When Speed Meets Intelligence: Scalable Conversational NER in an Ever-evolving World

Karim Ghonim
Amazon Alexa AI
Sapienza University of Rome
kghonim@amazon.it

Antonio Roberto
Amazon Alexa AI
xrobanto@amazon.it

Davide Bernardi
Amazon Alexa AI
dvdbe@amazon.it

Abstract

Modern conversational AI systems require sophisticated Named Entity Recognition (NER) capabilities that can handle complex, contextual dialogue patterns. While Large Language Models (LLMs) excel at understanding conversational semantics, their inference latency and inability to efficiently incorporate emerging entities make them impractical for production deployment. Moreover, the scarcity of conversational NER data creates a critical bottleneck for developing effective models. We address these challenges through two main contributions. First, we introduce an automated pipeline for generating multilingual conversational NER datasets with minimal human validation, producing 4,082 English and 3,925 Spanish utterances. Second, we present a scalable framework that leverages LLMs as semantic filters combined with catalog-based entity grounding to label live traffic data, enabling knowledge distillation into faster, production-ready models. On internal conversational datasets, our teacher model demonstrates 39.55% relative F1-score improvement in English and 44.93% in Spanish compared to production systems. On public benchmarks, we achieve 97.12% F1-score on CoNLL-2003 and 83.09% on OntoNotes 5.0, outperforming prior state-of-the-art by 24.82 and 8.19 percentage points, respectively. Finally, student models distilled from our teacher approach achieve 13.84% relative improvement on English conversational data, bridging the gap between LLM capabilities and real-world deployment constraints.

1 Introduction

Modern conversational AI systems are experiencing a paradigm shift from command-based interactions to natural dialogue. This shift requires sophisticated Named Entity Recognition (NER) capabilities that can handle complex, contextual conversational patterns. Unlike traditional voice commands that follow predictable structures, conversational requests contain implicit references, contextual dependencies, and nuanced intent expressions that challenge existing real-time Natural Language Understanding (NLU) models. Large Language Models (LLMs) have demonstrated remarkable success in understanding conversational semantics and handling dialogue complexity (White et al., 2025). However, their deployment in production conversational systems faces critical limitations: (1) inference latency, (2) inability to adapt to daily-emerging entities (e.g., new songs, products), and (3) scarcity of conversational data.

Knowledge distillation offers a solution to high latency by transferring LLM capabilities to smaller, faster models like BERT-based architectures (Devlin et al., 2019; Sanh et al., 2019). However, distillation requires high-quality training data that is often unavailable for conversational NER. Human annotation is expensive, time-consuming, and complex to scale across languages. Moreover, the dynamic nature of conversational system interactions presents an additional challenge, as user interaction styles continuously evolve, and entity catalogs change daily (e.g., new songs or products), requiring annotation frameworks that incorporate emerging entities without model retraining.

In this paper, we address these interconnected challenges through a comprehensive framework that first tackles the data scarcity problem. We introduce a scalable pipeline for automatically generating gold-standard conversational NER test sets with minimal human validation. Our approach leverages zero-shot learning to create conversational patterns with entity-type placeholders across two languages, then populates these patterns through weighted sampling from entity catalogs. After addressing data set availability, we developed a novel approach using LLMs as semantic filters with catalog-based entity grounding rather than direct extractors. This addresses the dynamic knowledge challenge without additional training and pro-

vides high-quality labels for live traffic, solving both data scarcity and quality issues. To meet latency constraints, we distill this knowledge into smaller BERT-based models that maintain accuracy while meeting real-time requirements.

We summarize our contributions as follows: (1) A scalable framework for automatically generating multilingual conversational NER datasets, addressing data scarcity in conversational AI systems; (2) A novel approach combining catalog grounding with LLM semantic filtering for automated live traffic labeling; (3) Empirical validation shows that BERT-based models trained with traffic-labeled data outperform those trained on traditional command-style datasets while maintaining production-suitable latency and cost requirements.

2 Related Works

Conversational Data and NER. Recent work has explored conversational dataset creation for NLP applications (Soudani et al., 2024; Majumdar et al., 2019), yet few resources target entity-centric tasks in dialogue contexts. Prior studies demonstrate that short, fragmented conversational utterances require contextual modeling across turns (Jayarao et al., 2018), while datasets with informal or user-generated queries highlight challenges in handling novel entities (Epure and Hennequin, 2023). Despite progress in dialogue data generation, conversational NER remains under-studied.

