Prompting Approaches to Abbreviation Expansion

Kyle Gorman
Google Research

Abstract

Abbreviations are an entrenched feature of the majority of the world's writing systems, and the ability to expand abbreviations in context is important for speech technologies and language understanding tasks. This study presents several experiments applying large language models, prompt engineering, and fine-tuning to the expansion of ad-hoc abbreviations in English, showing substantial improvements against noisy channel models previously used for this task.

Keywords: abbreviation expansion, prompting, text normalization

1. Introduction

Abbreviations are one of several types of entities processed during **text normalization** for speech and language processing applications. For speech applications like text-to-speech synthesis or automatic speech recognition, they are *non-standard words* (NSWs; Sproat et al. 2001), wordforms not generally pronounced according to the ordinary character-to-sound rules. Sproat et al. (2001) provide a taxonomy of abbreviations in English focusing on pronunciation. In English, for example, *NATO* is read as if it were a word, but *CIA* is read as a letter-sequence (i.e., an **initialism**), and *Blvd.* is read the same as the full word *Boulevard*. Abbreviation normalization poses difficulties that go beyond mere pronunciation, however. Unlike other types of NSWs, it is not always obvious what an abbreviation denotes, and context may be required to determine what word or phrase an abbreviation corresponds to. Thus abbreviation expansion is ultimately important both for pronunciation modeling and text understanding. Gorman et al. (2021) draw a distinction between high-frequency, *conventionalized* abbreviations—including units like *lbs* read as *pound(s)* or postal abbreviations like *QC* read as *Quebec*—and the open class of *ad-hoc* abbreviations coined on-the-fly as needed. Ad-hoc abbreviations are particularly common in contexts where brevity is important, as it is on mobile devices (Crystal 2008) or augmentative and alternative communication (AAC) devices used by people with gross motor impairments (e.g., Cai et al. 2024).

1.1. Prior work

There is a long literature on English abbreviation formation (e.g., Marchand 1969:§9, Cannon 1989). While much less is known about abbreviations in other languages, Gorman and Roark (2024) survey abbreviation strategies in 50 languages and scripts.

Expansion of abbreviations is by now an established experimental task (see Gorman et al. 2021:§1.2 for a survey up to that date). Abbreviation expansion has also been studied in highly-specific domains; e.g., Daza et al. (2022) study abbreviations occurring in a corpus of Slovenian-language biographies, and Hosseini et al. (2024) model English clinical abbreviations. However, the author is aware of only a few freely-available data sets for this task, mostly in English. Dekker and van der Goot (2020) study synthetic generation of abbreviation-rich English-language microblog posts. For evaluation, they ask human annotators to guess which of two posts is human-generated. They release the synthetic data generated by their best system. Gorman et al. (2021) release a data set created by asking human annotators to abbreviate sentences from English-language Wikipedia.

A wide variety of modeling strategies have been applied to abbreviation generation and expansion. Much prior work uses some form of the well-known noisy channel model (e.g., Aw et al. 2006). Gorman et al. (2021), for example, develop a noisy channel model in which a character-level pair n-gram model (e.g., Novak et al. 2016) is fused with a word-level language model over possible expansions, with the Viterbi algorithm used to decode. They find that imposing additional heuristics on the character-level expansions improves performance, as does replacing a conventional language model with a neural network language model. Cai et al. (2024), who study abbreviation expansion for AAC, use a large fine-tuned language model.

1.2. Contributions

In this study, I perform abbreviation expansion using the current generation of large language models (LLMs) in two different scenarios, one using data augmentation and distillation strategies for resource-constrained, low-latency applications, and one focusing on the quality of expansions generated, free of resource and latency concerns.

While abbreviation expansion is in some ways similar to other sequence-to-sequence tasks like machine translation, grapheme-to-phoneme conversion, etc., I hypothesize that modern LLMs are well-adapted to these tasks, for two reasons. First, abbreviation expansion is conceptually sim-

Proceedings of the Third Workshop on Computation and Written Language (CAWL 2026) @LREC 2026, pages 64–70
12 May, 2026.

