Auto311: A Confidence-Guided Automated System for Non-emergency Calls

Zirong Chen, Xutong Sun, Yuanhe Li, Meiyi Ma

Department of Computer Science, Vanderbilt University, Nashville, Tennessee 37235, USA
{zirong.chen, xutong.sun, yuanhe.li, meiyi.ma}@vanderbilt.edu

Abstract

Emergency and non-emergency response systems are essential services provided by local governments and critical to protecting lives, the environment, and property. The effective handling of (non-)emergency calls is critical for public safety and well-being. By reducing the burden through non-emergency callers, residents in critical need of assistance through 911 will receive fast and effective response. Collaborating with the Department of Emergency Communications (DEC) in Nashville, we analyzed 11,796 non-emergency call recordings and developed Auto311[1], the first automated system to handle 311 non-emergency calls, which (1) effectively and dynamically predicts ongoing non-emergency incident types to generate tailored case reports during the call; (2) itemizes essential information from dialogue contexts to complete the generated reports; and (3) strategically structures system-caller dialogues with optimized confidence. We used real-world data to evaluate the system's effectiveness and deployability. The experimental results indicate that the system effectively predicts incident type with an average F-1 score of 92.54%. Moreover, the system successfully itemizes critical information from relevant contexts to complete reports, evincing a 0.93 average consistency score compared to the ground truth. Additionally, emulations demonstrate that the system effectively decreases conversation turns as the utterance size gets more extensive and categorizes the ongoing call with 94.49% mean accuracy.

Introduction

Emergency and non-emergency response systems are essential services provided by local governments and critical to protecting lives, the environment, and property. While 911 is primarily used for emergency services, 311 is a non-emergency phone number that people can call to find information about municipal services, make complaints, or report problems like stolen property, road damage, etc. Both emergency and non-emergency calls are operated by the Department of Emergency Communication (DEC) in most cities. DECs across the nation receive an overwhelmingly high number of calls, with the national yearly average of 911 calls being close to 240 million (NYC911 2022; Ma et al. 2019). The growing use of response systems comes at a time when local governments face increasing pressure to do more with fewer resources. Indeed, the number of local government employees in the United States has shrunk by nearly 5% in 2021 (Saxon et al. 2022), and a large proportion of counties and municipalities anticipate a significant general fund shortfall as the United States transitions out of the COVID pandemic (Afonso 2021).

To mitigate this issue, we introduce Auto311, the first automated system to handle 311 non-emergency calls. Auto311 features two key components: incident type detection and information itemization. It identifies probable incident types from ongoing calls and dynamically generates and updates case reports accordingly. Simultaneously, these reports direct the information itemization module to gather necessary information, streamlining the process.

Previous works have aimed to optimize (non-)emergency management (Sun, Bocchini, and Davison 2020; Wex et al. 2014; Manoj and Baker 2007; Ma et al. 2021a; Ma, Stankovic, and Feng 2021, 2018; Chen et al. 2008). However, most of those works focus more on the emergency resource allocation after one case report is placed, for example, route optimization for ambulances and response center localization for faster responses (Mukhopadhyay et al. 2022). Although other available could frameworks like AWS's Lex[2] and Google's DialogFlow[3] can set up automated dialogue service within a few hours, they require clear and relatively fixed dialogue charts to guide the conversation. When it comes to non-emergency call handling, leaving alone privacy and safety issues brought by online solutions, most incident types have unique dialogue charts compared to others, making it unrealistic to separately handle the conversation for each incident type. However, Auto311, at a system level, takes full advantage of the emitted confidence scores of each component to strategically optimize the dialogue.

However, developing Auto311 poses some key **challenges**. First, unlike plenty of work that has been done to solve text classification problems (Kowsari et al. 2019; Mirończuk and Protasiewicz 2018; Minaee et al. 2021; Aggarwal and Zhai 2012) with outputting the most likely one category in the end, the incident prediction in this task has to cope with calls involving multiple incident types instead,

[1]Code and Demo: https://github.com/AICPS-Lab/Auto311

[2]AWS Lex: https://aws.amazon.com/lex/

[3]Diagflow: https://cloud.google.com/dialogflow?hl=en

see Section Motivating Study for more details. Second, although measuring confidence in machine learning models has become more and more popular recently, refer to Section Related Work for more details, there still lacks an effective method to measure the confidence score behind the model outputs in a textual format. Lastly, although pretrained models yield satisfying performance on datasets with general purposes, e.g., Bert (Devlin et al. 2018) on SQuAD (Rajpurkar et al. 2016), Auto311 has to align with more task-specific data and goals under the non-emergency call handling scope. We summarize our **contributions** as follows:

- We annotate and analyze 11,796 authentic audio recordings of non-emergency calls.
- We build Auto311, the first confidence-guided automated system to handle non-emergency calls by navigating the conversation with optimized confidence, dynamically predicting incident type, and generating reports with key information.
- We evaluate the performance of Auto311 using real-world recordings of non-emergency. It achieves an average F-1 score of 92.54% on the incident type prediction and an average score of 0.93 for information itemization.
- We emulate the usage of Auto311 using recordings from our dataset and analyze Auto311's system-level impacts. Auto311 dynamically adjusts to shifting incident types, reduces follow-up conversations, and yields an overall average accuracy of 94.49% for categorizing the emulated call utterances.

