Annotating FrameNet via Structure-Conditioned Language Generation

Xinyue Cui
University of Southern California
xinyuecu@usc.edu

Swabha Swayamdipta
University of Southern California
swabhas@usc.edu

Abstract

Despite the remarkable generative capabilities of language models in producing naturalistic language, their effectiveness on explicit manipulation and generation of linguistic structures remain understudied. In this paper, we investigate the task of generating new sentences preserving a given semantic structure, following the FrameNet formalism. We propose a framework to produce novel frame-semantically annotated sentences following an overgenerate-and-filter approach. Our results show that conditioning on rich, explicit semantic information tends to produce generations with high human acceptance, under both prompting and finetuning. Our generated frame-semantic structured annotations are effective at training data augmentation for frame-semantic role labeling in low-resource settings; however, we do not see benefits under higher resource settings. Our study concludes that while generating high-quality, semantically rich data might be within reach, the downstream utility of such generations remains to be seen, highlighting the outstanding challenges with automating linguistic annotation tasks.[1]

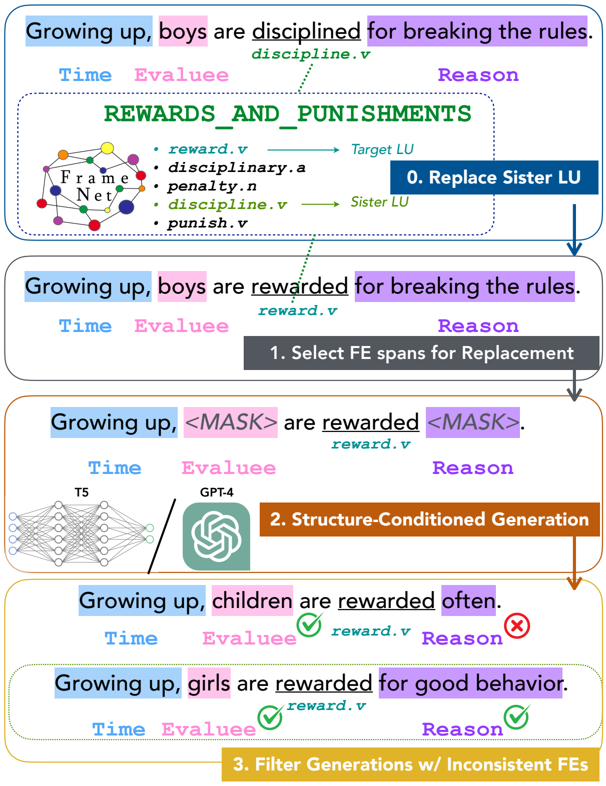

Figure 1: Our framework to generate frame semantic annotated data. Following Pancholy et al. (2021), we replace a sister LU with the target LU in an annotated sentence (0;§2). We select FEs appropriate for generating a new structure-annotated sentence (1;§3.1), and execute generation via fine-tuning T5 or prompting GPT-4 (2;§3.2). Finally, we filter out sentences that fail to preserve LU-FE relationships under FrameNet (3;§3.3).

1 Introduction

Large language models (LLMs) have demonstrated unprecedented capabilities in generating naturalistic language. These successes hint at LMs' implicit capabilities to *"understand"* language; but are they capable of processing explicit symbolic structures in order to generate language consistent with the structures? Not only would this help us understand the depth of LLMs' linguistic capabilities but would also serve to efficiently and cheaply expand existing sources of linguistic structure annotation. In this work, we investigate the abilities of LLMs to generate annotations for one such resource of linguistic structure: FrameNet, a lexical database grounded in the theory of frame semantics (Fillmore, 1985; Ruppenhofer et al., 2016). We propose an approach for language generation conditioned on frame-semantic structure such that the generation (i) is consistent with the frame structure, (ii) is acceptable by humans and (ii) is useful for a downstream task, namely frame-semantic role labeling (Gildea and Jurafsky, 2000b).

[1]Our code is available at https://github.com/X-F-Cui/FrameNet-Conditional-Generation.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 681–692
August 11-16, 2024

Our framework for generating frame-semantic annotations leverages both the FrameNet hierarchy and LLMs' generative capabilities to transfer annotations from existing sentences to new examples. Specifically, we introduce frame structure-conditioned language generation, focused on specific spans in the sentence such that the resulting sentence follows the given frame structure and is also acceptable to humans. Overall, we follow an overgenerate-and-filter pipeline, to ensure semantic consistency of the resulting annotations. Our framework is outlined in Figure 1.

Our intrinsic evaluation, via both human judgment and automated metrics, show that the generated sentences preserve the intended frame-semantic structure more faithfully compared to existing approaches (Pancholy et al., 2021). As an extrinsic evaluation, we use our generations to augment the training data for frame-semantic role labeling: identifying and classifying spans in the sentence corresponding to FrameNet frames. Under a low-resource setting, our generation annotations tend to be effective for training data augmentation for frame-semantic role labeling. However, these trends do not translate to a high-resource setting; these findings are consistent with observations from others who have reported challenges in leveraging LLMs for semantic parsing tasks, such as constituency parsing (Bai et al., 2023), dependency parsing (Lin et al., 2023), and abstract meaning representation parsing (Ettinger et al., 2023). Our findings prompt further investigation into the role of LLMs in semantic structured prediction.