	# sentences	# tokens
Training	21,318	332,829
Development	2,665	41,757
Testing	2,665	41,730
LM data	2,657,826	41,573,540

Table 1: Summary statistics for the data set (repeated from Gorman et al. 2021).

ilar to the masked word prediction task used during pre-training. Secondly, given the focus on the expansion of abbreviations within their sentential context, this task depends on the generation of locally-coherent text, one of the greatest apparent strengths of LLMs. Finally, I hypothesize that fine-tuning will condition models to properly attend to the input text and to conform to the constraints of abbreviation tasks.

2. Materials and methods

2.1. Data

The primary data for training, development and validation are drawn from the Gorman et al. (2021) corpus, described above, which contains sentences from English-language Wikipedia abbreviated by professional human annotators. Summary statistics are provided in Table 1, and example sentences drawn from this corpus are shown in Figure 1. Gorman et al. constrain the abbreviation task by requiring that any abbreviation be formed by deleting some—but not all—characters, so that any abbreviation is a proper non-empty subsequence of the word it abbreviates. These **deletion-based** abbreviations (Pennell and Liu 2010) are one of the most common form of ad-hoc abbreviation in English text, though as Gorman and Roark (2024) note, English also includes the aforementioned acronyms like *NATO* and *CIA*, stump compounds like *FiDi* (< *Financial District*), and abbreviations that involve a mix of deletion and substitution as in *cuz* (< *because*). Gorman et al. have a second team of annotators attempt to expand abbreviated sentences created by the first team to establish a human topline for the test set.

2.2. Tasks

Two different tasks were considered in this study. One potential application area for abbreviation expansion is in the generation of spoken driving directions, and this motivates a consideration of resources limitations and latency. Consider, for example, a New York driver whose trip requires them to take the next exit onto the Bear Mountain State Parkway. Highway signs for this exit may read *Bear Mtn Pkwy*, *Bear Mtn Pk*, and so on (see Figure 2 for an example). If the driving direction engine wants to give an instruction that the driver should take the next exit onto *Bear Mountain Parkway* any attempt to expand the abbreviations must be fed to the synthesis back-end before the driver reaches the off-ramp. In many cases, it is possible to offload this computation to a remote server via remote procedure call (RPC), but in others it is not possible to retrieve the RPC result (i.e., due to poor connectivity) before the driver reaches the exit. For this reason, a robust driving direction system must be "hybrid" in the sense that it is able to fall back to on-device generation with low latency and minimal degradation in quality. Similar concerns apply for traffic-sign recognition in advanced driver-assistance systems and self-driving vehicles, which read and interpret traffic signs.

In the first scenario, I consider **direct expansion** sentences using LLM prompting. In the driving direction scenario, a prompting expansion system could be run as a high-quality server-based system.

In the second, I consider the use of LLMs for **data augmentation**. That is, the LLM is prompted to generate more training data. This synthetic data is then combined with the original training data and used to train a noisy channel model, a naïve form of model distillation. While Gorman et al. did not design their noisy channel model with mobile devices in mind, that model requires several orders of magnitude fewer parameters than the LLMs used in the prompting experiments, and there are various ways one might reduce its computational footprint, such as lossy LM pruning (e.g., Stolcke 1998) or lossless compression techniques for weighted finite-state automata (e.g., Mohri et al. 2015). Similarly, inference with this model is efficient, allowing for latencies on the order of tens of milliseconds without specialized coprocessors. This simple data augmentation technique may allow the system to distill relevant knowledge into a resource-constrained, low-latency expansion system.

The downstream capabilities of language models—whether large or small, conventional or neural—are to some degree determined by decisions made during pre-processing (e.g., Gorman and Pinter 2025). Of these decisions, the choice of tokenization strategy is thought to be one of the most important (e.g., Bostrom and Durrett 2020). I hypothesize ad-hoc abbreviations might challenge the model in the sense that they might require a large number of short and/or rarely-used tokens. I also hypothesize that tokenization of ad-hoc abbreviations might have different effects in the data augmentation task, which requires generation of abbreviated text, and the direct abbreviation expansion task, which requires processing but not generation.

```
inernet is not th frst tech to rsult in time displcmnt .
they prvided asistnc to parishs wher ordrd by the chrc .
he ws dstngushd by a lngstndng intrst in pplr antiquities .
he latr bcam a sports brdcstr for the college ftbll games .
hwvr , due to their rare pblc apprnces , these rumors remn uncnfrmd .
```

Figure 1: Example human-abbreviated sentences (case-folded and tokenized) from the Gorman et al. development set.

