A hybrid approach to scalable and robust spoken language understanding in enterprise virtual agents

Ryan Price, Mahnoosh Mehrabani, Narendra Gupta, Yeon-Jun Kim
Shahab Jalalvand, Minhua Chen, Yanjie Zhao, Srinivas Bangalore
Interactions, LLC

Abstract

Spoken language understanding (SLU) extracts the intended meaning from a user utterance and is a critical component of conversational virtual agents. In enterprise virtual agents (EVAs), language understanding is substantially challenging. First, the users are infrequent callers who are unfamiliar with the expectations of a predesigned conversation flow. Second, the users are paying customers of an enterprise who demand a reliable, consistent and efficient user experience when resolving their issues. In this work, we describe a general and robust framework for intent and entity extraction utilizing a hybrid of statistical and rule-based approaches. Our framework includes confidence modeling that incorporates information from all components in the SLU pipeline, a critical addition for EVAs to ensure accuracy. Our focus is on creating accurate and scalable SLU that can be deployed rapidly for a large class of EVA applications with little need for human intervention.

1 Introduction

Advances in speech recognition in recent years have enabled a variety of virtual agents that answer questions, execute commands and engage in task-oriented dialogs in customer care applications. Beyond the accurate transcription of the user's speech, these virtual agents critically rely on interpreting the user's utterance accurately. Interpretation of a user's utterance – spoken language understanding (SLU) is broadly characterized as extracting intents – expressions that refer to actions, and entities – expressions that refer to objects. The entity expressions are further grounded to specific objects in the domain of the dialog (eg. `latest iphone` → `iphone 11`) or through world knowledge (eg. `Christmas` → `12/25`).

SLU has been a topic of research for the past three decades. Public data sets like ATIS (Price, 1990), SNIPS (Coucke et al., 2018), and recently FSC (Lugosch et al., 2019) have allowed for comparing various methodologies, including many recent developments driven by deep learning (Mesnil et al., 2014; Xu and Sarikaya, 2013; Liu and Lane, 2016; Price, 2020; Tomashenko et al., 2019). Such data sets are also a reasonable proxy for the intent classification and entity extraction handled by many consumer virtual agents (CVAs), applications that provide single shot question-answering and command-control services through smart-speakers or smart-home appliances. However, in contrast to the CVAs and the aforementioned data sets, enterprise virtual agents (EVAs) provide customer care services that rely on SLU in a dialog context to extract a diverse range of intents and entities that are specific to that business. SLU for EVAs encompasses a wide-ranging set of challenges. Speech recognition needs to be robust to varying microphone characteristics, diverse background noises, and accents. For EVAs, the robustness is further underscored as they are expected to deliver a better user experience to paying customers. Furthermore, SLU in EVAs needs to extract entities and intents that are specific to the domain of the enterprise. Matching expectations of novice users with the capabilities of SLU systems is challenging (Glass, 1999). Unlike users of CVAs, the users of EVAs are typically non-repeat users, who are not familiar with a particular EVA's conversational flow, leading them to provide unexpected and uncooperative responses to system prompts. Accordingly, EVAs need to contend with a larger space of alternative intents in a given dialog state. Other factors, like changes to the system that are dictated by business needs and continuous development of applications for new customers for which there is no labeled data yet, create a strong need for an SLU framework that can scale. Finally, while deep learning models with large modeling capacity can offer excellent results, latency at runtime is of great concern in paid for services like EVAs so designing towards lower computational complexity may be necessary (Tyagi et al., 2020).

To address the several challenges that relate to SLU in EVAs, we describe a general and robust framework for intent and entity extraction. Our primary goal is to create accurate and scalable SLU that can be widely deployed for a large class of EVA applications with little need for human intervention. We focus on techniques for the extraction and grounding of general entities (eg. dates, names, digit sequences) that are broadly used in SLU for EVAs, and also address the critical need for the extracted entities and intents to be associated with confidence scores that could be used by the dialog manager to

Proceedings of NAACL HLT 2021: IndustryTrack Papers, pages 63–71
June 6–11, 2021.

