Entity resolution for noisy ASR transcripts

Arushi Raghuvanshi Varsha Embar[‡] **Vijay Ramakrishnan**[‡]
Lucien Carroll Karthik Raghunathan
Cisco Systems
[‡]authors contributed equally

Abstract

Large vocabulary domain-agnostic Automatic Speech Recognition (ASR) systems often mistranscribe domain-specific words and phrases. Since these generic ASR systems are the first component of most voice assistants in production, building Natural Language Understanding (NLU) systems that are robust to these errors can be a challenging task. In this paper, we focus on handling ASR errors in named entities, specifically person names, for a voice-based collaboration assistant. We demonstrate an effective method for resolving person names that are mistranscribed by black-box ASR systems, using character and phoneme-based information retrieval techniques and contextual information, which improves accuracy by 40.8% on our production system. We provide a live interactive demo to further illustrate the nuances of this problem and the effectiveness of our solution.[1]

1 Introduction

General purpose ASR has improved by a large margin in recent years, with a reported word error rate (WER) of less than 10% for English voice search queries (Chiu et al., 2018). However for domain-specific vocabularies, uncommon terms like proper nouns, non-native English accents, and noisy acoustic settings, the WER is still high. Since ASR is the first component in a spoken dialog system, errors introduced in the recognized transcript cascade to downstream natural language understanding (NLU) components, leading to unsatisfactory user experiences. While building a domain-specific ASR system could address this problem (Gao et al., 2001; Zhao et al., 2018), doing so requires prohibitively high amounts of data and resources. Therefore, using a generic black-box ASR system and handling mistranscriptions as a post processing step is a more practical approach for most industry applications.

One such application affected by cascading ASR errors is *entity resolution*. It involves **identifying** and **linking** different references for the same real world object to a canonical form. For task-oriented dialog systems, robust entity resolution is a challenging task because users generally refer to entities informally using abbreviations and aliases, rather than official standardized names, and for many applications, entities tend to be domain-specific, uncommon terms that are most often mistranscribed (Laurent et al., 2014).

Consider an office voice assistant that helps employees start a call with a colleague by name. The assistant needs to 1) identify the name in the user query and 2) resolve it to an employee within the company. For instance, in a query *Call John please* the extracted person name entity *John* could resolve to *John Scott Edwardson (ID: 576253)*.

While popular English names like *John* or *David* are recognized by generic ASR systems with high accuracy, less common or non-English names are often mistranscribed. For example, *Dial into **Dorlis's** meeting* may get transcribed as *Dial into **doorless is** meeting*, where the person name entity is incorrectly transcribed as common English words. Similarly, *Call **Nguyen*** may be transcribed as *Call **Newman***, where the person name entity is incorrectly transcribed as a different and more common name. In the first case, the assistant fails to identify an entity to place a call and in the second it has the wrong name, leading to a call to the wrong person. In either case, it appears unintelligent to the end user.

ASR vendors provide some room for domain personalization in the form of "hints", i.e. a list of expected phrases the ASR system can bias its hypotheses towards. The number of allowed hints is usually capped at a relatively small number (e.g.

[1] https://vimeo.com/345579360

Proceedings of the 2019 EMNLP and the 9th IJCNLP (System Demonstrations), pages 61–66
Hong Kong, China, November 3 – 7, 2019.

500) and does not scale to the full set of domain-specific vocabulary required (e.g. a few thousand employee names in a company).

In this paper, we propose a scalable, unsupervised solution for tackling the problem of entity resolution of named entities in noisy ASR transcripts for an enterprise collaboration assistant. We demonstrate improvements on the task by utilizing text and phoneme information retrieval techniques along with contextual and personalization information available in an enterprise setting. This approach can easily be extended to other domains,[2] as demonstrated in MindMeld, our open source conversational AI platform.

2 Related Work

Previous work on entity resolution for noisy text mostly deals with spelling errors (Bassil and Semaan, 2012), ambiguous terms, or noise induced through style of writing (like in social media platforms) (Campbell et al., 2016). The problem of noise induced through ASR errors is different in nature. Some systems use a wide range of features like lexical, syntactic, phonetic and semantic features to identify presence of ASR errors in transcripts, asking the user to clarify the intended meaning when an error is detected (Hazen et al., 2002; Prasad et al., 2012; Marin et al., 2015). To detect out of vocabulary (OOV) name errors, a multi-task recurrent neural network language model was used by (Cheng et al., 2015).