Motivating Study

We annotate and analyze 11,796 real-world recordings of non-emergency calls, and make two important observations.

Unintentional Additional Information Examination of audio transcriptions indicates callers tend to provide supplementary details beyond the dispatcher's specific inquiries. We found that 72% of callers offer extra information, exceeding the question's scope in our dataset. Notably, this additional information can enhance the precision and comprehensiveness of the emergency response system. See the example below with the caller's personal information removed:

Dispatcher: Metro Nashville Police, Fire, and Medical, what is the location of your emergency? Caller: Oh, I'm not sure if this is an emergency. I am *#name*, *#phone_number*. The address is *#address*. It's the King Buffet. I saw a customer out in the parking lot smoking crackpipes in front of all the customers.

In this conversation turn, although the dispatcher only inquires about location, the caller provides additional details like name, phone number, and suspicious activity, suggesting a drug-related case. Strategically leveraging such extra information could optimize emergency response conversations by proactively addressing potential follow-up questions, thus streamlining the interaction.

Shifting and Multiple Incident Types Call recordings show that ongoing incidents mentioned by callers occasionally encompass multiple incident types (at a rate of 38.27%). Callers also tend to modify incident types as they uncover more details. For instance, phrases like "someone busted my car and my wallet is gone" indicate two incident types – damaged and lost stolen property. Similarly, in "I saw a car illegally parked... oh, wait, it's abandoned because the bumper is off and rusted," the caller initially reports illegal parking, then recognizes it as an abandoned vehicle based on new details. This underscores the need for our system to recognize various incident types concurrently and adapt to evolving conversations.

Overview of Auto311

Auto311 is designed to automatically handle non-emergency calls by engaging in interactive conversations with callers. An outline of our system's structure is shown in Figure 1. The system comprises five key components: the *conversational interface* for interacting with callers, the *handover control* to transfer calls to human operators if needed, the *incident type prediction* module to identify probable incident types, the *information itemization* module to organize details in case reports, and the *confidence-guided report generation and dialogue optimization* module for generating informed reports and optimizing subsequent conversations using confidence guidance.

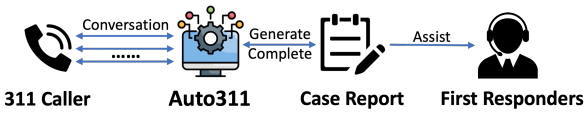

Figure 1: Auto311 in Emergency Response

At runtime, as shown in Figure 2, when a caller initiates contact, the *conversational interface* starts the dialogue with opening questions, collecting essential information like the caller's name and incident location. Each response from the caller forms an utterance. Subsequently, the *handover control* function evaluates whether human operator intervention is required for the call. The call proceeds to subsequent modules only if the handover is not needed. At the same time, the *incident type prediction* module forecasts the most likely incident type(s) for report generation, while the *information itemization* module provides potential details from the caller's utterance to fill the report's sections. Additionally, the *confidence-guided report generation and dialogue optimization* module constantly monitors confidence scores to ensure the report is filled with high confidence, thereby optimizing follow-up conversations.

Importantly, the case report referred to here differs from the question list within the conversational interface. The question list stores upcoming inquiries for the interface, while the case report shapes the subsequent dialogue by updating the question list with its incomplete sections after each conversation turn.

Conversational Interface

The conversational interface supports both voice and text inputs. Employing state-of-the-art audio transcription tools, notably the OpenAI Whisper model (OpenAI 2022), this