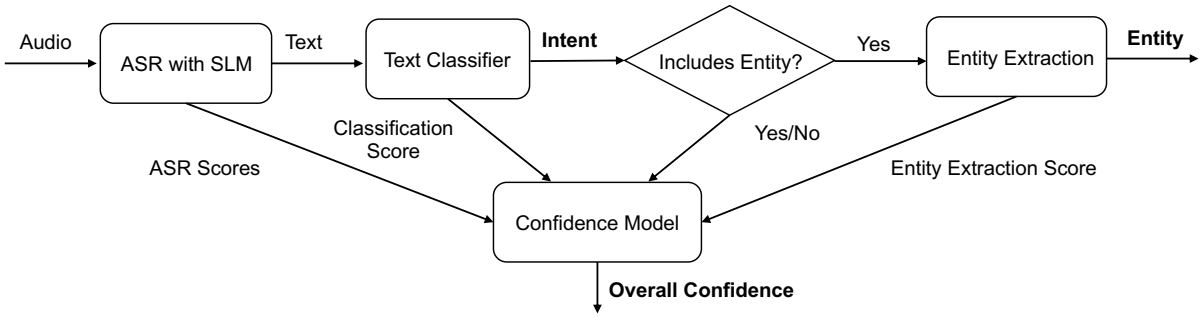

Figure 1: Flow diagram of the proposed pipeline. The outputs of interest for our human-in-the-loop SLU system are intents, entities, and overall confidence score.

either reprompt or to request human assistance. A variety of design considerations are discussed with insights drawn from real world EVA applications. We know of few previous studies having similar aim and scope of work as ours. Early work on industrial SLU systems sharing the aim of scalable SLU without human intervention was described in (Gupta et al., 2005), though without confidence modeling. An SLU pipeline is also addressed in (Coucke et al., 2018), but with design considerations made for CVA-like applications running on a device. While Gupta *et al.* (Gupta et al., 2019) does recognize that the needs of EVAs are different, their work primarily focuses on a framework for joint intent classification and slot filling that is modularized into different components.

This paper presents a complete study of a deployed SLU pipeline for handling intents and entities. The models described have been deployed in applications for Fortune 500 companies and a variety of design considerations are discussed with insights drawn from these real world EVA applications. In particular, we focus on improving performance on entities and intents for several core subtasks in a goal directed conversational system, namely date capture, number capture and name capture. Our contributions in this paper include (a) a unified framework for intent and entity identification (b) a synergistic combination of the robustness of statistical entity extraction models with rule-based value grounding (c) uncertainty modeling through confidence scoring and rejection criteria to maximize user experience (d) application of the framework for intent and entity extraction to new applications without the need for annotated data.

The outline of the paper is as follows. Section 2 provides an overview of the SLU framework for intent classification and entity extraction. Our experiments are presented in Sections 3, 4, and 5. Finally, conclusions and future work are given in Section 6.

2 Framework for Intent and Entity Extraction

In this section we describe the framework for simultaneous intent and entity extraction with confidence modeling. An illustration of the overall pipeline is show in Figure 1. We introduce the main components consisting of ASR, Text Classification, Entity Extraction, and Confidence Modeling depicted in Figure 1 in Sections 2.1, 2.2, 2.3, and 2.4, respectively. More details on the specific manifestations these components take on for a given task are described in Sections 3, 4, and 5.

2.1 ASR

The ASR systems used in our experiments consist of hybrid DNN acoustic models trained to predict tied context-dependent triphone HMM states with cross-entropy and sequential loss functions using 81-dimensional log-spectrum features. The pronunciation dictionaries consist of hand-crafted pronunciations for common words and grapheme-to-phoneme generated pronunciations for the rest.

Grammar-based language models (GLMs) can be very accurate in scenarios where the domain is constrained and the structure of likely utterances is predictable. Furthermore, GLMs have the advantage of not requiring much training data and provide recognition and semantic interpretation together, eliminating the need for an intent classifier and entity extractor. While there can be some overlap in GLMs used across similar dialog states making them attractive for immediate deployment, to really achieve peak accuracy in a non-trivial dialog state requires manual tuning by an expert, which is an obstacle to deploying GLMs rapidly at scale. Although it may seem that entity capture states in a well-designed dialog would elicit predictable user responses making them suitable for recognition with GLMs, in our goal-oriented dialogs deployed in EVAs we have observed that is not always the case. Statistical language models (SLMs) paired with intent classifiers and entity extraction methods can outperform GLMs. Therefore, we use SLMs built from n-grams or a hybrid LM combining SLMs and GLMs.

2.2 Intent Classification

We employ a linear Support Vector Machine (SVM) for intent classification, using n-gram based TF-IDF features. Although classifiers based on deep neural networks have gained popularity in recent years (Kim, 2014), linear classifiers remain as strong baselines (Wang and Manning, 2012), particularly on short text, with their ability to efficiently handle high-

dimensional sparse features, and their training stability through convex optimization.

