and Graham Neubig. 2018. A tree-based decoder for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4772–4777.

Yufei Wang, Can Xu, Huang Hu, Chongyang Tao, Stephen Wan, Mark Dras, Mark Johnson, and Daxin Jiang. 2021. Neural rule-execution tracking machine for transformer-based text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 16938–16950. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Bowen Yang, Cong Han, Yu Li, Lei Zuo, and Zhou Yu. 2022. Improving conversational recommendation systems' quality with context-aware item meta-information. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 38–48, Seattle, United States. Association for Computational Linguistics.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

A Hyper-parameters

In Table 6 we use the following hyper parameters for NEUROSTRUCTURAL DECODING. The same name convention used as in (Lu et al., 2021b).

Task	COMMON GEN	SuMe	WMT19
beam size	50	50	100
prune factor	100	50	200
length penalty	0.2	0.1	0.6
sat. tolerance	2	2	3
batch size	32	4	32

Table 6: Hyper-parameters used in NEUROSTRUCTURAL DECODING

B SuMe Full Evaluation

The complete results of using different decoding algorithms on the SuMe task is shown in Table 7

The mechanism sentences have to express the correct relation and explain how or why the relation

Model		Structural Coverage	ROUGE-L	BLEURT	RG (F1)
Vanilla Decoding		38.2	43.3	47.8	79
Neurologic Decoding		41.5	43.7	48.1	80
NEURO STRUCTURAL DECODING	Unary	54.1	43.8	48.9	80
	Binary	**55.0**	**44.1**	**49.1**	**81**
	Triplet	-	-	-	-

Table 7: Comparison of decoding strategies on the SuMe mechanism generation task. The generation model is SciFive fine-tuned on the SuMe dataset.

is true. This often means that the model has to get multiple semantic connections correct in the generated sentence some cases are shown in Table 8. The lexico-syntactic connections enforced by the structural constraints help the model with the main semantic connection and also results in longer sentences that have a better chance of describing the underlying mechanisms.

Table 8 shows some examples of the generated output via NEUROSTRUCTURAL DECODING vs NeuroLogic Decoding. These examples show NEUROSTRUCTURAL DECODING can solve some of the issues mentioned in (Bastan et al., 2022) like missing entities, wrong relation, and generated mechanism.

C Machine Translation Output Examples

Table 9 shows the effectiveness of using NEUROSTRUCTURAL DECODING over unconstrained translation systems.

NeuroStructural Decoding	NeuroLogic Decoding	Target
In conclusion, the results of the present study demonstrated that Dppa4 promotes <re> NSCLC <er> progression via the inhibition of cell proliferation and the promotion of glycolysis, in part, by downregulating GLUT-4 and activating the <el> LDHB <le> enzyme.	In conclusion, the results of the present study demonstrated that Dppa4 promotes <re> NSCLC <er> progression.	In conclusion, Dppa4 promotes <re> NSCLC <er> progression, partly through glycolysis by <el> LDHB <le>.
In conclusion, we show that RhoA/ <el> ROCK <le> signaling suppresses <re> chondrogenesis <er> through the regulation of Sox9 expression and actin organization.	We conclude that RhoA/ <el> ROCK <le> signaling suppresses <re> chondrogenesis <er> through the regulation of Sox9.	In conclusion, our data suggest that RhoA/<el> ROCK <le> signaling suppresses <re> chondrogenesis <er> through the control of Sox9 expression and actin organization.
In conclusion, Mg co-administration attenuated CDDP-induced <re> CIN <er> by downregulating the expression level of <el> renal transporters <le> (rOct2/rMate1) and decreasing renal Pt accumulation.	In conclusion, Mg co-administration attenuated <re> CIN <er> by downregulating <el> renal transporters <le> .	In conclusion, Mg co-administration reduced Pt accumulation by regulating the expression of the <el> renal transporters <le>, rOct2 and rMate1 and, thereby, attenuated <re> CIN <er>.
In conclusion, development of <re> insulin resistance <er> in obese Zucker rats is associated with impairment of pancreatic amylase-gene expression, which can be reversed by <el> Ciglitazone <le> or diet. negative	In conclusion, development of <re> insulin resistance <er> in the obese Zucker rat is associated with impaired glucose metabolism. positive	In conclusion, both the increase of <re> insulin resistance <er> with age and its partial reversal by <el> Ciglitazone <le> treatment appear to modulate pancreatic amylase-gene expression in the obese Zucker rat. negative

Table 8: The example of the generated text for the SuMe task in NeuroStructural Decoding vs NeuroLogic Decoding.

NeuroStructural Decoding	NeuroLogic Decoding	Facebook-FAIR	Target
Angry mother defends herself: Lindsay Lohan attacks refugee family on open street.	Angry mother resists: Lindsay Lohan attacks refugee family on open street.	Angry mother fights back: Lindsay Lohan attacks refugee family in the street.	Angry mother defends herself: Lindsay Lohan attacks refugee family on the street.
Currently, this office is held by the Brazilians Cacau and Celia Sasic who have family roots in Cameroon.	Currently, this office is held by the Brazilians Cacau and Celia Sasic.	Cacau, who was born in Brazil, and Celia Sasic, who has family roots in Cameroon, currently hold the posts.	Currently, this position is filled by native Brazilian Cacau and Celia Sasic, who traces her family back to Cameroon.

Table 9: The example of the generated text for the WMT19 task in NeuroStructural Decoding vs NeuroLogic Decoding.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Left blank.

☑ A2. Did you discuss any potential risks of your work?
Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☑ Did you run computational experiments?

Left blank.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.