LLMs for NER and Catalog Grounding. Large language models have been applied to NER by reformulating tagging as generative extraction (Wang et al., 2025). While LLMs demonstrate strong few-shot performance, they remain sensitive to domain shifts and entity distributions (Nandi and Agrawal, 2024; Chen et al., 2023). Although earlier work has integrated knowledge bases into NER models, explicit catalog-grounded approaches for conversational NER have received limited attention.

Model Distillation for NER. Knowledge distillation has proven effective for transferring LLM capabilities to smaller, more efficient NER models (Ma et al., 2022; Zhou et al., 2021; Wang et al., 2023; Chen and He, 2023). Distilled models can approach LLM-level performance while maintaining efficiency for real-time deployment, including domain-specific applications (Cocchieri et al., 2025). However, distillation approaches targeting conversational NER remains largely unexplored.

3 Dataset Generation Pipeline

In this section, we present our conversational NER dataset generation pipeline to address the scarcity of evaluation data in this domain. First, we identify conversational utterances from production traffic (Section 3.1). Second, we semantically cluster these utterances to ensure comprehensive coverage of patterns and intents (Section 3.2). Third, we employ an LLM to generate patterns with entity-type placeholders, which human annotators validate (Section 3.3). Finally, we populate these patterns by sampling entities from live traffic, generating thousands of test examples (Section 3.4). This process yields our multilingual conversational NER benchmark comprising 4,082 English and 3,925 Spanish examples.

3.1 Conversational Utterance Detection

We begin with utterances from live conversational system traffic, capturing actual user dialogues from production deployments. Conversational utterances exhibit features uncommon in traditional command-based interfaces (e.g., *play song*) but prevalent in modern dialogue systems, including multi-clause structures, discourse markers, personal pronouns, and contextual references. Using these linguistic features, we create a heuristic-based classifier that identifies conversational utterances suitable for pattern generation.

3.2 Utterance Clustering

To ensure comprehensive coverage of diverse conversational patterns, we semantically cluster the identified conversational utterances. We encode utterances using sentence-transformers (Reimers and Gurevych, 2019) to capture their semantic representations, then apply HDBSCAN clustering (McInnes et al., 2017) to group similar utterances, requiring at least 5 utterances per cluster. These clusters serve as sources of seed utterances for pattern generation, ensuring the final dataset represents the full range of intents and interaction patterns observed in production traffic.

3.3 Pattern Generation

We leverage Claude 3.5 Sonnet v2 (Anthropic, 2024) to generate patterns from cluster representatives. The LLM abstracts entities into typed placeholders, transforming specific utterances (e.g., *"Play Beyonce"*) into reusable patterns (e.g., *"Play <Artist>"*). Our approach is inherently multilin-

gual: we generate both English and Spanish patterns from English-only seeds by specifying the target language in the prompt. An example of the prompt used is provided in Appendix D. Human annotators validate the generated patterns, resulting in 409 English and 405 Spanish validated patterns. By limiting human review to patterns rather than individual utterances, we optimize efficiency while maintaining quality.

3.4 Pattern Population

Once validated, these patterns can be populated with different catalog entities to generate numerous test examples without additional human validation. We replace entity-type placeholders with catalog entities sampled from live traffic for each language. This sampling reflects real-world utterances, where popular entities (e.g., *"Beyonce"* in English, *"Bad Bunny"* in Spanish) regularly appear.

4 Semantic Filtering for NER

We introduce our approach for performing NER at scale by formulating it as a semantic filtering task where an LLM selects relevant entity pairs (span, entity type) from pre-identified candidates extracted from entity catalogs. This transforms traditional generative NER into a constrained selection problem, decoupling knowledge from reasoning. Catalogs provide entity knowledge while the LLM evaluates semantic relevance. This eliminates the need for the LLM to have prior knowledge of entities or catalogs, enabling our system to handle entities created after the model's training cutoff and adapt to catalog changes without model updates.

Formally, given an input query and a set of candidate entity pairs obtained through exact text matching against entity catalogs, the model determines which candidates are semantically appropriate for the given context. The model receives candidates in a structured format: `{"span": "entity_text", "label": "entity_type"}`, where each candidate represents a potential entity mention. For example, given the query *"play harry potter"* with candidates `{"span": "harry potter", "label": "movie"}` and `{"span": "harry potter", "label": "book"}`, the model evaluates each candidate's contextual relevance. In this ambiguous case, both options are semantically valid; in other contexts (e.g., *"watch harry potter"*), the movie label would be more appropriate.