In SLU, the outputs from ASR are inherently uncertain and erroneous. For example, an utterance corresponding to "I want to buy a phone" may result in multiple recognition hypotheses: ("I want to buy phone", "Want to buy phone", "I want a phone"), which we call ASR n-best. Instead of relying only on the first best ASR hypothesis, for intent extraction we use ASR n-best for better robustness and accuracy. There is a long history of leveraging information beyond the ASR 1-best for SLU in the literature (Hakkani-Tür et al., 2006; Li et al., 2020; Henderson et al., 2012).

To incorporate the ASR n-best information we take a sample-based approach. In this approach, we treat the hypotheses of an utterance as independent samples (with equal sample weights that sum to one), hence the number of samples will be larger than the number of original utterances. We apply this sample-augmentation process in the training phase, to account for the uncertainties in the ASR hypotheses. While in the testing phase, we first obtain the model scores for those independent samples, and then aggregate scores from the same utterance to yield the final scores for decision making. We use equal sample weights for hypotheses in the n-best because the weighting schemes we have tried based on ASR confidence for the entries in the n-best was not found to improve classification accuracy. Additionally, we found that an n-best list of three was sufficient for the tasks studied in this paper and increasing the number further just adds additional training time. The number of intents modeled by the text classifiers for the date, number, and name capture tasks we study ranges from approximately 20 to 40 different intents.

2.3 Entity Extraction

While increasingly accurate sequence tagging models for named entity recognition (NER) have been developed over the years, NER on speech input adds another complexity which cannot be mitigated by advanced algorithms developed for text alone. For speech input, recognition errors have to be accounted for at least in the form of a confidence value on the extracted values. EVAs must handle different types of spoken entities. Some appear with minor variations in surface forms (lexical strings) and appear in contexts where they are mostly unambiguous. For example, account numbers and phone numbers appear in the form of digit sequences of a predetermined length. Although ASR errors present some difficulties, such entities can be directly captured by a rule-based system and require little or no normalization. On the other hand, entities such as dates can appear in many surface forms like "this Monday", "New Year's day", "on the 7th", for example, and their context can cause ambiguities which require sequence tagging algorithms. In addition to sequence tagging, normalization is needed to convert the entity to the desired format. In any case, additional confidence models to account for ASR errors are required.

We also address entity capture tasks such as last name capture, that provide unique challenges in the context of speech input, but also have structure that can be leveraged to improve capture accuracy. EVAs for customer care dialogs must contend with a large number of unique names. Furthermore, many names may occur rarely and have unreliable pronunciations in the ASR lexicon. As a result, the main challenge is accurately recognizing the spoken name, rather than tagging and normalization. To accurately capture last names we leverage the spelling of the last name and utilize a hierarchical language model which combines SLMs and grammars.

2.4 Confidence Modeling

In order to maintain the high standard of customer experience demanded of EVAs, our SLU system utilizes a human-in-the-loop approach to ensure a sufficiently low error rate of the SLU system. Only high-confidence results from the SLU system are accepted, and utterances with low SLU confidence are handed-off to human agents who label them in real-time instead of being automated using the SLU output. The rejection of an SLU output is based on comparing the overall confidence measure for each utterance to a threshold. This utterance-level semantic confidence score quantifies the reliability of the information extracted from a spoken utterance, including entities and intents. It has been shown that combining speech recognition scores with semantic features to train a confidence model is an effective approach for semantic confidence estimation (Sarikaya et al., 2005; Mehrabani et al., 2018; San-Segundo et al., 2001). We use a logistic regression confidence model that is trained by passing each utterance through the SLU pipeline and the predicted result (intents and entities) is compared with the reference label containing the spoken intents and entities. After this binary model is trained, the following is used as the confidence measure:

$$p(\hat{y} = y|\vec{x}) = \frac{1}{1 + exp\big(-\sum_j \lambda_j x_j\big)} \quad (1)$$

where $\vec{x}$ is the confidence predictor feature vector, $\hat{y}$ is the predicted label (including all entities and intents) and y is the reference label. Confidence predictors x_j depend on the inputs and outputs of the SLU system and the feature weights that are estimated during confidence model training are denoted by λ_j.