Using a ranking mechanism on the n-best hypotheses generated by one or more ASR modules is another popular approach used in dialog systems (Morbini et al., 2012). Re-ranking systems like (Corona et al., 2017) make use of a language model and semantic parsing features, but ignore any OOV words encountered, losing valuable information. While these approaches improve downstream tasks like entity recognition (Zhai et al., 2004; Hakkani-Tür et al., 2006), they cannot correct to words that do not exist in one of the hypothesized transcripts.

While these methods are not directly applicable to our problem, we extend some of the features and ideas discussed in these papers for entity resolution.

[2] https://www.mindmeld.com/docs/blueprints/overview.html

3 Approach

Given a noisy ASR transcript of a user query that potentially contains named entities, our goal is to identify the span of text that corresponds to an entity and resolve each identified span to a canonical value that can be looked up in a database.

Our dialog system consists of a set of classifiers, information retrieval components and a dialogue manager as described in Raghuvanshi et al. (2018). We use MindMeld[3] to build and train the intent classifier and entity recognizer with crowd-sourced data for all the intents handled by the assistant including "call by name" which includes "person name" entities. Since we operate on a narrow domain, the model learns to rely on the patterns of surrounding common English words, which are generally transcribed correctly, to detect entity spans which may be mistranscribed.

For entity resolution, we store the organization's employee database, metadata, and extracted features in an inverted index which we describe in 3.1. At inference time, for each detected person name entity span, we extract the same features from the span along with metadata and use information retrieval methods to retrieve a ranked list of the most likely matching canonical names.

3.1 Features

Our system utilizes four broad feature categories. We describe each below and provide the implementation.[4]

3.1.1 Textual Similarity

For text-based retrieval, we leverage normalization, character n-grams, word n-grams, and edge n-grams. Exact matching is essential for resolving names that are already correctly transcribed. Matching against normalized text accounts for capitalization variations and special characters (e.g. *Oleary* to *O'Leary*). Character n-grams account for spelling variations which are common in entities like person names (e.g. *Ashley* to *Ashlee*). Word or token n-grams are useful for partial name matching (e.g. *Carly Rae* to *Carly Rae Jepsen*). We observed that the phonemes at the edges of tokens tend to contribute more to our notion of phonetic similarity than some of the middle phonemes

[3] https://github.com/cisco/mindmeld

[4] https://github.com/cisco/mindmeld/blob/master/mindmeld/components/entity_resolver.py

(e.g. *Monica* seems more similar to *Malika* than *Sonic*). Using edge n-grams accounts for this.

In addition, the index contains domain-specific metadata of synonyms or in our case, common nicknames. For example *Sid* is populated as a common nickname for *Siddharth*, *Teddy* for *Theodore* and *Bob* for *Robert*. This information matches colloquial name utterances to the formal "given" and "family" names in the index.

3.1.2 Phonetic Similarity

In many cases, relying solely on text matching will return results that are phonetically different from the original utterance. For example, *Gaurav Sharma* is transcribed as *quarter shawarma*, which is textually more similar to *Carter Warmac* than the original name, but phonetically quite different. In order to correct uncommon names for which the mistranscriptions are often beyond simple text variations, phonetic features are essential.

As we are leveraging third party ASR systems via APIs, we do not have direct access to the phonetic information from the original audio. Instead, we use techniques to recover the phonetic representation of the transcribed text. We use double metaphone (Philips, 2000) as well as grapheme-to-phoneme (G2P) representations (Daelemans and van den Bosch, 1997) to generate our phonetic features. Double metaphone is an algorithm that maps tokens to approximate phonetic representations using rules and heuristics developed primarily for English names, but extended to Chinese, Romance, and Slavic languages. The G2P toolkit in CMU Sphinx[5] is a sequence-to-sequence deep learning model that maps text to a phonetic representation. Its coverage and accuracy is dependent on the training data, which consists of common English words as well as person names. We found that the two representations had complementary information, and we benefit from using both.

While this feature is essential, no phonetic encoding technique is perfect, and the same name may have different phonetic representations when spoken by people with different accents. Therefore, it is important to balance it with the other feature categories.

3.1.3 n-best Transcripts

Almost all off-the-shelf ASR systems return a ranked n-best list of multiple possible transcripts. The n-best entity spans, extracted from each of the alternate transcripts, provide additional phonetic information about the original audio. In some cases, the exact correct name may even exist in one of the lower ranked transcripts. The reliability of each hypothesis generally decreases as we go down the n-best list, so while all n-best spans contribute to selecting the final candidate, our weighting scheme ensures that matches against higher ranked alternates have a larger impact.

Consider an utterance *Helen* which was mistranscribed to *Ellen*, but in the n-best list (*Ellen, Hellen, Helena, Hella, Hello, Helen*), all of the other hypotheses start with an 'H', and the original utterance *Helen* exists as one of the lower ranked hypotheses. By utilizing the n-best list in conjunction with phonetic similarity features, our retrieval method has a better chance of correcting to *Helen*.