This design offers several advantages over generative approaches: (1) all predictions correspond to entities in our knowledge base, reducing hallucinations; (2) multi-label scenarios are naturally supported where spans can belong to multiple entity types; and (3) the model does not need to learn catalog-specific entity definitions, enabling flexible deployment across varying entity schemas.

4.1 Prompt Structure and Design

Our prompt design enforces strict constraints for reliable production performance. We structure the prompt with distinct sections: task definition, selection constraints, entity type descriptions, examples, and the target query with candidates. Each section guides the model's reasoning while maintaining clear information boundaries, facilitating error diagnosis and performance optimization. The task definition frames NER as a selection task, emphasizing that the model must choose from provided candidates rather than generating new entities. We enforce selection constraints by requiring exact copying of candidate entries, prohibiting new span or label generation, and specifying the expected output format. These constraints reduce out-of-vocabulary predictions and ensure downstream compatibility. We leverage entity-type descriptions to provide semantic grounding for disambiguation, focusing on distinguishing features rather than exhaustive definitions. These zero-shot descriptions enable the model to understand entity type boundaries without task-specific fine-tuning, supporting informed decisions on borderline cases while maintaining consistency with human annotation standards. The complete prompt template is provided in Appendix C.

Dynamic Exemplar Selection While static exemplar selection improves few-shot learning performance, it often fails to provide optimal context for diverse queries (Nori et al., 2023). We implement dynamic retrieval that selects the most contextually relevant exemplars from annotated training data for each input, leveraging the observation that semantically similar queries benefit from similar annotation patterns. We employ EmbeddingGemma-300m (Vera et al., 2025) for dense vector representations, selected for its multilingual semantic similarity performance and efficiency. For each query, we compute cosine similarity and retrieve the top 10 most similar utterances with annotations. This enables automatic adaptation across query intents and domains without manual curation, providing

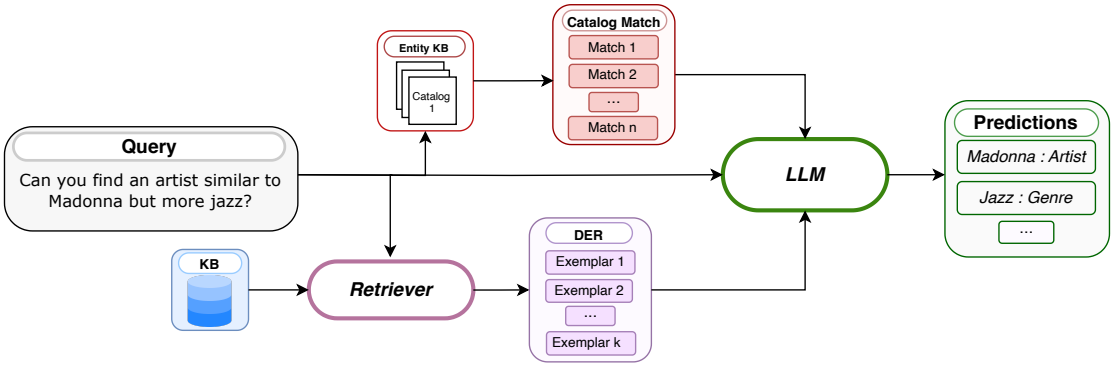

Figure 1: The NER Teacher Model pipeline including dynamic exemplar retrieval and catalog matching.

relevant disambiguation examples. Retrieved exemplars are formatted consistently with the target task, demonstrating semantic filtering application in similar contexts. An ablation study on retrieval size appears in Appendix B.

5 Experimental Setup

5.1 NER Teacher Model

We experiment with several prompting strategies to demonstrate each component's effect on overall performance using Claude 3.5 Sonnet v1 for all experiments. First, we evaluate performance using task-relevant instructions with randomly selected exemplars, referred to as ICL (In-Context Learning), serving as the baseline for our NER Teacher Model. This enables us to evaluate the effect of formulating NER as a Semantic Filtering (SF) task (Section 4), which uses the same randomly selected exemplars. Finally, we evaluate the effect of integrating dynamic exemplar retrieval (DER) into our final prompt. While we use the same prompt structure across languages, we leverage language-specific exemplars. We provide an example of our approach in Figure 1.