We used a number of ASR confidence scores, based on posterior probabilities, as well as comparing the ASR best path to alternative paths (Williams and Balakrishnan, 2009). Basic statistics of word-level scores were computed to create utterance-level features. The number of ASR n-best was used as another feature as an indication of ASR uncertainty (larger number of n-best shows uncertainty). We also used the text classification scores as semantic features. Another semantic feature that we used was the predicted intent category encoded as a 1-hot vector over the intent classes. ASR confi-

dence for digits or the number of digits in the ASR n-best text were also added as features. Finally, since for number and date capture dialog states we utilized a text classifier that in addition to intent, showed if the utterance included the relevant entity or not, we used this as a binary feature which was an effective indicator of semantic confidence.

3 Date Capture

In this section, we apply the described framework to the task of date capture and we also describe our approach to creating a generic date capture model in Section 3.1. Typically, dialog-state specific models are built using labeled data from a single dialog state to train an intent classifier and entity extraction pipeline for the target state. However, the generic date capture model enables rapid deployment of models for date capture states in new applications before any data can be collected.

At least four different components are essential for capturing dates in speech input. 1) A language model for ASR to reliably transcribe the input speech. 2) A sequence tagger for identifying the span of transcribed speech containing the date specifications. 3) A function that takes into account chances of errors and computes a confidence value in the extracted entity. Finally, 4) a normalizer that converts the identified span into the desired date format. In a fully rule-based approach, the grammar-based LM performs the functions of all four components. For ASR, we use an SLM trained on a large corpus of utterances containing dates as well as utterances containing different intents instead of date entities. For span identification we use a statistical sequence tagger (MEMM (McCallum et al., 2000) or BLSTM-CRF (Huang et al., 2015)) trained on date tagged data. For entity extraction confidence, we use logistic regression models trained with scores from the tagger and from text-based binary `Date` or `No-Date` classifiers. For normalization, we use a rule-based approach applying a grammar to the tagged sequence of text.

While a large majority of users do provide a response with a date to a system prompt requesting a date, a significant number of users do not, and instead respond with utterances expressing different intents that must be robustly identified for the dialog to progress gracefully. We trained a text classifier as described in Section 2.2, which in addition to many non-date related intents such as `Cancel Reservation`, `Billing and Charges`, and `Live Agent`, includes a `Date` label, as well as `Vague Date`, for when the user responds with a partial date, such as only the month, rather than an utterance with a date expression that could be grounded to a specific date. A `Vague Date` intent can be used to trigger a reprompting of the user to disambiguate. In the case that the sequence tagger detects a date but the intent classifier does not return a `Date` intent, the detected date entity is still returned by the system. Including the `DATE` intent, there are a total of 41 intents in this date capture task.

The training, development and test sets consist of approximately 53K, 5K, and 10K utterances labeled by humans-in-the-loop, respectively. We compare the proposed framework with an SLM and a MEMM sequence tagger against a grammar-based LM that has been hand-tuned for accuracy on the target dialog state. Confidence-based rejection is typically employed to ensure a sufficiently low error rate of EVAs at run-time. Therefore, it is more informative to analyze the performance of SLU systems by examining the error rate as a function of the utterance rejection rate at different thresholds, rather than just reporting the average error rate at 100% automation. In this way, a suitable operating point at a low error rate can be selected to evaluate the performance of an SLU system.

We plotted the error rate of accepted utterances versus the percentage of utterances rejected using a confidence-based threshold (FA-Rej curve) for each system in Figure 2. Both intent classification and entity extraction performance are reflected in these plots because both the intent and entity, if present, must be correct. We observe superior performance with the proposed approach, noting that the proposed approach starts with a slightly lower error rate but due to the effectiveness of the designed confidence modeling, the gap in performance between the two approaches grows considerably wider as low-confidence utterances are rejected. At an operating point of 5% error, the proposed approach offers about 10% more automation compared to the grammar-based approach, a significant gain.

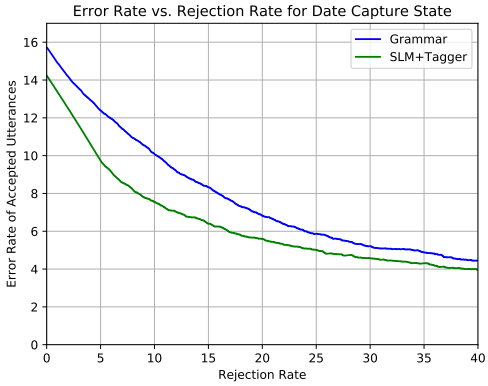

Figure 2: The error rate of accepted utterances versus the percentage of utterances rejected using a confidence-based threshold (FA-Rej curve) for a hand-tuned grammar-based LM compared to the proposed framework for a date capture state in a car rental dialog.