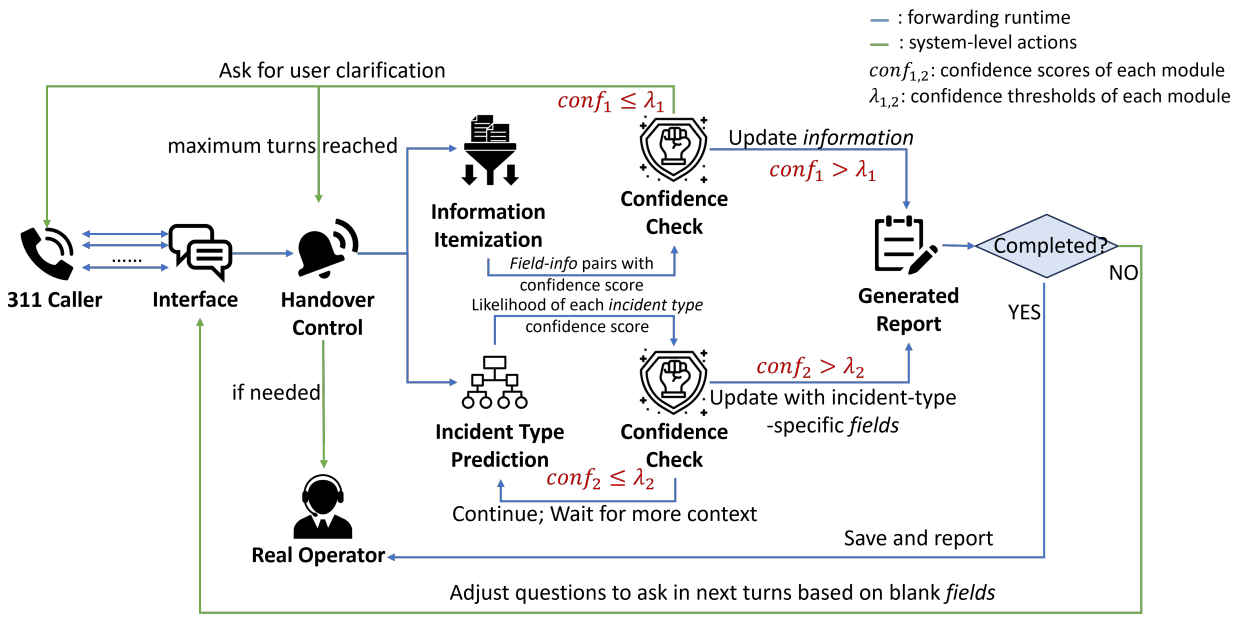

Figure 2: Confidence-guided System Design

interface adeptly transforms speech into text, accommodating a range of accents. For text-to-speech functionality, we harness advanced audio generation tools like the Suno-AI Bark model (suno-ai 2023), acclaimed for producing realistic voices within a lightweight model framework. Beyond speech-to-text and text-to-speech conversion, the interface engages in conversations using a dynamically updated question list. This list determines the query sequence and guides the interface's speech-to-text conversion process.

Always-on Handover Control

The handover control module remains active throughout runtime, redirecting calls to human operators when necessary. Collaborating with DEC, we identify specific scenarios that activate this module: (1) downstream module exceptions, like uncertain information; (2) caller's repeated request for human interaction; (3) proactive alerts for potential urgency. The first case is managed within system actions. For example, when Auto311 seeks clarification due to uncertain details, it limits such queries to three turns. Exceeding this threshold triggers exceptions and activates handover control. Addressing the other two cases, we develop an interpretable rule-based detection mechanism, prioritizing interpretability and control. Using Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003), we curate a sensitive word list through manual review. Our approach combines natural language processing (NLP) features (Bird, Loper, and Klein 2009), such as stemming, lemmatization, part-of-speech tags, and shallow parsing, with custom patterns to establish rules activating handover control, thus ending system interaction. More details are available in the appendix. Note that patterns and sensitive words are not exhaustive, allowing future expansion of trigger conditions. However, the broader process is beyond this paper's scope.

Incident Type Prediction

The incident type prediction module utilizes contextual information from previous caller utterances. The module takes the overall context, covering all prior utterances as input. Since a context can involve multiple incident types, we apply a multi-layer hierarchical structure and bootstrap-like procedure for classification. This tracks the possibility of the call belonging to each incident type (see Confidence-guided System Design for details). The hierarchical structure and iterative procedure enable the prediction module to handle multiple incident types per call using full conversational context.

Information Itemization

The information itemization module completes empty case report sections by quoting the caller's utterances. This involves narrative fields seeking explanatory details and yes/no fields confirming facts. For narratives, we leverage extractive question-answering frameworks - the blank fields are inputs, and outputs quote relevant caller utterances. Yes/no fields become binary classification, predicting yes or no from the last utterance. Unlike incident prediction using all contexts, itemization considers only the latest utterance.

Confidence-guided Report Generation and Dialogue Optimization

This module updates the report and optimizes dialogues as the conversation progresses. See technical details in Section Confidence-guided System Design.

Confidence-guided System Design

This section delves into the technical aspects of confidence guidance within Auto311. Firstly, we explain the method to derive confidence scores from the machine learning models.

Secondly, we elucidate the purpose of the generated confidence scores within the workflow.

Confidence Measurement

We define confidence as consistency over multiple trials with the same inputs. We leverage Monte Carlo Dropout(Gal and Ghahramani 2016; Ma et al. 2021b) to generate the confidence prediction. Specifically, dropout was set as active at test time and assesses consistency across trials to obtain scores. Preset thresholds determine if outputs are confident - meeting or exceeding the threshold means confident. Auto311 applies machine learning models to handle two major tasks component-wisely: incident type prediction and information itemization. Here we detail Auto311's approach to measuring confidence.

Confidence in Incident Type Prediction To address potential multiple incident types within a call, we employ a multi-layer hierarchy structure coupled with a bootstrap-like process. The initial layer of this structure involves training a neural network to assess if the present context corresponds to the most common incident type. Subsequently, the second layer trains another neural network to determine if the context aligns with the second most frequent incident type, excluding the previously identified type. This pattern continues for subsequent types. As a result, (1) the structure can identify all possible incident types within a call; (2) each category operates independently, facilitating future adjustments based on new data or expansion to more categories. At runtime, we establish confidence scores by measuring consistency across output distributions for the same input, utilizing active dropout to quantify prediction uncertainty. The hierarchical cascade structure adeptly handles multiple types, while confidence scoring measures prediction certainty.