2 FrameNet and Extensions

Frame semantics theory (Gildea and Jurafsky, 2000a) posits that understanding a word requires access to a **semantic frame**—a conceptual structure that represents situations, objects, or actions, providing context to the meaning of words or phrases. **Frame elements (FEs)** are the roles involved in a frame, describing a certain aspect of the frame. A **Lexical Unit (LU)** is a pairing of tokens (specifically a word lemma and its part of speech) and their evoked frames. As illustrated in Figure 1, the token "disciplined" evokes the LU *discipline.v*, which is associated with the frame `REWARDS_AND_PUNISHMENT`, with FEs including `Time`, `Evaluee`, and `Reason`. Grounded in frame semantics theory, FrameNet (Ruppenhofer et al., 2006) is a lexical database, featuring sentences that are annotated by linguistic experts according to frame semantics. Within FrameNet, the majority of sentences are annotated with a focus on a specific LU within each sentence, which is referred to as lexicographic data; Figure 1 shows such an instance. A subset of FrameNet's annotations consider all LUs within a sentence; these are called full-text data; Figure 1 does not consider other LUs such as *grow.v* or *break.v*.

FrameNet has defined 1,224 frames, covering 13,640 lexical units.The FrameNet hierarchy also links FEs using 10,725 relations. However, of the 13,640 identified LUs, only 62% have associated annotations. Our approach seeks to automatically generate annotated examples for the remaining 38% of the LUs, towards increasing coverage in FrameNet without laborious manual annotation.

Sister LU Replacement Pancholy et al. (2021) propose a solution to FrameNet's coverage problem using an intuitive approach: since LUs within the same frame tend to share similar annotation structures, they substitute one LU (the **target LU**) with another (a **sister LU**) to yield a new sentence. This replacement approach only considers LUs with the same POS tag to preserve the semantics of the original sentence; for instance, in Figure 1, we replace the sister LU *discipline.v* with the target LU *reward.v*. However, due to the nuanced semantic differences between the two LUs, the specific content of the FE spans in the original sentence may no longer be consistent with the target LU in the new sentence. Indeed Pancholy et al. (2021) report such semantic mismatches as their primary weakness.

To overcome this very weakness, our work proposes leveraging LLMs to generate FE spans that better align with the target LU, as described subsequently. For the rest of this work, we focus solely on verb LUs, where initial experiments showed that the inconsistency problem was the most severe. Details of FrameNet's LU distribution by POS tags, along with examples of non-verb LU replacements can be found in Appendix A.

3 Generating FrameNet Annotations via Frame-Semantic Conditioning

We propose an approach to automate the expansion of FrameNet annotations by generating new annotations with language models. Given sister LU-replaced annotations (§2; Pancholy et al., 2021), we select FE spans which are likely to be semantically inconsistent (§3.1), generate new sentences with replacement spans by conditioning on frame-semantic structure information (§3.2) and finally filter inconsistent generations (§3.3).

3.1 Selecting Candidate FEs for Generation

We identify the FEs which often result in semantic inconsistencies, in order to generate replacements of the spans corresponding to such FEs. Our selection takes into account the FE type, its ancestry under FrameNet, and the span's syntactic phrase type. Preliminary analyses, detailed in Appendix B, help us narrow the criteria as below:

1. **FE Type Criterion**: The FE span to be generated must belong to a core FE type, i.e., the essential FEs that are necessary to fully understand the meaning of a frame.
2. **Ancestor Criterion**: The FE should not possess Agent or Self-mover ancestors.
3. **Phrase Type Criterion**: The FE's phrase type should be a prepositional phrase.

Qualitative analyses revealed that it suffices to meet criterion (1) while satisfying either (2) or (3). For instance, in Figure 1, under `REWARDS_AND_PUNISHMENTS`, only the FEs `Evaluee` and `Reason` are core (and satisfy (2)) while `Time` is not; thus we only select the last two FE spans for generation.

3.2 Generating Semantically Consistent Spans

We generate semantically consistent FE spans for selected candidate FEs via two approaches: fine-tuning a `T5-large` model (Raffel et al., 2019) and prompting `GPT-4 Turbo`, following Mishra et al. (2021). In each case, we condition the generation on different degrees of semantic information:

No Conditioning We generate FE spans without conditioning on any semantic labels.

FE-Conditioning The generation is conditioned on the type of FE span to be generated.

Frame+FE-Conditioning The generation is conditioned on both the frame and the FE type.

The above process produces new sentences with generated FE spans designed to align better with the target LU, thereby preserving the original frame-semantic structure. However, despite the vastly improved generative capabilities of language models, they are still prone to making errors, thus not guaranteeing the semantic consistency we aim for. Hence, we adopt an overgenerate-and-filter approach (Langkilde and Knight, 1998; Walker et al., 2001): generate multiple candidates and aggressively filter out those that are semantically inconsistent. Details on fine-tuning `T5` and prompting `GPT-4` are provided in Appendix C.

3.3 Filtering Inconsistent Generations

We design a filter to ensure that the generated sentences preserve the same semantics as the expert annotations from the original sentence. This requires the new FE spans to maintain the same FE type as the original. We propose a new metric **FE fidelity**, which checks how often the generated spans have the same FE type as the original. To determine the FE type of the generated spans, we train an FE type classifier on FrameNet by finetuning SpanBERT, the state-of-the-art model for span classification (Joshi et al., 2019).[2] We use a strict filtering criterion: remove all generations where the FE classifier detects even a single FE type inconsistency, i.e. only retain instances with perfect FE fidelity.