3.1 Generic Date Capture Model

Building out models for new dialog states and applications at scale is challenging under the paradigm of collecting data for training dialog-state specific intent classifiers and entity taggers. To address this issue, we propose a modeling approach that enables deployment of models for new capture states on day zero. First, a representative set of dialog states for a given entity, such as date, are identified and data from those states

is aggregated. For example, to build a generic date capture model date capture states pertaining to service start or stop dates, hotel check-in dates, car rental pick-up dates, service appointment dates, and so on are pooled together. Then either rule-based or statistical models for entity extraction are trained using the combined data. There can be a "long tail" of unique dialog state specific intents that may appear in one dialog state from one application but would result in an invalid output that can not be handled by the dialog manager in the dialog state of another application. Thus, a set of fairly "universal" intents for this collection of dialog states must be found. The generic model can then be applied to a new target domain or task that is semantically similar without additional training data. However, the generic model does not generalize to new entity types, meaning that a generic date capture model would be applied to new date capture states only.

The training data for the generic date capture model is aggregated across six date capture states from five different EVA applications. Approximately 1.1 million utterances were used for training the intent classifier and entity extraction pipeline for the generic date capture model. Testing is done on approximately 10K utterances from a held-out date capture state from a novel application whose data never appeared in the training set. The generic intent classifier model supports 38 different intent classes that were determined based on the intents observed in the cross-application training data. The test data from the held-out dialog state contains unique intents that are not covered by the intent classifier because they did not occur in the other states comprising the training data. We compare the generic date capture model having a MEMM sequence tagger to a dialog state specific model having an intent classifier and a BLSTM-CRF sequence tagger trained on 62K utterances from the target dialog state. We use a BLSTM-CRF for the model trained on target dialog state data because it improved performance slightly but we use a MEMM in the case of the generic model because the BLSTM-CRF did not improve performance on that data.

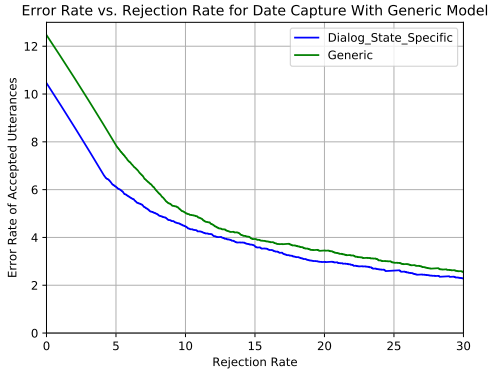

Figure 3: FA-Rej curves for a generic model and a dialog state specific model in a utility start of service date capture state.

FA-Rej curves for both the generic and dialog state specific SLU systems are shown in Figure 3. As expected, there is some loss in performance relative to the dialog state specific model trained on data from the target dialog state. However, analysis of the errors reveals that the performance on entity extraction is unchanged and the loss is largely due to a few specific intents that were not covered in the generic model in this case. Furthermore, the generic model results in a loss of only about 2.5% in automation at an operating point of 5% error, which we believe is reasonable given that this model can be deployed immediately once a new application goes online since data from the target dialog state or application is not required for training.

4 Number Capture

The goal in number capture dialog states is to capture a long sequence of digits, such as phone or account numbers. While the majority of users provide the numeric input as requested by the system prompt, approximately 30% of utterances do not include a digit sequence. Therefore the challenge in such dialog states is two-fold: 1) ensuring that if the user provided a digit sequence, it is captured accurately – a challenge due to ASR errors (even if one digit is substituted or deleted, the entire digit sequence is inaccurate) 2) if the user responds with a non-digit utterance, capture the provided intents in the utterance.

Traditional SLU systems use ASR with a carefully hand-tuned grammar-based LM to capture the digit sequence but a separate grammar needs to be designed and tuned for every new application to cater to that application's intents so it is difficult to scale. In contrast, we demonstrate in Section 4.1 that our proposed pipeline for generic digit sequence models, once trained, can be applied to any utterance with digit sequences. As an alternative to hand-tuned grammar-based models, DNN-based slot-filling models could be applied but they typically require large amounts of domain-specific annotated data for training.