Confidence in Information Itemization Regarding textual outputs for information itemization, determining confidence necessitates a consistency assessment between texts. Traditional text comparison methods often prioritize aspects like edit distances and length. See details in Related Work. However, our collaboration with DEC underscores the value of succinct outputs with ample details for case reports. Consider the scenario where the module generates "on the 2525 West End Ave" while the ground truth is "2525 West End Ave." Traditional methods yield low scores, such as 0.5 from BLEU-bigram. However, when dispatchers gather incident location details, these outputs should exhibit high consistency due to matching location keywords and similar semantics. To address this, we adopt a new approach. For keywords, we employ an unsupervised state-of-the-art keyword extractor (Campos et al. 2018) to extract key segments from model outputs. The overlap between keyword segment lists is calculated. For semantics, we utilize SentenceBERT (Reimers and Gurevych 2019) to project each output into a latent space, assessing the similarity between represented textual string lists. The overall score, calculated via Polyak averaging (p=0.2), integrates keyword overlaps and embedding distances. This metric enables us to gauge consistency and generate a confidence score.

Report Generation and Dialogue Optimization

Auto311 is designed to be confidence-guided. With maximum confidence guaranteed, it (1) dynamically updates case reports at every turn of the conversation and (2) guides the follow-up conversation based on the generated case report. See Figure 2 for more detailed system logic.

Confidence drives precise report completion via information itemization. Using the latest caller utterance, it populates report details. Textual output confidence is determined through text comparison. As in Figure 2, uncertainty ($conf_1 \leq \lambda_1$) prompts Auto311 to seek clarification for guiding questions, capped at three turns before handover control. Confidence ($conf_1 > \lambda_1$) skips further dialogue optimization for filled fields. Confidence also aids the incident type prediction module's adaptability and report efficiency. Using complete utterance context, it predicts likely incident types. As shown in Figure 2, low confidence ($conf_2 \leq \lambda_2$) excludes uncertain predictions from reports. High confidence ($conf_2 > \lambda_2$) incorporates predictions. Systemically, prediction persists each turn, even post early confident identification. This iterative process tracks trends in types and scores, updating reports on confidence drops ($conf_2 \leq \lambda_2$). Confidence-driven adaptation detects and responds to evolving incident types during calls, further updating reports.

Previous component confidence scores further optimize future dialogues. In the case study illustrated in Figure 3, when a new caller utterance is received, Auto311 identifies fields to complete in the case report (e.g., incident-address, caller-name, caller-phone). High-confidence completion marks them as done. Unfinished fields guide subsequent questions (e.g., requesting a callback number if caller-phone is missing). Concurrently, incident type is confirmed if confidence surpasses a set threshold (0.85 in Figure 3). With the type established, specialized fields are identified (e.g., property description for lost/stolen cases), updating the report. Auto311 then prioritizes shared general fields (e.g., property description, time, ownership status), streamlining dialogue to focus initially on universal details. This avoids asking about the damage nature before finalizing the incident type, which applies only to damaged property cases. This optimization confirms type(s) with more context. In the example, the next turn's details indicate a shift to "lost/stolen" only. The report is updated accordingly. The dialogue concludes when all report fields are complete. Confidence scoring thus optimizes the flow by collecting universal details first and adapting to emergent incident types.

Evaluation

Experiments assess the performance of (1) confidence-guided incident prediction, (2) confidence-guided itemization, and (3) the overall system. Our dataset includes 11,796 non-emergency calls from the DEC in Nashville, TN. Metrics for incident type prediction are precision, recall, F-1, and accuracy. The newly introduced text comparison method evaluates itemization module outputs. The experiments were run on a machine with 2.50GHz CPU, 32GB memory, and Nvidia GeForce RTX 3080Ti GPU.