3.4 Intrinsic Evaluation of Generations

We evaluate our generated frame-semantic annotations against those from Pancholy et al. (2021), before and after filtering (§3.3). We consider three metrics: perplexity under Llama-2-7B (Touvron et al., 2023) for overall fluency, FE fidelity, and human acceptance. We randomly sampled 1000 LUs without annotations under FrameNet and used our generation framework to generate one instance each for these LUs. For human acceptability, we perform fine-grained manual evaluation on 200 examples sampled from the generated instances.[3] We deem an example acceptable if the FE spans semantically align with the target LU and preserve the FE role definitions under FrameNet. We provide a qualitative analysis of generated examples in Appendix E.

Results in Table 1 shows that our filtering approach—designed for perfect FE fidelity—improves performance under the other two metrics. Compared to rule-based generations from Pancholy et al. (2021), our filtered generations fare better under both perplexity and human acceptability, indicating improved fluency and semantic consistency. Most importantly, models incorporating semantic information, i.e., FE-conditioned and Frame+FE-

[2] Our SpanBERT FE classifier attains 95% accuracy on the standard FrameNet 1.7 splits; see Appendix D for details.

[3] Human evaluation is mainly conducted by the first author of this work. These annotations were validated by two independent volunteers unfamiliar with generated data evaluating the same examples from `GPT-4` | Frame+FE, where the ratings differ by only 1% from our primary ratings. This suggests a consistent rating quality across different observers.

	Before Filtering ($\|D_{\text{test}}\|$=1K)			After Filtering (FE Fid. = 1.0)	
	FE Fid.	**ppl.**	**Human ($\|D_{\text{test}}\|$=200)**	**ppl.($\|D_{\text{test}}\|$)**	**Human ($\|D_{\text{test}}\|$)**
Human (FN 1.7)	0.979	78.1	1.000	97.0 (975)	1.000 (199)
Pancholy et al.	0.953	127.8	0.611	146.0 (947)	0.686 (189)
`T5`	0.784	139.3	0.594	117.5 (789)	0.713 (156)
`T5` | FE	0.862	127.6	0.711	112.7 (850)	0.777 (168)
`T5` | Frame + FE	**0.882**	136.8	0.644	124.4 (**873**)	0.704 (**172**)
`GPT-4`	0.704	114.9	0.528	114.2 (724)	0.723 (132)
`GPT-4` | FE	0.841	**106.3**	0.700	**103.4** (838)	**0.826** (164)
`GPT-4` | Frame + FE	0.853	117.2	**0.733**	111.8 (845)	0.821 (165)

Table 1: Perplexity, FE fidelity and human acceptability of `T5` and `GPT-4` generations conditioned on different degrees of semantic information. Number of instances after filtering are in parantheses. Best results are in boldface.

conditioned models, achieve higher human acceptance and generally lower perplexity compared to their no-conditioning counterparts, signifying that semantic cues improve both fluency and semantic consistency. Even before filtering, FE fidelity increases with the amount of semantic conditioning, indicating the benefits of structure-based conditioning. We also provide reference-based evaluation in Appendix F.

4 Augmenting Data for Frame-SRL

Beyond improving FrameNet coverage, we investigate the extrinsic utility of our generations as training data to improve the frame-SRL task, which involves identifying and classifying FE spans in sentences for a given frame-LU pair. Here, we consider a modified Frame-SRL task, which considers gold-standard frames and LUs, following Pancholy et al. (2021). This remains a challenging task even for powerful models like GPT-4, which achieves a test F1 score of only 0.228 in contrast to Lin et al. (2021)'s state-of-the-art F1 score of 0.722. For experimental ease, we fine-tune a SpanBERT model on FrameNet's full-text data as our parser[4] and avoid using existing parsers due to their reliance on weaker, non-Transformer architectures (Swayamdipta et al., 2017), complex problem formulation (Lin et al., 2021), or need for extra frame and FE information (Zheng et al., 2022).

As a pilot study, we prioritize augmenting the training data with verb LUs with F1 scores below 0.75 on average. This serves as an oracle augmenter targeting the lowest-performing LUs in the test set. For the generation of augmented data, we use our top-performing models within `T5` and `GPT-4` models according to human evaluation: `T5` | FE and `GPT-4` | Frame+FE models. Of 2,295 LUs present in the test data, 370 were selected for augmentation, resulting in 5,631 generated instances. After filtering, we retain 4,596 instances from `GPT-4` | Frame+FE and 4,638 instances from `T5` | FE. Additional experiments using different augmentation strategies on subsets of FrameNet are in Appendix G.

[4]This parser obtains an F1 score of 0.677, see Table 2.

	All LUs F1	Aug. LUs F1
Unaugmented	0.677 ± 0.004	0.681 ± 0.012
Aug. w/ `T5` | FE	0.683 ± 0.000	0.682 ± 0.006
Aug. w/ `GPT-4` | Frame+FE	0.684 ± 0.002	0.677 ± 0.010

Table 2: F1 score of all LUs and augmented LUs under unaugmented setting, augmented settings with generations from `T5` | FE and `GPT-4` | Frame+FE, averaged across 3 random seeds.