Post-Processing step Despite prompt-level constraints, LLMs occasionally generate outputs that violate specified requirements. We implemented post-processing pipeline step to ensure output quality. The pipeline includes validation of output format, verification of candidate adherence, and filtering of invalid predictions.

5.2 Student Model Distillation

Despite their strong performance and generalization capabilities, LLMs are prohibitively slow and expensive for latency-constrained production settings. For this reason, we investigate using them as teacher models for smaller BERT-based architectures. We study using an LLM to annotate conversational NER data, then training a smaller model to replicate these annotations, distilling the LLM's knowledge. In our experiments, we use XLM-RoBERTa as the backbone for our student model. To evaluate the impact of LLM-annotated conversational data, we train two variants with and without conversational NER data annotated by the NER Teacher Model. All models were trained with batch size 128 for 5 epochs using AdamW optimizer (Loshchilov and Hutter, 2019), learning rate 10^{-5}, 100 warmup steps, and 0.01 weight decay.

5.3 Datasets

Our internal evaluation uses three test sets: the generated multi-lingual conversational dataset (Section 3), high-frequency user requests, and entities absent from training data. These evaluate the NER Teacher Model's ability to handle conversational requests, process head-of-traffic distribution, and generalize to unseen entities. To avoid data leakage, we removed requests containing unseen entities from the example retrieval set.

Additionally, in order to assess generalization beyond our domain, we evaluate on CoNLL-2003 (Sang and De Meulder, 2003) and OntoNotes 5.0 (Pradhan et al., 2013), two widely-used open-source NER benchmarks. These datasets validate whether our approach extends beyond conversational AI to general NER. Since these open-source datasets lack entity catalogs, we create entity-type catalogs using all entities from the provided splits. Consistent with our internal evaluation, we use the training datasets as exemplar sources. Dataset

	English			Spanish		
Model	**Conversational**	**Head**	**Unseen**	**Conversational**	**Head**	**Unseen**
Baseline	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
ICL	+31.33 %	-2.93 %	+3.31 %	+32.80 %	-4.40 %	+1.04 %
SF	+36.83 %	+3.72 %	+19.70 %	+43.77 %	-2.22 %	+19.86 %
SF + DER	**+37.79** %	**+4.11** %	**+24.83** %	**+44.25** %	**+4.64** %	**+26.65** %
Baseline + PS	+5.54 %	**+5.82** %	+11.00 %	+5.29 %	**+6.69** %	+11.27 %
SF + DER + PS	**+39.55** %	+5.09 %	**+27.63** %	**+44.93** %	+4.97 %	**+28.07** %

Table 1: Relative performance improvements (% F1-score) compared to production baseline on internal test sets (English and Spanish). We also report performance using post-processing (PS). **Bold** indicates the best performance.

statistics are provided in Appendix A. For student model experiments, we use an internal dataset containing no conversational examples. As conversational training data, we include only 3,724 English samples annotated by the NER Teacher Model.

5.4 Evaluation Metrics

Following standard NER evaluation practice, we employ exact match precision, recall, and F1-score at the entity level, requiring both correct span identification and accurate entity type classification. This strict criterion ensures comprehensive assessment of entity detection and type disambiguation while supporting multi-label scenarios where entities belong to multiple entity-types.

6 Results

We present NER Teacher Model results on public and internal datasets, including production evaluation, followed by Student Model results when trained on Teacher-labeled conversational data.

6.1 Internal Dataset Evaluation

As shown in Table 1, our approach demonstrates exceptional performance gains on conversational data. In English, the basic ICL approach achieves a 31.33% relative improvement over the baseline, while SF raises this to 36.83%. Meanwhile, SF with dynamic exemplars (SF+DER) yields the strongest performance with a 37.79% relative improvement, highlighting the critical importance of contextually relevant examples for conversational NER tasks. We observe consistent results for Spanish as well, with the best-performing prompt (SF+DER) achieving 44.25% gains.

Head traffic evaluation reveals more nuanced results. While ICL shows slight performance degradation in both English and Spanish (-2.93% and -4.40% respectively), adding dynamic exemplars recovers performance (+4.11% in English and +4.64% in Spanish). This suggests head traffic benefits particularly from relevant contextual examples that help disambiguate common but potentially ambiguous entity mentions. For unseen entities, even basic ICL achieves a 3.31% increase in English, with SF reaching 19.70% and dynamic exemplars achieving 24.83% relative improvement. Consistent with other test sets, we observe a similar pattern in Spanish with dynamic exemplars boosting performance by 25.65%. These results demonstrate the system's ability to handle entities absent from training data, critical for production systems adapting to evolving catalogs.