Figure 2: A highway sign on the Palisades Parkway indicating an exit towards Perkins Memorial Drive and Bear Mountain State Parkway. Image credit: Alps' Roads (https://www.alpsroads.net); reproduced with permission of the photographer.

2.3. Models

Prompting experiments were conducted using Gemini 3.0 (size medium, instruction-tuned) with default generation and fine-tuning hyperparameters. The original pair n-gram model used by Gorman et al. is retained as a baseline and is used for data augmentation and distillation experiments.

In both tasks, prompts instructed the system not to insert words or substitute characters. In the data augmentation task, the model was also instructed not to delete words. In **few-shot** experiments, the prompt also included additional sample pairs of abbreviated and expanded sentences ("shots"), randomly sampled without replacement, from the training data, before providing the target example to either be expanded or abbreviated, depending on task. In direct expansion experiments using **fine-tuning**, the LLM was simply fine-tuned to produce the expanded sentence observed in the training data, without any shots in the prompts.

Sample prompts for both tasks are given in the appendix.

2.4. Metrics

Following Gorman et al. (2021), I use **word error rate** (WER), the percentage of incorrectly expanded words, as our primary metric. To facilitate error analysis, several additional statistics are reported. **Overexpansion rate** (OER) is the percentage of words in the hypothesis which are expanded but do not require expansion, i.e., because they are not abbreviated in the first place. **Underexpansion rate** (UER) is the percentage of abbreviated words which are not expanded in the hypothesis. **Incorrect expansion rate** (IER) is the percentage of abbreviated words which are expanded to the wrong word. I also introduce a novel metric, **length error rate** (LER). This statistic, computed at the sentence (rather than word) level, is the percentage of hypotheses whose length in words does not match that of the input sentence.

OER and IER, in particular, measure the degree to which a system is "Hippocratic" (Roark and Sproat 2014) in the sense that it does no harm to a human's ability to interpret abbreviated text. In other words, one might prefer to let human users cope with the unexpanded abbreviations rather than incorrect expansion or overexpansion.

LER can be thought of as a measure of the ability of a system to adhere to the instruction to neither insert nor delete words. It is undefined (or zero) for the Gorman et al. human topline because the annotator interface did not permit insertion and deletion, and it is similarly undefined for the Gorman et al. noisy channel model because that model is similarly constrained.

3. Results

3.1. Direct expansion

In direct expansion experiments, each prompt ends with a single abbreviated sentence and the response is a hypothesis expansion of that sentence. If the response contains a different number of words than the abbreviated sentence, a length error is recorded, and all abbreviated tokens are left as such, resulting in underexpansion errors.[1]

Results are shown in Table 2. All metrics are error rates, and thus lower values indicate better performance. General performance, as measured by WER, improved as the number of shots used was increased. With as few as 5 shots, prompt-based expansion outperformed the noisy channel baseline, and with 20 shots, it may have reached a

[1] One can imagine various heuristics for partially resolving length errors, but I consider the presence of a substantial number of length errors to be highly unexpected and undesirable, so no heuristics of this sort are employed here.

plateau. Further improvements were obtained with zero-shot fine-tuning, which produced the lowest overall WER. This best model represents a 3.62-point absolute (and an 81% relative) reduction in word-level error over the baseline, a substantial improvement. The zero-shot fine-tuning model also achieved the lowest UER and IER observed.

3.2. Data augmentation

In the data augmentation experiments, each prompt ends with a single full sentence from the Gorman et al. training set, and the response is a hypothesis abbreviation of that sentence. If the response contains a different number of words than the full sentence, a length error is recorded and the hypothesis is discarded. Each full sentence is presented 5 times with a different random sample of shots. All hypothesis abbreviations—except those containing length errors, which are discarded—are then concatenated with human-generated training data and the resulting data is used to fit the noisy channel baseline model. Two hyperparameters—the n-gram order of the pair n-gram model and the language model—are set to minimize WER on the held-out development set, and the best model is then evaluated against the test set. Results are shown in Table 3. Unexpectedly, none of the augmented models outperformed the baseline; some possible explanations are proffered in section 4.