Table 2 shows the Frame-SRL performance, with and without data augmentation on all LUs and on only the augmented LUs. Despite the successes with human acceptance and perplexity, our generations exhibit marginal improvement on overall performance, and even hurt the performance on the augmented LUs. We hypothesize that this stagnation in performance stems from two factors: (1) the phenomenon of diminishing returns experienced by our Frame-SRL parser, and (2) the limited diversity in augmented data. Apart from the newly generated FE spans, the generated sentences closely resemble the original, thereby unable to introduce novel signals for frame-SRL; see subsection G.3 and Appendix H for more experiments on generation diversity. We speculate that Pancholy et al. (2021)'s success with data augmentation despite using only sister LU replacement might be attributed to use of a weaker parser (Swayamdipta et al., 2017), which left more room for improvement.

4.1 Augmenting Under Low-Resource Setting

To further investigate our failure to improve frame-SRL performance via data augmentation, we simulate a low-resource scenario and conduct experiments using increasing proportions of FrameNet training data under three settings: (1) training our SRL parser with full-text data, (2) training our SRL parser with both full-text and lexicographic data (which contains 10x more instances), and (3) training an existing frame semantic parser (Lin et al., 2021)[5] with full-text data, to control for the use of our specific parser.

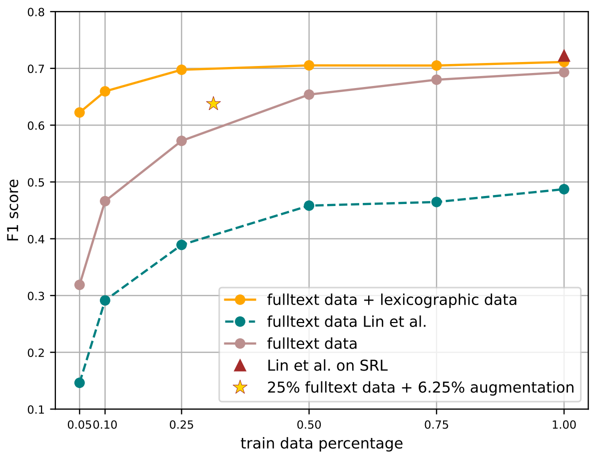

Figure 2: Learning curves for our frame-SRL model and Lin et al. (2021)'s end-to-end parser show diminishing returns on adding more human-annotated training data. The triangle marker denotes the performance of Lin et al. (2021)'s parser on SRL with gold frame and LU.

Figure 2 shows that parsers across all three settings exhibit diminishing returns, especially on the second setting, which utilizes the largest training set. This indicates that there seems to be little room for improvement in frame-SRL, even with human annotated data.

Following our learning curves, we further evaluate the utility of our generations without the influence of diminishing returns, by performing data augmentation in a low-resource setting. Specifically, we augment 25% of the full-text training data with an additional 6.25% of data generated using our method. As demonstrated in Figure 2, the performance of the model in this scenario not only exceeds that of the 25% dataset without augmentation but the results of the 25% dataset augmented with 6.25% of human-annotated data. This showcases the high utility of our generations for targeted data augmentation in a low-resource setting.

[5] Lin et al. (2021) break frame-SRL into three subsequent sub-tasks: target identification, frame identification, and SRL, contributing to worse overall performance.

5 Related Work

Data Augmentation for FrameNet While FrameNet annotations are expert annotated for the highest quality, this also limits their scalability. In an effort to improve FrameNet's LU coverage, Pavlick et al. (2015) proposes increasing the LU vocabulary via automatic paraphrasing and crowd-worker verification, without expanding the lexicographic annotations. Others address this limitation by generating annotations through lexical substitution (Anwar et al., 2023) and predicate replacement (Pancholy et al., 2021); neither leverages the generative capabilities of LLMs, however.

Controlled Generation Other works have explored using semantic controls for generation tasks. Ou et al. (2021) propose FrameNet-structured constraints to generate sentences to help with a story completion task. Ross et al. (2021) studied controlled generation given target semantic attributes defined within PropBank, somewhat coarse-grained compared to FrameNet. Similarly, Ye et al. (2024) employ the rewriting capabilities of LLMs to generate semantically coherent sentences that preserve named entities for the Named Entity Recognition task. Guo et al. (2022) introduced GENIUS, a novel sketch-based language model pre-training approach aimed at reconstructing text based on keywords or sketches, though not semantic structures; this limits its effectiveness in capturing the full context.

6 Conclusion

Our study provides insights into the successes and failures of LLMs in manipulating FrameNet's linguistic structures. When conditioned on semantic information, LLMs show improved capability in producing semantically annotated sentences, indicating the value of linguistic structure in language generation. Under a low-resource setting, our generated annotations prove effective for augmenting training data for frame-SRL. Nevertheless, this success does not translate to a high-resource setting, echoing challenges reported in applying LLMs to other flavors of semantics (Bai et al., 2023; Lin et al., 2023; Ettinger et al., 2023). These outcomes underline the need for further exploration into how LLMs can be more effectively employed in automating linguistic structure annotation.

Acknowledgements

We thank the anonymous reviewers and area chairs for valuable feedback. This work benefited from several fruitful discussions with Nathan Schneider, Miriam R. L. Petruck, Jena Hwang, and many folks from the USC-NLP group. We thank Ziyu He for providing additional human evaluation on generated annotations. This research was partly supported by the Allen Institute for AI and an Intel Rising Stars Award.