Finally, in our production configuration, our post-processing (Section 5.1) achieves meaningful gains across all test sets. Most notably, it shows significant increases in conversational data (39.55% in English and 44.93% in Spanish) and unseen entities (27.63% in English and 28.07% in Spanish), indicating that our quality control pipeline effectively captures and corrects systematic errors.

6.2 Public Benchmark Evaluation

As shown in Table 2, our method achieves substantial gains on CoNLL-2003. SF raises the F1-score from 82.55% to 94.54%, representing an 11.99 percentage point increase. Similar to our internal evaluation, adding DER further enhances performance to 97.12%, confirming the effectiveness of contextually relevant example selection. Notably, our approach significantly outperforms the 72.30% F1-score reported by Ma et al. (2023), achieving a 24.82 percentage point gain while requiring no task-specific training. The balance between precision and recall is particularly noteworthy, with both metrics exceeding 97% in our best configuration.

	CoNLL-2003			OntoNotes 5.0		
Model	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
ICL	79.89%	85.39%	82.55%	65.09%	72.66%	68.67%
SF	94.21%	94.87%	94.54%	73.17%	89.39%	80.48%
SF + DER	**97.00%**	**97.24%**	**97.12%**	**77.58%**	**89.44%**	**83.09%**
Few-shot SoTA	N/A	N/A	72.30%	N/A	N/A	74.90%

Table 2: NER Performance comparison on public benchmarks. **Bold** indicates the best performance.

	English			Spanish		
Model	**Conversational**	**Head**	**Unseen**	**Conversational**	**Head**	**Unseen**
Baseline	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
With labeled data	**+13.84** %	**+5.61** %	**+6.33** %	**+9.21** %	**+7.51** %	**+7.39** %

Table 3: Student model performance gains (relative F1-score %) when trained with LLM-labeled traffic data compared to training on traditional datasets only (Baseline). **Bold** indicates best performance.

This indicates our semantic filtering approach effectively minimizes both false positives and false negatives, crucial for production deployment where prediction reliability is paramount.

OntoNotes 5.0 results show similar trends albeit the dataset's inherent complexity and larger entity type vocabulary. Semantic filtering raises the F1-score from 68.67% (ICL baseline) to 80.48% (SF), with dynamic exemplars pushing performance to 83.09%. Again, our approach substantially outperforms the state-of-the-art few-shot baseline of 74.90%, achieving an 8.19 percentage point increase. The consistent gains across both public datasets validate that our semantic filtering methodology generalizes effectively beyond conversational AI to traditional NER tasks.

6.3 Conversational NER Student Model

Table 3 presents student model performance when trained on teacher-labeled data, addressing whether knowledge from our semantic filtering methodology transfers effectively to production-ready models. Results show positive outcomes across all test sets. For conversational data, we observe impressive relative gains of 13.84% in English and 9.21% in Spanish. Remarkably, adding only 3.7k English conversational examples to our training dataset improves performance beyond conversational contexts: 6.51% average increase on head traffic and 6.86% on unseen entities across both languages. The positive results indicate that semantic understanding from our LLM teacher can be successfully distilled into smaller, faster production-ready models. The improvements on head traffic and unseen entities show that conversational data improvements transfer to command-like transactional settings, even for entities unseen during training.

7 Conclusions

We present a scalable framework for conversational Named Entity Recognition combining catalog-based entity grounding with LLM semantic filtering. Our approach transforms NER from a generative task into a constrained selection problem, enabling automated labeling of live traffic data without human annotation while ensuring factual accuracy through knowledge base grounding. We introduce an automated pipeline for generating multilingual conversational NER datasets reducing annotation costs while maintaining gold-standard quality. Our semantic filtering approach achieves 39.55% improvement on internal conversational data and 97.12% F1-score on CoNLL-2003, outperforming prior state-of-the-art by 24.82%. Student models trained with LLM-labeled traffic data show consistent improvements on both conversational and traditional transactional NER data. This work establishes a foundation for scalable conversational NER adapting to evolving entity catalogs while maintaining speed and reliability for real-time conversational AI systems. The methodology's generalization across domains and languages makes it broadly applicable to modern dialogue systems requiring sophisticated semantic understanding.