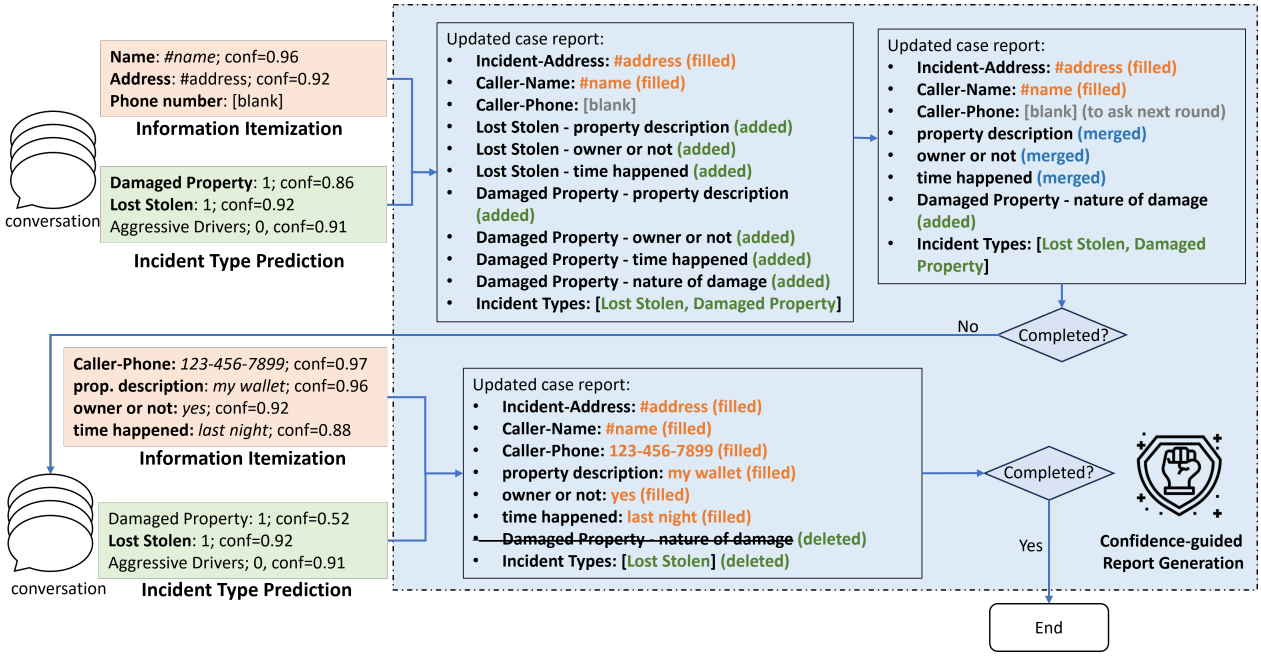

Figure 3: A General Case Study in Confidence Guidance

	Minor Crash	Lost Stolen	Aggressive Drivers	Check Welfare	Damaged Property	Noise Violation	Roadway Hazard	Abandoned Vehicles	Drug-Pros Activity
LSTM	56.47%	0.00%	53.85%	82.35%	0.00%	0.00%	63.83%	69.23%	46.15%
CNN	75.86%	85.71%	72.72%	81.08%	40.00%	44.44%	80.00%	62.07%	25.00%
RCNN	90.57%	82.35%	61.54%	82.35%	76.92%	83.33%	86.67%	26.09%	25.00%
RNN	63.33%	40.00%	52.71%	73.33%	44.44%	83.33%	74.29%	28.57%	26.67%
Self-Attn	88.46%	88.89%	66.67%	87.50%	66.67%	50.00%	81.25%	9.52%	35.29%
Attention	91.69%	62.50%	50.00%	53.66%	54.55%	60.00%	81.08%	66.67%	62.50%
Bert	95.04%	91.50%	92.31%	90.00%	88.89%	83.33%	90.91%	94.12%	92.40%
Auto311	**95.71%**	**93.00%**	**93.75%**	**90.00%**	**94.12%**	**83.33%**	**90.91%**	**94.12%**	**92.40%**

Table 1: Averaged performance (F-1) over 30 trials on incident type prediction

Confidence-guided Incident Type Prediction

This section aims to evaluate Auto311's performance on incident type prediction. Baselines use various neural networks like LSTM, CNN, RCNN, Self-Attention, Bahdanau's Attention, and BERT (Hochreiter and Schmidhuber 1997; Kim 2014; Lai et al. 2015; Vaswani et al. 2017; Bahdanau, Cho, and Bengio 2014; Wolf et al. 2019)(see Table 1). We include 9 categories due to the page limit (full results including standard deviation stats in Appendix). Experiments comprehensively assess prediction with different model architectures on real call data.

Analysis shows traditional models like CNN perform poorly, with just 62.99% average F-1 on these 9 types. Transcription diversity from varied callers increases task complexity (e.g., different speaking habits), challenging learning without prior knowledge. BERT surpasses other models, with 92.54% average F-1 and 100% max across all 11 types. Leveraging BERT's pre-trained weights and confidence guidance, Auto311 further improves BERT F-1 from 91.50% to 93.00% for lost/stolen cases. Results demonstrate prediction difficulties due to call diversity and Auto311's enhancements over BERT using confidence scoring.

In summary, based on the results, Auto311 effectively dispatches the ongoing call to the given incident types. In terms of F-1 score, the BERT backend has the most competitive results. Confidence guidance further improves performance.

Confidence-guided Information Itemization

This experimental setup assesses the performance of the information itemization module of Auto311 using various backends (DistilBERT, BERT, RoBERTa, LongFormer, BigBird (Sanh et al. 2019; Beltagy, Peters, and Cohan 2020; Zaheer et al. 2020)) and benchmarks (SQuAD, CUAD, TriviaQA (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018; Hendrycks et al. 2021; Joshi et al. 2017)) and compares it to large language models (LLMs) like GPT3.5 and 4[4], with re-

[4]Prompt: "*I will provide context and a set of questions. Please respond to the questions using exact quotes from the context. Your answers should be concise and comprehensive. Context: ...; Question Set: ...*"

sults in Table 2. We utilize our consistency score in this evaluation. Two types of data samples are evaluated for performance: random test samples from archived data (collected by the end of 2022) and the latest data samples (collected from the beginning of 2023) from the call center. Recognizing that city-related information evolves (e.g., new places, activities), we simulate Auto311's usage under knowledge evolution by assessing performance on the latest data samples. Furthermore, we assess Auto311's performance in information itemization when queried with various fields, encompassing basic fields, less specific fields, and more specific fields. Basic fields include essential data like incident location, less specific fields cover descriptors like vehicle and human/suspect descriptions, and more specific fields pertain to incident-specific details such as the timing of the incident (DamagedProperty-when).