We propose a hybrid grammar-based and statistical approach that overcomes the limitations of grammar-based models alone, yet is scalable and maintains high accuracy. Following the framework described in Section 2, we use an SLM-based ASR system and train a text classifier on the output for intent detection. A `Number` label is used for all utterances that only include a digit sequence, along with a broad set of other intent labels to cover the approximately 30% of utterances that do not include digit sequences. If an utterance is classified as including a digit sequence via the `Number` label, a rule-based system is used to extract and normalize the number. Note that this approach yields the best accuracy for utterances in a specific dialog state since the structure of the digit sequence is predetermined, but for more general number capture an entity tagger could be applied. The rule-based system finds the best digit sequence match in any of the ASR n-best results. Addi-

tionally, we trained a confidence model to produce an overall confidence score. An important factor in confidence estimation for number capture is the presence or absence of the digit sequence, and therefore we use that as an additional binary confidence predictor feature. Furthermore, if a digit sequence is detected in the utterance, ASR word scores for the recognized digits, and the length of the digit sequence are used as input features for the number capture confidence model.

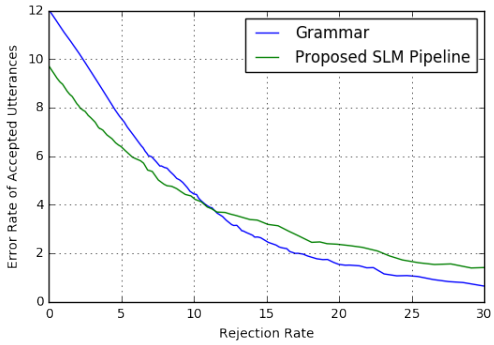

Figure 4: FA-Rej curves for a hand-tuned grammar-based LM compared to the proposed framework for a phone/account number capture state.

We compare the proposed pipeline to a hand-tuned application-specific grammar-based LM approach for account/phone number capture. For this experiment, 2K utterances with reference labels were used for testing, and about 3M utterances for training. Note that only a small subset of the training data (∼10%) which had low SLU confidence with an existing grammar-based system were labeled by humans in an online fashion. The data to train the confidence model included about 300K utterances with online human labels. Results are shown in Figure 4. The accuracy (at zero rejection) with the proposed approach has improved by 2.35% absolute, and at an operating point of 5% error, the proposed approach offers 1.2% more automation compared to the grammar-based approach. As shown the grammar-based approach outperforms the SLM-based pipeline for error rates of lower than 4%, which is due to several rounds of careful hand-tuning of the grammar-based LM for some of the less frequent utterances. However, the proposed approach is still superior because of its flexibility to be easily applied to any application.

4.1 Generic Number Capture Model

Following a methodology similar to the one described in Section 3.1, a generic model for digit sequence capture was built. Data for the generic number capture model was pooled from five different applications containing digit capture states with digit sequence lengths ranging from 5-10 digits. In total, 715K utterances were used for training an intent classifier that covered 69 unique intents for these digit capture states, including a label to indicate the presence of a digit sequence. Approximately 67% of the training utterances contained digit sequences and the remaining 33% were only other intents. As before, a rule-based system is used to extract and normalize the number when the intent classifier predicts a digit sequence is present. To train a system-level confidence model, a total of 88k held-out utterances having human-in-the-loop annotated labels from the set of five applications was used. The generic intent and confidence models for digit capture were tested on a test set from one of the five applications included in the model using held-out data and compared to a dialog state specific model trained with data from the target application.

Similar to the results for the generic date capture model in Section 3.1, we observe that the generic model for number capture does perform slightly worse than the dialog state specific model but still offers an acceptable level of automation at an operating point of 5% error. The number capture accuracy of the generic digit capture model is approximately 1% lower than that of dialog state specific model at zero rejection, and less than 2% performance difference at other rejection rates. Error rate versus rejection rate curves for the two models are shown in Figure 5.

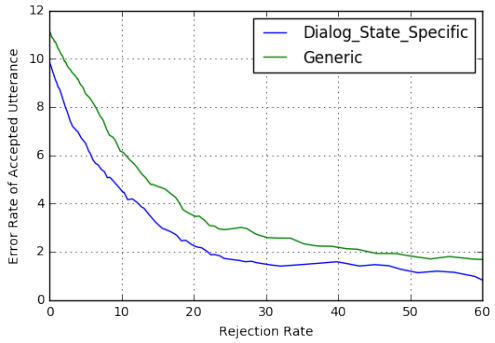

Figure 5: FA-Rej curves comparing a generic model and a dialog state specific model for a number capture dialog state.