Limitations

Our in-production analysis reveals that despite these constraints, LLMs occasionally generate predictions outside the provided candidate set. While post-processing filters catch such violations, we observe that applying them yields performance improvements, suggesting some out-of-catalog predictions. However, predictions missing from our catalogs does not necessarily indicate errors as they may represent legitimate entities absent from our knowledge base. As future work, we could leverage such predictions to automatically update entity catalogs after manual or automatic validation, rather than simply discarding them. This could create a feedback loop that continuously improves catalog coverage based on real-world usage patterns.

Additionally, while our pattern generation pipeline demonstrates impressive results across multiple languages, it relies solely on English seed utterances. Although the LLM successfully generates grammatically and semantically appropriate patterns in target languages, this approach may introduce cultural bias and miss language-specific cultural cues in user requests. For instance, culturally-specific ways of requesting content may not be fully captured when patterns are grounded by English seeds. Future work should investigate the impact of using native-language seeds or culturally-diverse seed collections to better represent the cultural diversity of different user populations.

References

Anthropic. 2024. Model card addendum: Claude 3.5 haiku and upgraded claude 3.5 sonnet. https://www.anthropic.com/index/model-card-addendum-claude-3-5. Accessed: 2025-04-08.

Jiawei Chen, Yaojie Lu, Hongyu Lin, Jie Lou, Wei Jia, Dai Dai, Hua Wu, Boxi Cao, Xianpei Han, and Le Sun. 2023. Learning in-context learning for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13661–13675, Toronto, Canada. Association for Computational Linguistics.

Yi Chen and Liang He. 2023. SKD-NER: Continual named entity recognition via span-based knowledge distillation with reinforcement learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6689–6700, Singapore. Association for Computational Linguistics.

Alessio Cocchieri, Giacomo Frisoni, Marcos Martínez Galindo, Gianluca Moro, Giuseppe Tagliavini, and Francesco Candoli. 2025. Open-BioNER: Lightweight open-domain biomedical named entity recognition through entity type description. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 818–837, Albuquerque, New Mexico. Association for Computational Linguistics.

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

Elena Epure and Romain Hennequin. 2023. A human subject study of named entity recognition in conversational music recommendation queries. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1281–1296, Dubrovnik, Croatia. Association for Computational Linguistics.

Pratik Jayarao, Chirag Jain, and Aman Srivastava. 2018. Exploring the importance of context and embeddings in neural NER models for task-oriented dialogue systems. In *Proceedings of the 15th International Conference on Natural Language Processing*, pages 132–137, International Institute of Information Technology, Hyderabad, India. NLP Association of India.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jun-Yu Ma, Beiduo Chen, Jia-Chen Gu, Zhenhua Ling, Wu Guo, Quan Liu, Zhigang Chen, and Cong Liu. 2022. Wider & closer: Mixture of short-channel distillers for zero-shot cross-lingual named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5171–5183, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601.

Sourabh Majumdar, Serra Sinem Tekiroglu, and Marco Guerini. 2019. Generating challenge datasets for task-oriented conversational agents through self-play. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 693–702, Varna, Bulgaria. INCOMA Ltd.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205.

Subhadip Nandi and Neeraj Agrawal. 2024. Improving few-shot cross-domain named entity recognition by instruction tuning a word-embedding based retrieval augmented large language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 686–696, Miami, Florida, US. Association for Computational Linguistics.

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, and 1 others. 2023. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *Medicine*, 84(88.3):77–3.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 142–147.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Heydar Soudani, Roxana Petcu, Evangelos Kanoulas, and Faegheh Hasibi. 2024. A survey on recent advances in conversational data generation. *arXiv preprint arXiv:2405.13003*.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, and 1 others. 2025. Embeddinggemma: Powerful and lightweight text representations. *arXiv preprint arXiv:2509.20354*.

Rui Wang, Tong Yu, Junda Wu, Handong Zhao, Sungchul Kim, Ruiyi Zhang, Subrata Mitra, and Ricardo Henao. 2023. Federated domain adaptation for named entity recognition via distilling with heterogeneous tag sets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7449–7463, Toronto, Canada. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. 2025. GPT-NER: Named entity recognition via large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4257–4275, Albuquerque, New Mexico. Association for Computational Linguistics.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. Livebench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations*.

Xuan Zhou, Xiao Zhang, Chenyang Tao, Junya Chen, Bing Xu, Wei Wang, and Jing Xiao. 2021. Multi-grained knowledge distillation for named entity recognition. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5704–5716, Online. Association for Computational Linguistics.