Limitations

While our work contributes valuable insights into LLMs' capabilities towards semantic structure-conditioned generation, we acknowledge certain limitations. First, our research is exclusively centered on the English language. This focus restricts the generalizability of our findings to other languages, which likely present unique linguistic structures with associated semantic complexity. The exploration of LLMs' capabilities in linguistic structures manipulation and generation in languages other than English remains an open direction for future research.

Moreover, we do not consider the full complexity of the frame semantic role labeling task, which also considers target and frame identification. Even for the argument identification task, we use an oracle augmentation strategy. Despite this relaxed assumption, the generations had limited improvement in performance, except in low-resource settings, where targeted data augmentation proved more effective. This indicates potential for improvement in scenarios with limited annotated data but highlights the need for further research in diverse and complex settings.

Ethics Statement

We recognize the inherent ethical considerations associated with utilizing and generating data via language models. A primary concern is the potential presence of sensitive, private, or offensive content within the FrameNet corpus and our generated data. In light of these concerns, we carefully scrutinize the generated sentences during the manual analysis of the 200 generated examples and do not find such harmful content. Moving forward, we are committed to ensuring ethical handling of data used in our research and promoting responsible use of dataset and language models.

References

Saba Anwar, Artem Shelmanov, Nikolay Arefyev, Alexander Panchenko, and Christian Biemann. 2023. Text augmentation for semantic frame induction and parsing. *Language Resources and Evaluation*, pages 1–46.

Xuefeng Bai, Jialong Wu, Jialong Wu, Yulong Chen, Zhongqing Wang, and Yue Zhang. 2023. Constituency parsing using llms. *ArXiv*, abs/2310.19462.

Allyson Ettinger, Jena D. Hwang, Valentina Pyatkin, Chandra Bhagavatula, and Yejin Choi. 2023. "you are an expert linguistic annotator": Limits of llms as analyzers of abstract meaning representation. In *Conference on Empirical Methods in Natural Language Processing*.

Charles J. Fillmore. 1985. Frames and the semantics of understanding. *Quaderni di Semantica*, 6(2):222–254.

Daniel Gildea and Dan Jurafsky. 2000a. Automatic labeling of semantic roles. In *Annual Meeting of the Association for Computational Linguistics*.

Daniel Gildea and Daniel Jurafsky. 2000b. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 512–520, Hong Kong. Association for Computational Linguistics.

Biyang Guo, Yeyun Gong, Yelong Shen, Songqiao Han, Hailiang Huang, Nan Duan, and Weizhu Chen. 2022. Genius: Sketch-based language model pre-training via extreme and selective masking for text generation and augmentation. *ArXiv*, abs/2211.10330.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Meghana Kshirsagar, Sam Thomson, Nathan Schneider, Jaime G. Carbonell, Noah A. Smith, and Chris Dyer. 2015. Frame-semantic role labeling with heterogeneous annotations. In *Annual Meeting of the Association for Computational Linguistics*.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 704–710, Montreal, Quebec, Canada. Association for Computational Linguistics.

Boda Lin, Xinyi Zhou, Binghao Tang, Xiaocheng Gong, and Si Li. 2023. Chatgpt is a potential zero-shot dependency parser. *ArXiv*, abs/2310.16654.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*.

Zhichao Lin, Yueheng Sun, and Meishan Zhang. 2021. A graph-based neural model for end-to-end frame semantic parsing. In *Conference on Empirical Methods in Natural Language Processing*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. In *Annual Meeting of the Association for Computational Linguistics*.

Jiefu Ou, Nathaniel Weir, Anton Belyy, Felix Yu, and Benjamin Van Durme. 2021. InFillmore: Frame-guided language generation with bidirectional context. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 129–142, Online. Association for Computational Linguistics.

Ayush Pancholy, Miriam R. L. Petruck, and Swabha Swayamdipta. 2021. Sister help: Data augmentation for frame-semantic role labeling. *ArXiv*, abs/2109.07725.

Ellie Pavlick, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Dredze, and Benjamin Van Durme. 2015. FrameNet+: Fast paraphrastic tripling of FrameNet. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 408–413, Beijing, China. Association for Computational Linguistics.

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith. 2018. Learning joint semantic parsers from disjoint data. *ArXiv*, abs/1804.05990.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Alexis Ross, Tongshuang Sherry Wu, Hao Peng, Matthew E. Peters, and Matt Gardner. 2021. Tailor: Generating and perturbing text with semantic controls. In *Annual Meeting of the Association for Computational Linguistics*.

Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, Collin F. Baker, and Jan Scheffczyk. 2016. *FrameNet II: Extended Theory and Practice*. ICSI: Berkeley.

Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. Framenet ii: Extended theory and practice.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. *ArXiv*, abs/1706.09528.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Marilyn A. Walker, Owen Rambow, and Monica Rogati. 2001. SPoT: A trainable sentence planner. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Junjie Ye, Nuo Xu, Yikun Wang, Jie Zhou, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llm-da: Data augmentation via large language models for few-shot named entity recognition. *ArXiv*, abs/2402.14568.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *ArXiv*, abs/2106.11520.

Ce Zheng, Yiming Wang, and Baobao Chang. 2022. Query your model with definitions in framenet: An effective method for frame semantic role labeling. *ArXiv*, abs/2212.02036.