5 Name Capture

Person name recognition is a difficult task in spoken language understanding due to the size of the vocabulary and confusions in name pronunciations (Yu et al., 2003; Raghavan and Allan, 2005; Bruguier et al., 2016). In the course of customer care dialogs users are often asked to provide their last name for identification purposes. There are a very large number of last names, some of which are similar sounding like “Stuard”, “Stuart”, and “Stewart”, making it difficult to accurately recognize names in isolation. However, if the user is also asked to provide the spelling as well that can be leveraged to correctly capture the name. We observe that names at the beginning of an utterance are very difficult for ASR to recognise correctly but spelled letters are often recognized more accurately and can be concatenated to capture the name. To recognize potentially hundreds of thousands of last names using a traditional n-gram SLM or grammar, every possible last name and spelling sequence should be encoded, resulting in a very large LM. Instead, we propose a hierarchical language model,

which consists of sub language models derived from the beginning sounds of the last names (hereafter, we call this language model *1-layer LM*). This is motivated by the fact that the beginning of a name's pronunciation leads the rest of name and spelling sequence, unlike other ASR tasks (see Figure 6a).

Still, asking the user to also spell their name does not make the recognition task trivial. When spelling a word, there are frequent confusion pairs such as 'f and s', 'b and v', 'p and t' and 'm and n'. To distinguish between such confusion pairs, a common practice is to use the NATO phonetic alphabet - "Sam S as in sierra A as in alpha M as in Mike". However, people tend to use any word they can think of easily for distinguishing the characters in their name, rather than adhering to the NATO phonetic alphabet which may not be familiar to many users. Thus, we added another layer of sub grammar at the bottom of last name sub grammars in the hierarchical language model to cover the NATO phonetic alphabet, as well as a large number of other words people use to distinguish characters (hereafter, *2-layer LM*) shown in Figure 6b. Similar to the date and number capture systems, our approach for last name capture also incorporates an intent classifier covering a set of intents which are likely to occur when last names are not given.

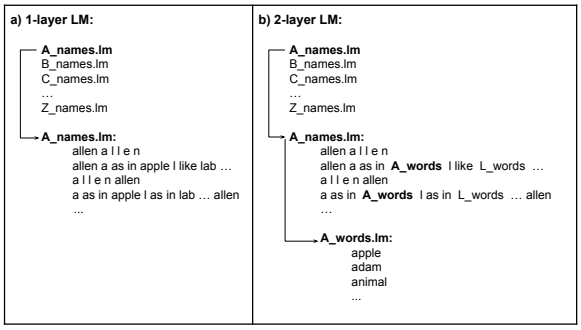

Figure 6: Last name LMs: a) 1-layer LM is trained on name and spell as it is; b) 2-layer LM is trained on names and spells but taking NATO words as another LM component.

We compare four different systems to capture last names in spoken input: 1) *SVM* classifier trained on 170K ASR hypotheses using bi-gram features and human annotated labels for the names; 2) *1-layer LM* with which we decode the utterances and then concatenate the spelled letters to predict the last name. Note that the usual ASR confidence score is used as prediction confidence to draw the rejection curves; 3) *2-layer LM* which is used in the same way as the second system and 4) *2-layer LM with confidence model* which is used in the same way as the third system, but instead a confidence model (described in Section 2.4) is exploited to generate the confidence scores. The confidence model is trained on a 29K data set with features consisting of the ASR-based confidence scores and utterance length.

A test set containing 1K utterances labeled with the last name by human annotators is used for testing. Curves for the various systems on the test set are shown in Figure 7. As expected, the SVM classifier performs very poorly due to the problem of data sparsity in the data set. We selected this approach as one of our baselines for comparison because it shows reasonable performance on a first name capture task where the sparsity of data is less than it is for last names. The second algorithm in which we use the *1-layer LM* to decode the utterances and then concatenate the spelled letters to determine the last names performs better on average but it fails in many cases due to the inclusion of characters that distinguish words in the utterance. However, the *2-layer LM* resolves many of those issues and it significantly improves the accuracy, requiring far fewer utterances to be rejected at an operating point of 5% error. Confidence modeling only marginally helps performance with the simple ASR confidence features used and we suspect more informative features need to be designed.

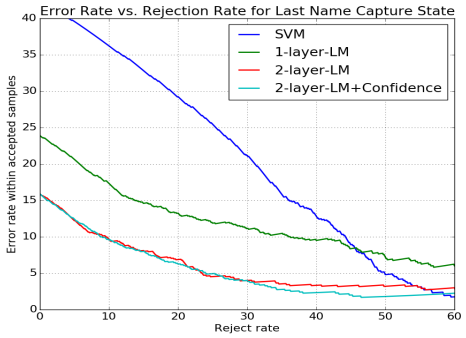

Figure 7: FA-Rej curves for last name capture.