A Dataset Statistics

Table 4 presents comprehensive statistics for the datasets used in our evaluation, including both internal datasets and external public benchmarks. For each dataset, we report the total number of samples, average utterance length in words, and the number of entity types.

B Impact of number of retrieved exemplars

Table 5 presents the relative F1-score improvements of our semantic filtering approach using varying numbers of retrieved exemplars. The number of exemplars indicated for each row is used during inference. The model consistently benefits from dynamic exemplar retrieval, with performance improvements evident even at 5 exemplars compared to semantic filtering alone (SF). Head traffic shows particular sensitivity to exemplar count, degrading significantly at 5 exemplars (-2.22%) but performing best at 10 (4.11%), suggesting high-frequency requests benefit from sufficient diverse examples to disambiguate common patterns.

We observe optimal performance with 10 exemplars, achieving the highest average improvement (22.24%) and strong results across all test sets: 37.79% on conversational data, 4.11% on head traffic, and 24.83% on unseen entities. Beyond this point, performance degrades at 20 exemplars (20.63% average), with declines across all metrics. We hypothesize that this decline is due to increasing noise from excessive context. With too many exemplars, less relevant examples may

Language	Dataset	Public	Total samples	Average length	Total entity types
English	**Conversational**	N	4,082	19.49	22
	Head Traffic	N	13,405	5.42	22
	Unseen Entities	N	12,283	9.01	22
	CoNLL-2003	Y	3,453	13.44	4
	OntoNotes 5.0	Y	8,262	18.48	18
Spanish	**Conversational**	N	3,925	14.99	22
	Head Traffic	N	4,951	3.75	22
	Unseen Entities	N	9,506	7.92	22

Table 4: Statistics for public and private test sets used for evaluation.

Model	Exemplars	Conversational	Head	Unseen	Avg
Baseline	✗	0.00%	0.00%	0.00%	0.00%
SF	✗	+36.83%	+3.72%	+19.70%	+20.08%
SF with Exemplars	5	+37.04%	-2.22%	+21.93%	+18.92%
SF with Exemplars	10	+37.79%	+4.11%	+24.83%	+22.24%
SF with Exemplars	20	+36.69%	+1.57%	+23.64%	+20.63%

Table 5: Relative performance difference (% points) compared to production baseline on internal English NER datasets using varying numbers of retrieved exemplars.

dilute the signal, making it harder for the model to identify the most pertinent patterns for the given query. Additionally, 20 exemplars substantially increase prompt length and inference latency without corresponding performance gains. Based on these findings, we use $k = 10$ exemplars for all reported experiments, balancing performance with computational efficiency

C NER Teacher Model Prompt

To ensure reproducibility, we provide the complete prompt template used for our NER Teacher Model. Table 6 presents the prompt template that performs entity selection through semantic filtering with catalog-grounded candidates. The template includes task definition, selection constraints, entity type descriptions, exemplars, and the target query with candidate entities. The template uses Jinja2 syntax for dynamic content insertion.

D Data Generation Prompt

We provide the complete prompt template used in our conversational NER data generation pipeline. Table 7 shows the prompt template used to create conversational NER patterns with entity-type placeholders. The template includes critical instructions for entity replacement, allowed and forbidden tags, correct and incorrect replacement examples, and detailed pattern generation rules. The template uses Jinja2 syntax for dynamic content insertion.

```
You are an NER (Named Entity Recognition) tool that performs entity extraction by selecting from pre-identified candidate entities.
<task>
Your task is to SELECT ONLY from the provided candidate entities based on semantic relevance and contextual appropriateness. You are given a query and a list of candidate (entity, entity-type) pairs. Your job is to choose which candidates are semantically relevant to the query.
CRITICAL: This is a SELECTION task, NOT a generation task. You must ONLY select from the exact candidate pairs provided below. Each output entry must be an EXACT COPY of a candidate entry.
</task>

<task_specific_instructions>
  {{ task_specific_instructions }}
</task_specific_instructions>

<output_instructions>
SELECTION CONSTRAINTS:
- You MUST ONLY select from the exact {"span": "X", "label": "Y"} pairs provided in the candidates section
- Every entry in your output MUST be an exact copy of a candidate entry
- NEVER generate new spans, labels, or combinations not present in candidates
- If no candidates are semantically relevant, output an empty array []

OUTPUT FORMAT REQUIREMENTS:
- Return ONLY a raw JSON array of objects with no markdown formatting
- DO NOT include “‘json markers, explanations, or introductory text
- The output must be directly parseable by json.loads()
- Each object must be an exact copy from the candidates section
</output_instructions>

<entity_types>
{% - for tag, description in ner_labels.items()}  {{ tag }} : {{ description }}
{% - endfor %}
</entity_types>

<examples>
{% for example in examples %}
<example>
Input: "{{ example.query }}"

<candidates>
{%- for candidate in example.catalog_matches %}
<candidate>
{"span": "{{ candidate.span }}", "label": "{{ candidate.label }}"}
</candidate>
{%- endfor %}
</candidates>

Output: [{%- for response in example.response %} {"span": "{{ response.span }}", "label": "{{
response.label }}"}, {%- endfor %}]
</example>
{% endfor %}
</examples>

Input: "{{ query }}"

<candidates>
{%- for candidate in candidates %}
<candidate>
{"span": "{{ candidate.span }}", "label": "{{ candidate.label }}"}
</candidate>
{%- endfor %}
</candidates>

Output:
```