A FrameNet Statistics

A.1 Distribution of Lexical Units

Table 3 illustrates a breakdown of FrameNet corpus categorized by the POS tags of the LUs. Specifically, we report the number of instances and the average count of candidate FEs per sentence, corresponding to LUs of each POS category. The two predominant categories are verb (v) LUs and noun (n) LUs, with verb LUs exhibiting a higher average of candidate FE spans per sentence compared to noun LUs.

LU POS	# Inst.	# FEs	# C. FEs	# Cd. FEs
v	82710	2.406	1.945	1.354
n	77869	1.171	0.675	0.564
a	33904	1.467	1.211	1.025
prep	2996	2.212	2.013	1.946
adv	2070	1.851	1.717	1.655
scon	758	1.906	1.883	1.883
num	350	1.086	0.929	0.549
art	267	1.547	1.543	1.408
idio	105	2.162	1.933	1.486
c	69	1.957	0.841	0.826

Table 3: Number of instances and average number of all, core, and candidate FE spans per sentence, categorized by POS tags of LUs in FrameNet. **C. FEs** represents Core FEs and **Cd. FEs** represents Candidate FEs.

A.2 Replacement of non-verb LUs

Table 4 shows several examples of non-verb LU replacement, where the resulting sentences mostly preserve semantic consistency. Given the extensive number of annotated verb LUs available for LU replacement and candidate FEs per sentence for masking and subsequent structure-conditioned generation, our generation methodology is primarily applied to verb LUs.

A.3 Full-Text and Lexicographic Data

Table 5 shows the distribution of the training, development, and test datasets following standard splits on FrameNet 1.7 from prior work (Kshirsagar et al., 2015; Swayamdipta et al., 2017; Peng et al., 2018; Zheng et al., 2022). Both the development and test datasets consist exclusively of full-text data, whereas any lexicographic data, when utilized, is solely included within the training dataset. Since our generation approach is designed to produce lexicographic instances annotated for a single LU, when augmenting fulltext data (§4), we break down each fulltext example by annotated LUs and process them individually as multiple lexicographic examples.

Frame	LU	Sentence
Leadership	king.n (rector.n)	No prior Scottish king (rector) claimed his minority ended at this age.
Sounds	tinkle.n (yap.n)	Racing down the corridor, he heard the tinkle (yap) of metal hitting the floor.
Body_part	claw.n (back.n)	A cat scratched its claws (back) against the tree.
Disgraceful _situation	shameful.a (disgraceful.a)	This party announced his shameful (disgraceful) embarrassments to the whole world .
Frequency	always.adv (rarely.adv)	The temple is always (rarely) crowded with worshippers .
Concessive	despite.prep (in spite of.prep)	Despite (In spite of) his ambition , Gass ' success was short-lived .
Conditional _Occurrence	supposing.scon (what if.scon)	So , supposing (what if) we did get a search warrant , what would we find ?

Table 4: Example sentences of non-verb LUs where semantic consistency is preserved after sister LU replacement. The original LU is in teal and the replacement LU is in orange and parentheses.

Dataset Split	Size
Train (full-text + lex.)	192,364
Train (full-text)	19,437
Development	2,272
Test	6,462

Table 5: Training set size with and without lexicographic data, development set size, and test set size in FrameNet 1.7.

B Details on Candidate FEs Selection

There are three criteria for determining a candidate FE span, i.e., FE Type Criterion, Ancestor Criterion, and Phrase Type Criterion. In preliminary experiments, we have conducted manual analysis on the compatibility of FE spans with replacement LUs on 50 example generations. As demonstrated through the sentence in Figure 1, the FE Type criterion can effectively eliminate non-core FE that do not need to be masked, i.e., "Growing up" of FE type `Time`. Also, the Phrase Type Criterion can identify the candidate FE "for breaking the rules", which is a prepositional phrase. Moreover, we find that FEs of Agent or Self-mover type describes a human subject, which is typically independent of

Sentence After Replacement	FE Type
She was bending over a basket of freshly picked flowers , organizing them to her satisfaction .	Agent (Agent)
The woman got to her feet , marched indoors , was again hurled out .	Self_mover (Self_mover)
While some presumed her husband was dead , Sunnie refused to give up hope .	Cognizer (Agent)

Table 6: Example sentences after LU replacement with FEs of type Agent, Self_mover, or their descendants, which are compatible with the new replacement LU. The ancestors of FE types are reported in parentheses. The FEs are shown in teal and the replacement LUs are shown in orange.

Model	Input
No Conditioning	Growing up, <mask> are rewarded <mask>.
FE-Conditioning	Growing up, <FE: Evaluee> <mask> </FE: Evaluee> are rewarded <FE: Reason> <mask> </FE: Reason>.
Frame-FE-Conditioning	Growing up, <Frame: Rewards_and_Punishments + FE: Evaluee> <mask> </Frame: Rewards_and_Punishments + FE: Evaluee> are rewarded <Frame: Rewards_and_Punishments + FE: Reason> <mask> </Frame: Rewards_and_Punishments + FE: Reason>.

Table 7: Template of finetuning T5 models on an example sentence.

the LU evoked in the sentence. Since FE types within the same hierarchy tree share similar properties, we exclude FEs of Agent and Self-mover types, as well as any FEs having ancestors of these types, from our masking process, as illustrated in Table 6.