3.3. Error analysis

LLM-based direct expansion, whether via few-shot prompting or fine-tuning, results in low OER, UER, and IER across the board. While all prompts included explicit instructions to the model instructing it to neither delete nor insert words, this was not sufficient: pilot experiments with zero-shot expansion (without fine-tuning) did not produce usable results due to excessive length errors, and nearly a quarter of sentences still contain length errors with 1-shot expansion. However, increasing the number of shots and employing fine-tuning were both effective in reducing such errors. Non-"Hippocratic" overexpansion and incorrect expansion errors are very uncommon in both tasks and all systems, though the baseline achieved a lower (i.e., perfect) OER than the prompting models.

Focusing on the fine-tuning-based direct expansion, underexpansion errors—themselves mostly due to the conventions used to handle length errors—and incorrect expansion errors are roughly equally common. Echoing a similar pattern of errors reported by Gorman et al. (2021), many incorrect expansion errors involve morphologically related words (e.g., **decreases* for *decrease*, **technology* for *technological*). As also noted by Gorman et al., the Wikipedia data consists of a mixture of both American and British spelling conventions, and while this results in incorrect expansions according to the exact match criterion used for evaluation, the resulting "errors" (e.g., American **theater* for British *theatre*, British **faecal* for American *fecal*) are mostly harmless for downstream applications.

The proposed data augmentation strategy appears to be capable of generating data that is both diverse and human-like. Table 4, for example, shows five abbreviated versions of a single, randomly-selected training sentence, generated using 20-shot prompting. Of these five, four are unique, and one is identical to the human-generated example from Gorman et al. corpus, despite the fact that the model itself was not exposed to this example.

4. Discussion

As hypothesized, direct expansion performed well, with errors decreasing as the number of shots increased; however, the best result was obtained with zero-shot fine-tuning. The one surprise was the high rate of length errors. While some prior work has studied constraining neural network generation using finite-state automata (e.g., Zhang et al. 2019, Koo et al. 2024), no available implementation was capable of expressing the conceptually simple constraints—i.e., each output word o_n must be a supersequence of the corresponding input word i_n—that characterize this task. Explicit prompt instructions and even fine-tuning were insufficient to completely remove these errors in both direct expansion or data augmentation. Future work should consider the performance of prompting in other output-constrained speech and language processing tasks, such as those which can be framed as tagging tasks. One potential direction is to consider reinforcement learning as an alternative to fine-tuning, under the hypothesis this may more effectively enforce the length constraint.

Despite the increased data diversity introduced by data augmentation, the naïve distillation technique did not improve the performance of the noisy channel model. While the reason for this failure must be speculative at present, it should be noted that the proposed technique provides synthetic data to the pair n-gram component modeling the formation of abbreviations (i.e., what characters are deleted). However, it does not meaningfully enrich the expansion LM component operating at the word level. Gorman et al. (2021) report that replacing the conventional LM with a neural network LM substantially improved performance, and it simply may be the case that there is much less headroom to improve the pair n-gram component. Data augmentation is unlikely to improve the expansion LM, since it itself is trained with naturally-occurring sen-

	WER	OER	UER	IER	LER
Human topline	3.51	2.23	0.30	4.88	(n.a.)
Noisy channel	2.90	0.00	2.13	4.08	(n.a.)
1-shot expansion	12.52	0.10	25.52	0.98	24.13
2-shot expansion	4.19	0.07	7.97	0.87	7.39
5-shot expansion	1.93	0.10	3.07	0.94	2.44
10-shot expansion	1.11	0.08	1.46	0.81	0.94
20-shot expansion	0.97	0.09	1.06	0.90	0.60
Zero-shot fine-tuning	0.57	0.08	0.52	0.59	0.49

Table 2: Direct expansion results on the Gorman et al. abbreviation data test set, with human topline and noisy channel model results from Gorman et al. 2021 for comparison. The best overall performance, as indicated by WER, are obtained using zero-shot prompts and fine-tuning. WER: word error rate; OER: overexpansion rate; UER: underexpansion rate; IER: incorrect expansion rate; LER: length error rate.