	Archived (test)	Latest (runtime)
DistilBERT-SQuAD2	0.5546	0.5330
BERT-SQuAD2	0.2422	0.2791
RoBERTa-SQuAD2	0.1172	0.2581
RoBERTa-CUAD	0.2188	0.2378
LongFormer-TriviaQA	0.3260	0.1424
BigBird-TriviaQA	0.5289	0.5015
GPT3.5 (June 2023)	0.6343	0.6529
GPT4 (June 2023)	0.6578	0.6264
Auto311	**0.9329**	**0.8605**

Table 2: Performance on information itemization

Table 2 yields the following insights: (1) pretrained model limitations: DistilBERT and BERT struggle in current non-emergency dispatch scenarios, performing notably lower than other methods. For instance, BERT pretrained on SQuAD achieves only 0.2422 on the test batch; (2) LLMs vs. Auto311: Despite LLMs' general NLP success, Auto311 consistently outperforms them on both datasets. For example, Auto311's performance surpasses GPT3.5 by 47% on archived samples and 41% on the latest samples; (3) adaptation to evolving knowledge: Auto311's 37% performance lead over GPT4, despite a minor drop, underscores its proficiency in capturing evolving local city knowledge.

Table 3 highlights how confidence guidance enhances itemization across field types, e.g., improving consistency from 0.8255 to 0.9164 for aggressive driver behavior details. Auto311 achieves 100% recall for binary questions, correctly predicting traffic blockage and in-person meetup needs. In real-world non-emergency scenarios, prioritizing high recall ensures comprehensive coverage of potential requests. Results showcase confidence scoring's role in optimizing itemization, particularly for critical binary fields.

In summary, the results underscore the difficulty of the information itemization task for both pretrained models and existing LLMs. Auto311's fine-tuning on our dataset effectively integrates task-specific knowledge, leading to competitive performance surpassing LLMs. Moreover, through runtime emulation, Auto311 adapts to evolving city knowledge, indicating potential for long-term deployment. Additionally, confidence guidance empowers Auto311 to enhance the completion of various case report fields.

System Level Performance

The subsequent experiments focus on assessing Auto311's system-level performance. Due to data limitations, audio recordings only capture fixed dialogue paths, preventing direct interaction with the same caller in identical call scenarios. Instead, we emulate conversations by merging utterance segments. This emulation facilitates evaluating Auto311's capabilities in two aspects: (1) assessing its management of changing incident types and (2) analyzing the optimization of follow-up dialogues during emulation.

Adjustments to Shifting Incident Types We evaluate Auto311's adaptability to shifting incident types. Using common shifts observed in the dataset (see Figure 4), we emulate conversations where the caller firmly states type A (blue lines and grey regions), then adds type-specific details for type B (orange lines and regions), indicating a type shift. These experiments assess Auto311's real-time adaptation to emergent types in emulated interactions.

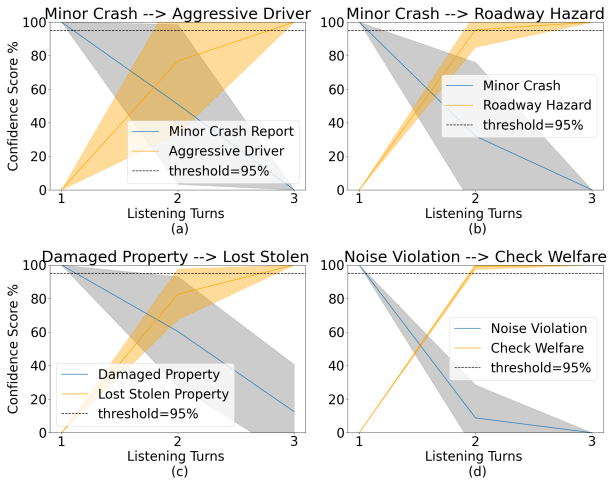

Figure 4: Confidence Changes in Shifting Incident Types

From Figure 4, we observe that first, Auto311 handles all four major shifting situations in three future turns with more type-specific descriptions being fed as input to the incident type prediction module. Second, with the introduction of confidence guidance, Auto311 adjusts the prediction results to align with the shifting trend dynamically.

In summary, Auto311 adeptly handles shifting incident types in simulations, updating its understanding and creating optimized reports with more specific information over 3 follow-up turns.

Optimizations to Upcoming Dialogues We emulate Auto311 usage by composing caller utterances and assessing the relationship between saved turns and utterance size (see Figure 5). Utterance size represents the count of past segments included. Across 100 emulations per size, we monitor saved turns and categorization accuracy. These experiments evaluate Auto311's ability to optimize dialogues through accumulated context and to make type predictions effectively.

	Basic Fields			Less Specific Feilds		
	Inc-Loc	Caller-Name	Caller-Phone	Veh Desc	Human/Suspect Desc	Prop Desc
w/o Conf Guide	0.9035	0.9478	1.0000	0.8538	0.9678	0.9512
w/ Conf Guide	**0.9631**	**0.9962**	**1.0000**	**0.9104**	**1.0000**	**1.0000**

	More Specific Fields								
	DamgProp -When	AggDriver -Behavior	CheckWel -Relation	MinorCrash -BlockTraffic (Y/N)			CheckWel -InpersonMeet (Y/N)		
				P	R	F-1	P	R	F-1
w/o Conf Guide	0.9045	0.8255	0.9023	66.67%	100.00%	80.0%	83.33%	83.33%	83.33%
w/ Conf Guide	**1.000**	**0.9164**	**0.9855**	**88.89%**	**100.00%**	**94.12%**	**83.33%**	**100.00%**	**90.91%**

Table 3: Auo311's performance on different fields

Longer composed utterances contain more itemizable details. The blue line indicates Auto311's saved turns during emulation, while the light blue region represents total information provided. The average and maximum real-world utterance lengths are denoted by green and red dashed lines. Emulation demonstrates Auto311 effectively using additional information in caller utterances to minimize follow-up turns. Furthermore, Auto311 achieves a 94.49% accuracy (not shown in Figure 5) when handling composed utterances.