C Details on Span Generation

C.1 T5-large Fine-Tuning

During the fine-tuning process of T5-large, we incorporate semantic information using special tokens, which is demonstrated in Table 7 through the example sentence in Figure 1. T5 models are fine-tuned on full-text data and lexicographic data in FrameNet for 5 epochs with a learning rate of 1e-4 and an AdamW (Loshchilov and Hutter, 2017) optimizer of weight decay 0.01. The training process takes around 3 hours on 4 NVIDIA RTX A6000 GPUs.

C.2 GPT-4 Few-shot Prompting

When instructing GPT-4 models to generate FE spans, we provide the task title, definition, specific instructions, and examples of input/output pairs along with explanations for each output, as demonstrated in Table 8.

D FE Classifier Training Details

Our classifier operates on the principle of classifying one FE span at a time. In cases where multiple FE spans are present within a single sentence, we split these into distinct instances for individual processing. For each instance, we introduce special tokens—<LU_START> and <LU_END>—around the LU, and <FE_START> and <FE_END> around the FE span. Additionally, the name of the evoked frame is appended to the end of the sentence. To train our classifier to effectively discern valid FE spans from invalid ones, we augment training data with instances where randomly selected word spans are labeled as "Not an FE", constituting approximately 10% of the training data. The FE classifier is fine-tuned on full-text data and lexicographic data for 20 epochs with a learning rate of 2e-5 and an AdamW optimizer with weight decay 0.01. The training process takes around 4 hours on 4 NVIDIA RTX A6000 GPUs.

E Human evaluation of generated examples

We perform fine-grained manual analysis on 200 generated sentences to evaluate the quality of model generations based on two criteria: (1) sentence-level semantic coherence and (2) preservation of original FE types. We present 10 example sentences from the overall 200 in Table 9.

F Intrinsic Evaluation on FrameNet Test Data

To evaluate the quality of generated sentences on reference-based metrics such as ROUGE (Lin, 2004) and BARTScore (Yuan et al., 2021), we perform §3.1 and §3.2 on the test split of FrameNet 1.7 with verb LUs. As observed in Table 10, the T5 | FE model surpasses others in ROUGE scores, signifying superior word-level precision, while GPT-4 achieves the highest BARTScore, indicat-

Title	Sentence completion using frame elements
Definition	You need to complete the given sentence containing one or multiple blanks (<mask>). Your answer must be of the frame element type specified in FE Type.
Example Input	**Frame**: Rewards_and_Punishments. **Lexical Unit**: discipline.v. **Sentence**: Growing up, <mask> are disciplined <mask>. **FE Type**: Evaluee, Reason.
Example Output	boys, for breaking the rules
Reason	The frame "Rewards_and_Punishments" is associated with frame elements "Evaluee" and "Reason". The answer "boys" fills up the first blank because it is a frame element (FE) of type "Evaluee". The answer "for breaking the rules" fills up the second blank because it is an FE of type "Reason".
Prompt	Fill in the blanks in the sentence based on the provided frame, lexical unit and FE type. Generate the spans that fill up the blanks ONLY. Do NOT generate the whole sentence or existing parts of the sentence. Separate the generated spans of different blanks by a comma. Generate the output of the task instance ONLY. Do NOT include existing words or phrases before or after the blank.
Task Input	**Frame**: Experiencer_obj. **Lexical Unit**: please.v. **Sentence**: This way <mask> are never pleased <mask> . **FE Type**: Experiencer, Stimulus.
Task Output	

Table 8: Example prompts for GPT-4 models. Texts in green only appear in FE-Conditioning and Frame-FE-Conditioning models. Texts in orange only appear in Frame-FE-Conditioning models.

ing its generated sentences most closely match the gold-standard FE spans in terms of meaning. For reference-free metrics, GPT-4 | FE performs well in both log perplexity and FE fidelity, showcasing its ability to produce the most fluent and semantically coherent generations.

G More on Augmentation Experiments

G.1 Experiments using Non-oracle Augmentation Strategy

To evaluate the robustness and generalizability of our model under realistic conditions, we employed an augmentation strategy similar to that used by Pancholy et al. (2021). Specifically, we remove all annotated sentences of 150 randomly selected verb LUs from the full text training data and train our baseline parser using the remaining training data. Our full model was trained on instances of the 150 verb LUs re-generated by our framework along with the data used to train the baseline model. As a result, the test F1 scores for the baseline model and full model were 0.689 and 0.690, respectively, which echos the lack of significant improvement using the oracle augmentation strategy.

G.2 Experiments on Verb-only Subset

Since our generation method mainly focuses on augmenting verb LUs, we conduct additional augmentation experiments using a subset of FrameNet that includes only verb LU instances. To ensure model performance on a subset of data, we incorporate lexicographic data with verb LUs into our training set, resulting in a training set enriched with 80.2k examples, a development set comprising approximately 600 examples, and a test set containing about 2k examples. We experimented with different augmentation percentages both with and without filtering, as shown in Table 11. We use an oracle augmenter to augment LUs inversely proportional to their F1 scores from the unaugmented experiments. To expand coverage on more LUs during augmentation, we augment all LUs rather than limiting to those with F1 scores below 0.75. Although the improvements are marginal, the outcome from filtered augmentations is generally better than those from their unfiltered counterparts.