3.1.4 Personalization Features

The personalization features are highly domain and user specific, but have a high impact on the precision of our model. For the use case of calling a person, we capitalize on the observation that a user is more likely to call someone they often interact with or who is close to them in the organization hierarchy. The caller's identity can be determined by a variety of methods including authentication, device pairing, face recognition, or speaker identification. Based on information like the interaction frequency between employees, we generate a personalization factor from the user's identity to help match the entity span to the intended name.

The personalization features are generalizable across different domains. For instance, consider a food ordering use-case where the user Alice is trying to order a dish called "Dabo Kolo". Based on personalized knowledge that Alice regularly orders Ethopian cuisine in San Francisco, we can accordingly boost relevant dish search matches even if it was mistranscribed to "Debbie Carlo".

3.2 Hyperparameter Tuning

We used a combination of manual and random walk tuning to learn the optimum weights of the different features. Quantitative evaluation is based on whether the correct name exists in the top 1 or top 5 ranked results. For the random walk tuning, we define an objective function that optimizes the recall score over these features.

As a production application, we are concerned with not only recall, but also the relevance of the

[5] https://github.com/cmusphinx/g2p-seq2seq

other top results, and how egregious the errors are when the correct name is not found. For tuning on these qualitative factors, we rely on manual analysis.

4 Data and Experimental Setup

We test our technique on a crowd-sourced dataset of audio transcripts for the "call by name" intent. The name used in each transcript was sampled from ∼100 k employees in the directory of a global company with up to 10 speakers for each name. We used Google Speech-to-Text[6] to collect 10-best ASR transcripts for each sample. Table 1 gives the overall stats and name type distribution of the evaluation dataset.

Table 1: Summary of dataset used for evaluation

# Samples	# First Names	# Full Names
2915	1234	1681

For evaluating the personalization features, we augment the data with four different settings. For each name span in the dataset, we record the entity resolution accuracy for each of the following scenarios:

1. **Same team**: the caller directly works with the person they are trying to call
2. **Same department**: the caller is a few hops away from the person they are trying to call
3. **Different department**: the caller has no previous interaction and is far removed from the person they are trying to call
4. **No information**: the system is unable to identify the caller

This ensures that the system is not over-optimizing for close interaction distances and users are still able to call people who they have little or no previous interaction with. We report the recall averaged across these interaction distance settings.

Assuming that a person can only call someone within the company, we populate the index with all ∼100 k employees. For each entry, we have a unique identifier, the full name of the employee, common nicknames for the name, as well as the job title and location. We also have an interaction corpus which contains information on the previous interaction history of users and the organizational hierarchy which is used to generate a "personalization score" between any two users in the index.

We evaluate our system using the IR metric of recall at n (R@n). We report numbers for $n = 1$ and $n = 5$. R@1 evaluates the effectiveness of the entity resolver by measuring the quality of the top result, while R@5 gives the likelihood with which the user can find the correct entity at least within the top 5 suggestions provided by the system.

5 Results and Discussion

Our approach significantly improves the recall of recovering the correct name from a noisy ASR transcript. The final setup gives a 40.8% improvement of R@1 over the baseline (Table 2). Table 3 breaks down the recall of the system by the interaction distances. For the most common scenario—that of users calling those they often interact with—the improvement is even larger. Because these four categories of caller distance are equally weighted in the optimization process, increased recall in the cases of same team/department leads to lower recall in the different department cases, where personalization features are generally misleading.

Table 2: Evaluation of the entity resolver with addition of different features

Features	R@1	R@5
+ Textual Features	0.100	0.120
+ Phonetic Features	0.255	0.347
+ n-best List	0.326	0.454
+ Personalization	**0.508**	**0.627**

Table 3: Performance of the final system based on the distance between the caller and the callee

Caller Distance	R@1	R@5
Same team	**0.765**	**0.864**
Same department	0.659	0.777
No information	0.326	0.454
Different department	0.281	0.412

The name resolution improvement converts an unusable product to a reasonably intelligent one with significantly less cost and computation than alternate approaches of building a domain-specific ASR. The remaining names that are not correctly resolved in the top position, often appear in one of the following ranked positions that a user can scroll between, as illustrated by the R@5 metric, or the name can be resolved by a follow-up query.