6 Conclusions

SLU for EVAs encompasses a wide-ranging set of practical challenges and investigations into the design of accurate and scalable SLU systems that can quickly be deployed for new applications without requiring much human intervention each time is warranted. In this paper, we have presented an enterprise-grade deployed SLU pipeline for handling intents and entities and demonstrated its effectiveness across several real world subtasks in a deployed customer care virtual agent. We have also highlighted the importance of confidence modeling using features from each component in the pipeline. The proposed approach to create generic date and digit capture models for intents and entities allows for day zero deployment of models for new applications. In the future, we will incorporate word confusion networks and lattices for the different capture tasks presented in this paper.

References

Tony Bruguier, Fuchun Peng, and Françoise Beaufays. 2016. Learning personalized pronunciations for contact names recognition. In *Interspeech*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

James Glass. 1999. Challenges for spoken dialogue systems. In *Proceedings of the 1999 IEEE ASRU Workshop*, volume 696.

Arshit Gupta, AI Amazon, John Hewitt, and Katrin Kirchhoff. 2019. Simple, fast, accurate intent classification and slot labeling for goal-oriented dialogue systems. In *20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 46.

Narendra Gupta, Gokhan Tur, Dilek Hakkani-Tur, Srinivas Bangalore, Giuseppe Riccardi, and Mazin Gilbert. 2005. The at&t spoken language understanding system. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):213–222.

Dilek Hakkani-Tür, Frédéric Béchet, Giuseppe Riccardi, and Gokhan Tur. 2006. Beyond asr 1-best: Using word confusion networks in spoken language understanding. *Computer Speech & Language*, 20(4):495–514.

Matthew Henderson, Milica Gašić, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young. 2012. Discriminative spoken language understanding using word confusion networks. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 176–181. IEEE.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Mingda Li, Weitong Ruan, Xinyue Liu, Luca Soldaini, Wael Hamza, and Chengwei Su. 2020. Improving spoken language understanding by exploiting asr n-best hypotheses. *arXiv preprint arXiv:2001.05284*.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Interspeech*, pages 685–689.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech Model Pre-Training for End-to-End Spoken Language Understanding. In *Proc. Interspeech 2019*, pages 814–818.

Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Icml*, volume 17, pages 591–598.

Mahnoosh Mehrabani, David Thomson, and Benjamin Stern. 2018. Practical application of domain dependent confidence measurement for spoken language understanding systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 185–192.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2014. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Patti Price. 1990. Evaluation of spoken language systems: The atis domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Ryan Price. 2020. End-to-end spoken language understanding without matched language speech model pretraining data. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7979–7983. IEEE.

Hema Raghavan and James Allan. 2005. Matching inconsistently spelled names in automatic speech recognizer output for information retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 451–458, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Rubén San-Segundo, Bryan Pellom, Kadri Hacioglu, Wayne Ward, and José M Pardo. 2001. Confidence measures for spoken dialogue systems. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 1, pages 393–396. IEEE.

Ruhi Sarikaya, Yuqing Gao, Michael Picheny, and Hakan Erdogan. 2005. Semantic confidence measurement for spoken dialog systems. *IEEE Transactions on Speech and Audio Processing*, 13(4):534–545.

Natalia Tomashenko, Antoine Caubrière, and Yannick Estève. 2019. Investigating adaptation and transfer learning for end-to-end spoken language understanding from speech. In *Interspeech 2019*, pages 824–828. ISCA.

Akshit Tyagi, Varun Sharma, Rahul Gupta, Lynn Samson, Nan Zhuang, Zihang Wang, and Bill Campbell. 2020. Fast intent classification for spoken language understanding systems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8119–8123. IEEE.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea. Association for Computational Linguistics.

Jason D Williams and Suhrid Balakrishnan. 2009. Estimating probability of correctness for asr n-best lists. In *Proceedings of the SIGDIAL 2009 Conference*, pages 132–135.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *ieee workshop on automatic speech recognition and understanding*, pages 78–83.

Dong Yu, Kuansan Wang, Milind Mahajan, Peter Mau, and Alex Acero. 2003. Improved name recognition with user modeling. In *Eighth European Conference on Speech Communication and Technology*.