G.3 Experiments on Multiple Candidate Generations

In the main experiments conducted in this paper, we generated one instance for each LU-sentence pair. However, instances could be filtered out due to inconsistent FE spans, which could hurt generation diversity. To address this, we further experimented with generating three candidate instances for each LU-sentence pair to improve generation coverage.

Specifically, we augmented the full-text training data by 25% under both the 1-candidate and 3-candidate settings. However, as shown in Table 12, generating three candidates did not lead to performance improvements in the F1 score. This suggests that simply increasing the number of generated candidates may not be sufficient to enhance

Frame	LU	Sentence	Original FEs	GPT-4 | FE	Human Eval.
Verification	verify.v (confirm.v)	The bank, upon confirming <Unconfirmed_content>, released the goods to the customer.	compliance with the terms of the credit	the transaction details	✓ ✓
Distributed_position	blanket.v (line.v)	<Theme> lines <Location> and the lake is covered with ice.	snow many feet deep, the land	the first snowfall, the shore	✓ ✓
Being_located	sit.v (stand.v)	Against the left-hand wall nearest to the camera are three storage shelves; <Theme> stands <Location>.	a lidless unvarnished coffin in the process of construction, on the middle shelf	a tall vase, on the top shelf	✓ ✓
Evoking	conjure.v (evoke.v)	A name like Pauline Gascoyne inevitably evoke <Phenomenon>.	an image of a bimbo Gazza in a GTi	memories of a bygone era	✓ ✓
Event	happen.v (take place.v)	Jamaicans appear to worry little about the future; sometimes it seems that they worry little even about what takes place <Time>.	in the next few minutes	tomorrow	✓ ✓
Self_motion	climb.v (walk.v)	My mother parked her bicycle in the shoulder and took my hand, and we walked <Goal>.	to the top of the hill	to the park	✓ ✓
Process_materials	stain.v (process.v)	If you accidentally process <Material> <Alterant>, leave it for a week or two.	walls, with woodworm fluid	the wood, too much	✓ ×
Self_motion	creep.v (make.v)	Matilda took the knife she had been eating with, and all four of them make <Path>.	towards the dining-room door	their way to the living room	✓ ×
Hunting	hunt.v (fish.v)	<Food> too were mercilessly fished and often left, plucked and dying, where the sealers found them.	The albatrosses	The penguins	× ✓
Change_position_on_a_scale	dip.v (rise.v)	<Attribute> rose <Final_value> in the summer, but has recently climbed above $400 and last night was nudging $410.	The price per ounce, below $360	The price, to $410	× ✓

Table 9: Example Generations of GPT-4 | FE, our best model according to human acceptance. The two marks in human evaluation represent whether the generations satisfy the two criteria individually: (1) sentence-level semantic coherence and (2) preservation of all FE types. A sentence is deemed acceptable only when it satisfies both criteria. The new replacement LUs are presented in orange or parentheses. Masked FE spans are presented in teal and their corresponding FE types in angle brackets.

	BARTScore	ROUGE-1	ROUGE-L	Perp.	FE Fid.
Human	-	-	-	4.82	-
T5	-5.939	0.301	0.298	447.874	0.829
T5 | FE	-5.922	**0.318**	**0.316**	434.231	0.840
T5 | Frame + FE	-6.179	0.276	0.274	441.639	0.843
GPT-4	**-4.060**	0.228	0.227	85.820	0.880
GPT-4 | FE	-4.336	0.218	0.217	**82.977**	**0.930**
GPT-4 | Frame + FE	-4.395	0.210	0.209	87.548	0.929

Table 10: Log BARTScore, ROUGE scores and perplexity of generations on FrameNet test set without LU replacement.

	All LUs F1	**Aug. LUs F1**
Unaugmented	0.751	0.779
5% Aug. w/o filter	0.745	0.778
5% Aug. w/ filter	0.752	**0.781**
25% Aug. w/o filter	0.752	0.776
25% Aug. w/ filter	**0.753**	**0.781**

Table 11: F1 score of all verb LUs and augmented LUs in augmentation experiments using different percentages of augmentations generated by T5 | FE with and without filtering, compared to baseline results without data augmentation. Best results are in boldface

generation diversity. Future work may need to explore more effective strategies to improve the diversity of generated data.

	All LUs F1
Unaugmented	0.693
1-candidate	0.688
3-candidate	0.673

Table 12: F1 score of SRL parsers trained on unaugmented data and augmented data generated by T5 | FE under 1-candidate and 3-candidate strategies.

H Effect of Filtering on Generation Diversity

To examine the effect of filtering on the diversity of generated data, we have conducted experiments to compute the Self-BLEU scores to measure diversity for the same 1,000 instances discussed in §3.4. A lower Self-BLEU score indicates higher diversity, as it signifies less overlap within the generated texts. As demonstrated in Table 13, the diversity of the generated candidates increases after applying the filter, even surpassing the diversity of the original instances created by humans. This substantiates the effectiveness of our filtering process in enhancing the variability and quality of the generated sentences.

	Before Filtering	**After Filtering**
Human	0.298	-
T5	0.302	0.278
T5 | FE	0.295	0.277
T5 | Frame+FE	0.295	0.271
GPT-4	0.270	0.249
GPT-4 | FE	0.268	0.246
GPT-4 | Frame+FE	0.271	0.253

Table 13: Self-BLEU scores of the 1000 instances created in §3.4 before and after filtering.