Table 6: Complete NER prompt template used for entity selection task. The template includes task definition, selection constraints, entity type descriptions, few-shot examples, and the target query with candidate entities. Template variables use Jinja2 syntax (shown in {{ }} and are populated at runtime. Structural sections are delimited with XML-like tags (e.g., <task>, <candidates>).

<task>
Your task is to generate patterns in {{ target_lang }} starting from a set of customer requests provided between <seeds> and </seeds>. The generated patterns should be representative of the seeds semantically and share their intent. The generated patterns should be made in a conversational or natural request manner.
These patterns will be used to create Named Entity Recognition (NER) datasets. For this reason, instead of containing the actual named entities in the seeds, they should contain the tag representing the entity-type provided in <entity_types>. These tags are the NER label so restrict entity types to the list provided below. This will enable us to replace them with our own entities of interest in a flexible and scalable manner. Adhere to the provided instructions.
</task>

<critical_instructions>
IMPORTANT: ONLY REPLACE SPECIFIC NAMED ENTITIES WITH TAGS, NEVER GENERIC WORDS

ALLOWED ENTITY TAGS - USE ONLY THESE EXACT TAGS AND NO OTHERS:
{%- for tag in allowed_tags %}
  - <{{ tag }}>
{%- endfor %}

FORBIDDEN TAGS - NEVER USE THESE TAGS OR ANY TAGS NOT PROVIDED IN THE ALLOWED ENTITY TAGS LIST:
{%- for tag in forbidden_tags %}
  - <{{ tag }}>
{%- endfor %}
  - Any other tag not in the ALLOWED list above

CORRECT REPLACEMENTS:
{%- for example in correct_replacements %}
  - "{{ example.text }}" → <{{ example.tag }}>
{%- endfor %}

INCORRECT REPLACEMENTS (DO NOT DO THESE):
{%- for example in incorrect_replacements %}
  - "{{ example.text }}" → <{{ example.tag }}> (WRONG! {{ example.reason }})
{%- endfor %}

STRICTLY FORBIDDEN PATTERNS (NEVER GENERATE THESE):
<negative_examples>
{%- for example in negative_examples %}
  {{ loop.index }}. "{{ example }}"
{%- endfor %}
</negative_examples>

THE RULE IS SIMPLE:
{%- for rule in entity_specific_rules %}
  - <{{ rule.tag }}> ONLY replaces {{ rule.description }}
{%- endfor %}

REMEMBER: Tags are ONLY for replacing SPECIFIC NAMED ENTITIES, not generic concepts or common nouns.

OUTPUT FORMAT REQUIREMENTS:
- Only generate patterns in the target language: {{ target_lang }}
- Return ONLY a raw JSON array of strings with no markdown formatting
- DO NOT include “‘json or “‘ markers
- DO NOT include any explanations or comments
- DO NOT include any introductory text like "Here is a JSON array ..."
- The output should be directly parseable by json.loads()
</critical_instructions>

<rules>
{%- for rule in rules %}
{{ loop.index }}. {{ rule.text }}
{%- endfor %}
</rules>

<seeds>
{%- for seed in seeds %}
  {{ seed }}
{%- endfor %}
</seeds>

Output:

Table 7: Complete pattern generation prompt template using Jinja2 syntax for dynamic content insertion. All template variables are populated at runtime to generate conversational NER patterns in the target language.