	WER	OER	UER	IER	LER	# sentences
Noisy channel	2.90	0.00	2.13	4.08	(n.a.)	21,318
1-shot augmentation	3.42	0.00	2.78	4.56	14.31	112,655
2-shot augmentation	3.42	0.00	2.83	4.52	9.46	117,828
5-shot augmentation	3.53	0.00	2.94	4.66	5.72	121,812
10-shot augmentation	3.58	0.00	2.95	4.76	4.01	123,636
20-shot augmentation	3.44	0.00	2.94	4.45	3.26	124,429

Table 3: Expansion results on the Gorman et al. abbreviation data test set with data augmentation and naïve distillation, with un-augmented results from Gorman et al. 2021 for comparison. Augmentation data that did not conform to the length specification was excluded, resulting in differing numbers of training sentences. WER: word error rate; OER: overexpansion rate; UER: underexpansion rate; IER: incorrect expansion rate; LER: length error rate; # sentences: number of training sentences.

tences of English. This data requires no additional labeling either by human or LLM.

5. Conclusions

Few-shot prompting and fine-tuning were both highly-effective methods for expanding English deletion-based abbreviations, resulting in substantial improvements over previous models and the human topline. However, data augmentation, combined with a naïve distillation strategy, was less effective, perhaps because the augmentation strategy failed to introduce diversity relevant to improving task performance. Future work should consider alternative augmentation strategies, generalize beyond deletion-based abbreviations, consider languages other than English, and make incremental improvements in prompt design, hyperparameter tuning, generation/decoding, and the like.

6. Limitations

The experiments reported here use the Gemini 3.0 size-medium instruction-tuned model, and the results may depend in part on the particular strengths or weaknesses of that model with respect to these tasks. However, pilot experiments using smaller models gave similar results, and given that the primary data is publicly available, the experiments could easily be reproduced using various public LLM endpoints and other publicly-available families of models.

The results reported here naturally also depend on hyperparameters and the wordings of the prompts, but no attempt was made to tune the generation hyperparameters, or the training hyperparameters for fine-tuning, or the structure of prompts.

Given that the models targeted here are pretrained on large amounts of web text, there is reason to suspect that these model might already have been exposed to some of the target sentences, given that these sentences were taken from Wikipedia and that the corpus was distributed via GitHub. Furthermore, many LLM products have an end-user license agreement which permits any submitted prompts to be used for future training. For example, Balloccu et al. (2024) perform a systematic review of recent NLP conference proceedings to identify data sets "contaminated" via submission to OpenAI's GPT-3.5 or GPT-4 endpoints. It is not clear how to rule out the possibility that the improved performance seen with the LLM prompting techniques reflects previous exposure to the Gorman et al. data.

```
thus , th tw powrs wer relatvly unabl to fght decisve battls .
thus , th two powrs wer relativly unabl to fght decisiv battls .
thus , th two powrs wer relatvly unabl to fght decisve battls .
thus , th two powrs wer reltivly unabl to fight dcisiv bttls .
thus , th two powrs wer relativly unabl to fght decisiv battls .
```

Table 4: Different versions of the same sentence abbreviated via 20-shot prompting, showing the natural diversity obtained by randomly varying which shots were used.

A final limitation of the experiments reported here is inherited from the Gorman et al. data, which only contains deletion-based English abbreviations. Given the considerable cross-linguistic diversity in abbreviation formation strategies (e.g., Gorman and Roark 2024) and general interest in the cross-linguistic capabilities of LLMs, future work should gather abbreviation data from other languages and scripts, and even in English, consider other types of abbreviation formation strategies. Finally, given the relevance of abbreviation expansion to traffic-sign recognition, it may make sense to consider a multimodal, vision-to-text abbreviation expansion task in future work.

7. Acknowledgments

Thanks to Adrian Benton, Christo Kirov, Shankar Kumar, and Brian Roark for technical assistance.

8. Bibliographical References

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 33–40.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624.