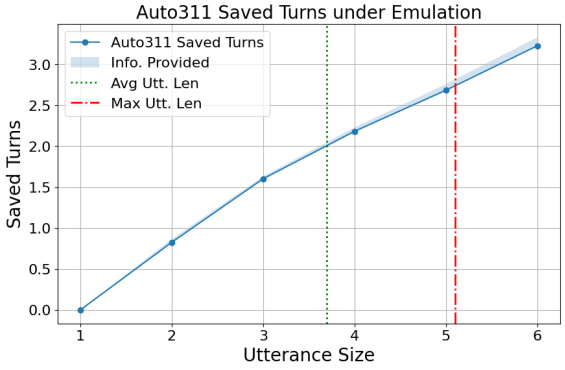

Figure 5: Emulated Usage of Auto311

In summary, these emulations show our solution, at a system level, not only piratically optimizes future conversations by utilizing additional information provided in caller utterances but also effectively categorizes the potential incident types with an overall accuracy of 94.49%.

Related Work

Question Answering and Large Language Models. In recent years, advanced question-answering systems have evolved across various scenarios (Chen et al. 2023, 2022b,a; Diefenbach et al. 2018). Black-box abstractive QA systems like mBART and T5 (Chipman et al. 2022; Raffel et al. 2019) lack output control. Although large language models, like Claude[5], especially for QA dialogues (Brown et al. 2020; Ouyang et al. 2022), gain attention, we still argue that prompt-based models are unsuitable for emergency response due to compromised input preservation and advocate for a transparent, controllable offline approach prioritizing reliability and decision transparency.

Confidence Scores in Machine Learning. While significant efforts have been dedicated to assessing the model confidence (Poggi, Tosi, and Mattoccia 2017; Hüllermeier and Waegeman 2021; Poggi and Mattoccia 2016), we redefine confidence as internal consistency across identical inputs, deviating from common definitions. For text classification like call dispatching, this consistency is seen in distributional shifts. However, quantifying and analyzing output text changes across domains remains challenging in current information itemization setups. Most open-source QA models provide confidence scores for single runs, like HuggingFace's Bert-QA (Face 2022), which measures confidence in a single trial through simple multiplication of softmax distributions. Hence, a robust confidence measurement mechanism for Auto311 in incident type prediction and information itemization is crucial.

Metrics for Text Comparison. Many text comparison metrics are unsuitable for Auto311's information itemization. For our goal of concise, detailed outputs that allow deviations from the ground truth, metrics like Damerau-Levenshtein distance (Damerau 1964) and BLEU (Papineni et al. 2002) fall short. N-gram metrics like ROUGE (Lin 2004) and WER lack semantic understanding. Although end-to-end metrics like embeddings and learned metrics (Reimers and Gurevych 2019; Cer et al. 2018; Artetxe et al. 2019) consider semantics, they misalign with our criteria and lack interpretability and generalization in emergency response. Thus, a metric that gauges key information coverage from user utterances while meeting dispatch center requirements becomes essential.

Summary

In this paper, we introduce Auto311, the first automated system tailored for non-emergency call management. Our evaluations with real-world and emulated interactions show strong performance in (1) incident type prediction, (2) case report generation, and (3) enhanced follow-up conversations using confidence-based guidance. In future work, we will enhance Auto311 and deploy it to handle non-emergency calls in the real world. By reducing the burden through non-emergency callers, residents in critical need of assistance through 911 will receive a fast and effective response.

[5]Anthropic Claude: https://claude.ai

Acknowledgements

This material is based upon work supported by the National Science Foundation (NSF) under Award Numbers 2228607. This work is a collaborative effort, and we are grateful for the support and contributions of everyone involved. In particular, we would like to express our gratitude for the valuable input and expertise provided by Stephen Martini, Director of the Department of Emergency Communication, and his team throughout the project. We also sincerely appreciate Keith Durbin, Chief Information Officer and Director of Information Technology Services, and Colleen Herndon, Assistant Director of GIS & Data Insights, Information Technology Services for the Metropolitan Government of Nashville and Davidson County, for their valuable collaboration and insights.

References

Afonso, W. 2021. Planning for the unknown: Local government strategies from the fiscal year 2021 budget season in response to the COVID-19 pandemic. *State and Local Government Review*, 53(2): 159–171.

Aggarwal, C. C.; and Zhai, C. 2012. A survey of text classification algorithms. *Mining text data*, 163–222.

Artetxe, M.; Schwenk, H.; Marquez, L.; and Cho, K. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. 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)*, 3293–3302.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Bird, S.; Loper, E.; and Klein, E. 2009. NLTK: The Natural Language Toolkit. http://www.nltk.org/. Online; accessed [insert date].

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3: 993–1022.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A. M.; Nunes, C.; and Jatowt, A. 2018. YAKE! Collection-Independent Automatic Keyword Extractor. In *European Conference on Information Retrieval*. Springer.