Shanqing Cai, Subhashini Venugopalan, Katie Seaver, Xiang Xiao, Katrin Tomanek, Sri Jalasutram, . . . , and Michael P. Brenner. 2024. Using large language models to accelerate communication for eye gaze typing users with ALS. *Nature Communications*, 15:9449.

Garland Cannon. 1989. Abbreviations and acronyms in English word-formation. *American Speech*, 64(2):99–127.

David Crystal. 2008. *Txtng: The Gr8 Db8*. Oxford University Press.

Angel Daza, Antske Fokkens, and Tomaž Erjavec. 2022. Dealing with abbreviations in the Slovenian biographical lexicon. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8715–8720.

Kelly Dekker and Rob van der Goot. 2020. Synthetic data for English lexical normalization: How close can we get to manually annotated data? In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6300–6309.

Gemini Team Google. 2023. Gemini: A family of highly capable multimodal models. ArXiv preprint arXiv:2312.11805. URL: https://arxiv.org/abs/2312.11805.

Kyle Gorman, Christo Kirov, Brian Roark, and Richard Sproat. 2021. Structured abbreviation expansion in context. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 995–1005.

Kyle Gorman and Yuval Pinter. 2025. Don't touch my diacritics. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 285–291.

Kyle Gorman and Brian Roark. 2024. Abbreviation across the world's languages and scripts. In *Proceedings of the Second Workshop on Computation and Written Language (CAWL) @ LREC-COLING 2024*, pages 36–42.

Manda Hosseini, Mandana Hosseini, and Reza Javidan. 2024. Leveraging large language models for clinical abbreviation disambiguation. *Journal of Medical Systems*, 48:27.

Terry Koo, Frederick Liu, and Luheng He. 2024. Automata-based constraints for language model decoding. In *First Conference on Language Modeling*.

Hans Marchand. 1969. *The Categories and Types of Present-Day English Word-Formation*, 2nd edition. Beck.

Mehryar Mohri, Michael Riley, and Ananda Theertha Suresh. 2015. Automata and graph compression. In *2015 IEEE International Symposium on Information Theory*, pages 2989–2993.

Joseph Novak, Nobuaki Minematsu, and Keikichi Hirose. 2016. Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-grams models in the WFST framework. *Natural Language Engineering*, 22(6):907–938.

Deana Pennell and Yang Liu. 2010. Normalization of text messages for text-to-speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4842–4845.

Brian Roark and Richard Sproat. 2014. Hippocratic abbreviation expansion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369.

Richard Sproat, Alan W. Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333.

Andreas Stolcke. 1998. Entropy-based pruning of backoff language models. In *Proceedings of the DARPA Broadcast News And Understanding Workshop*, pages 270–274.

Hao Zhang, Richard Sproat, Axel H. Ng, Felix Stahlberg, Xiaochang Peng, Kyle Gorman, and Brian Roark. 2019. Neural models of text normalization for speech applications. *Computational Linguistics*, 45(2):293–337.

A. Sample prompts

A.1. One-shot expansion prompt

```
You are a natural language
    annotation system. The following
    text is an English sentence that
    has been abbreviated by deleting
    some of the characters. Expand it
     to a complete, pragmatically
    plausible English sentence by
    inserting alphabetic characters
    into words. Respond in all
    lowercase. Do not substitute or
    delete characters. Do not insert
    or delete whitespace. Do not
    insert or delete words. Please
    return only the expanded sentence
    .
Here is an example:
th key featurs of knwldge rpostories
     nclud comuncatn forums . | the
    key features of knowledge
    repositories include
    communication forums .
at th turnamnt teams hd to present
    smthng based n ther projct . |
```

A.2. One-shot augmentation prompt

```
You are a natural language
    annotation system. The following
    text is an English sentence.
    Abbreviate it by deleting some of
     the characters. Do not
    substitute or insert characters.
    Do not insert or delete
    whitespace. Do not insert or
    delete words. Please return only
    the abbreviated sentence.
Here is an example:
this reaction uses calcium as a
    cofactor and plays an important
    role in the intracellular
    transduction of many
    extracellular signals . | this
    reactn uses calcum as a cofctr
    and plys an imprtn rle in the
    intracelular trnsductn of many
    extracellular signals .
the game features high precision
    physics simulation , online
    multiplayer and open architecture
     . |
```