Cer, D.; Yang, Y.; Kong, S.-y.; Hua, N.; Limtiaco, N.; John, R.; Ngu, A.; Christopher, A.; Constant, N.; Guajardo-Cespedes, M.; et al. 2018. Universal Sentence Encoder. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 169–174.

Chen, R.; Sharman, R.; Rao, H. R.; and Upadhyaya, S. J. 2008. Coordination in emergency response management. *Communications of the ACM*, 51(5): 66–73.

Chen, Z.; Li, I.; Zhang, H.; Preum, S.; Stankovic, J. A.; and Ma, M. 2022a. Cityspec: An intelligent assistant system for requirement specification in smart cities. In *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, 32–39. IEEE.

Chen, Z.; Li, I.; Zhang, H.; Preum, S.; Stankovic, J. A.; and Ma, M. 2023. CitySpec with shield: A secure intelligent assistant for requirement formalization. *Pervasive and Mobile Computing*, 92: 101802.

Chen, Z.; Li, I.; Zhang, H.; Preurn, S.; Stankovic, J. A.; and Ma, M. 2022b. An Intelligent Assistant for Converting City Requirements to Formal Specification. In *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, 174–176. IEEE.

Chipman, H. A.; George, E. I.; McCulloch, R. E.; and Shively, T. S. 2022. mBART: multidimensional monotone BART. *Bayesian Analysis*, 17(2): 515–544.

Damerau, F. J. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3): 171–176.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Diefenbach, D.; Lopez, V.; Singh, K.; and Maret, P. 2018. Core techniques of question answering systems over knowledge bases: a survey. *Knowledge and Information systems*, 55: 529–569.

Face, H. 2022. BERT: Pre-trained Transformer for Question Answering. Accessed: 2023-08-03.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059. PMLR.

Hendrycks, D.; Burns, C.; Chen, A.; and Ball, S. 2021. CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review. *arXiv preprint arXiv:2103.06268*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Hüllermeier, E.; and Waegeman, W. 2021. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110: 457–506.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; and Brown, D. 2019. Text classification algorithms: A survey. *Information*, 10(4): 150.

Lai, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Ma, M.; Bartocci, E.; Lifland, E.; Stankovic, J. A.; and Feng, L. 2021a. A novel spatial–temporal specification-based monitoring system for smart cities. *IEEE Internet of Things Journal*, 8(15): 11793–11806.

Ma, M.; Preum, S. M.; Ahmed, M. Y.; Tärneberg, W.; Hendawi, A.; and Stankovic, J. A. 2019. Data sets, modeling, and decision making in smart cities: A survey. *ACM Transactions on Cyber-Physical Systems*, 4(2): 1–28.

Ma, M.; Stankovic, J.; Bartocci, E.; and Feng, L. 2021b. Predictive monitoring with logic-calibrated uncertainty for cyber-physical systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 20(5s): 1–25.

Ma, M.; Stankovic, J. A.; and Feng, L. 2018. Cityresolver: a decision support system for conflict resolution in smart cities. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, 55–64. IEEE.

Ma, M.; Stankovic, J. A.; and Feng, L. 2021. Toward formal methods for smart cities. *Computer*, 54(9): 39–48.

Manoj, B. S.; and Baker, A. H. 2007. Communication challenges in emergency response. *Communications of the ACM*, 50(3): 51–53.

Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; and Gao, J. 2021. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3): 1–40.

Mirończuk, M. M.; and Protasiewicz, J. 2018. A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, 106: 36–54.

Mukhopadhyay, A.; Pettet, G.; Vazirizade, S. M.; Lu, D.; Jaimes, A.; El Said, S.; Baroud, H.; Vorobeychik, Y.; Kochenderfer, M.; and Dubey, A. 2022. A review of incident prediction, resource allocation, and dispatch models for emergency management. *Accident Analysis & Prevention*, 165: 106501.

NYC911. 2022. Next-Gen 911 on Target for 2024 Completion. https://www.nyc.gov/content/oti/pages/press-releases/next-gen-911-on-target-2024-completion. Accessed: 08-15-2023.

OpenAI. 2022. Whisper: Speech Recognition. Accessed: 2023-08-03.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Poggi, M.; and Mattoccia, S. 2016. Learning from scratch a confidence measure. In *Bmvc*, volume 2, 4.

Poggi, M.; Tosi, F.; and Mattoccia, S. 2017. Quantitative evaluation of confidence measures in a machine learning world. In *Proceedings of the IEEE International Conference on Computer Vision*, 5228–5237.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Saxon, N.; Villena, P.; Wilburn, S.; Andersen, S.; Maloney, D.; and Jacobson, R. 2022. Annual Survey of Public Employment & Payroll Summary Report: 2021. *US Census Bureau*.

Sun, W.; Bocchini, P.; and Davison, B. D. 2020. Applications of artificial intelligence for disaster management. *Natural Hazards*, 103(3): 2631–2689.

suno-ai. 2023. Text-Prompted Generative Audio Model. https://github.com/suno-ai/bark. Accessed: 9 August 2023.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wex, F.; Schryen, G.; Feuerriegel, S.; and Neumann, D. 2014. Emergency response in natural disaster management: Allocation and scheduling of rescue units. *European Journal of Operational Research*, 235(3): 697–708.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; and Brew, J. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. https://github.com/huggingface/transformers. Accessed: 2023-02-20.

Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33: 17283–17297.