[6] https://cloud.google.com/speech-to-text/

We also present the WER of the transcripts in Table 4. While the entity resolution recall metric is more relevant, the ASR metric of WER illustrates the magnitude of errors in the original ASR transcriptions and further reinforces the effectiveness of our system. We compare the full user query transcripts from the ASR system with the transcript where recognized name spans are replaced with the top ranked name string from our entity resolution model. We find that using our IR entity resolution approach we get a relative WER reduction of 29.0% on name tokens and 12.0% on the full query. Note that this additionally demonstrates that this approach can be extended beyond entity resolution to the task of ASR correction with compelling results, particularly for correcting mistranscriptions of domain-specific entities.

Table 4: WER comparison before and after ASR correction

Model	Name WER	Transcript WER
ASR transcripts	$86.0 \pm 1.6\%$	$40.8 \pm 0.6\%$
IR entity correction	$61.1 \pm 0.7\%$	$35.9 \pm 0.4\%$

5.1 Qualitative Analysis

We performed manual evaluation while tuning the system, and found several broad categories of ASR errors that appeared often. We analyze some of them and discuss features of our ranking approach that help correct for those errors. These examples can be visualized in our interactive UI.

- Language model of generic ASR systems incorrectly biases to common vocabulary.

  ```
  Gold: Prasanth Reddy
  ASR: croissant ready
  ```

  In these cases, n-best lists are often the most important feature, since the less common name tokens usually appear in one of the alternate transcripts.
- ASR model mistranscribes to similar phonetically but textually different tokens.

  ```
  Gold: Kiran Prakash's
  ASR: Corrine precautious
  ```

  In these cases, the phonetic features like double metaphone and G2P are important, as they allow us to match at a phonetic level.
- Entities are mistranscribed to other entities.

  ```
  Gold: Didi
  ASR: Stevie
  ```

  Here, contextual features are the key. Names are often mistranscribed to other more popular valid names. A single ASR transcript may be a correctly transcribed name or a mistranscription of another name, but personalization and the set of n-best transcripts provide evidence of the correct transcription. In this example, if a user had actually said *Stevie*, we shouldn't only return people whose name is *Didi*. However, if the intended person is *Didi*, which is consistently mistranscribed to *Stevie*, we need a way to recover that correct name. The personalization factor can boost names like *Didi* towards the top of the ranked list, instead of only returning names like *Stevie* or *Steven*. The n-best list can also help determine the confidence of the name. If all of the transcripts contain *Stevie*, then that is likely what the user actually said. But if the n-best results contain many terms which start with 'D', then it is more likely the user said something else and we can use this combined with other signals to recover the correct name.
- Some phonemes are not recognized due to noisy audio.

  ```
  Gold: Mahojwal
  ASR: my jaw
  ```

  There are many cases where some phonemes are dropped or added in the mistranscription. Again, tuning to account for this type of noise is a balancing act, since we don't want our system to hallucinate sounds that don't exist in the original utterance, but it needs to have enough leeway that it can recover the correct name from noisy transcripts where a phoneme may be dropped or added. Fuzzy matching of both characters and phonemes are useful for correcting these cases.

Another error category is when the context words i.e, words surrounding the name in a query, are fused with the name token. For example, *connect me with Heather* gets transcribed as *connect Merriweather*. In such cases, the phonemes of the name span in the transcript are different from those of the intended name. While we do not evaluate our method on such transcripts in this paper, they form an important error category in the real world. Fuzzy matching of characters and phonemes can help correct these cases.

The interactive demo UI enables exploration of these errors and of which features help correct for them, as shown in the video submission.

6 Conclusion

We present a novel approach for handling ASR errors in entity resolution, highlighting the advantages of using contextual features for the task. Our proposed approach shows promising results when resolving error-prone domain-specific entities in noisy ASR transcripts against an index of up to hundreds of thousands of terms. Our results on the use case of person name resolution for voice calling can generalize to many other use cases with a fixed set of resolvable terms, such as restaurant names for food ordering, song titles in a music player, or the resolution of actor names in a movie browsing voice assistant.

Acknowledgments

This work has benefited from the comments, code, and UI design contributions of Marvin Huang, Chris Liu, Jui-Pin Wang, Chad Oakley, Qian Yu.

References

Youssef Bassil and Paul Semaan. 2012. ASR context-sensitive error correction based on Microsoft n-gram dataset. *Journal of Computing*, 4(1):34–42.

William M Campbell, Lin Li, C Dagli, Joel Acevedo-Aviles, K Geyer, Joseph P Campbell, and C Priebe. 2016. Cross-domain entity resolution in social media. *arXiv preprint arXiv:1608.01386*.

Hao Cheng, Hao Fang, and Mari Ostendorf. 2015. Open-domain name error detection using a multi-task RNN. In *Proceedings of the 2015 EMNLP*, pages 737–746.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Katya Gonina, Navdeep Jaitly, Bao Li, Jan Chorowski, and Michiel Bacchiani. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE ICASSP*.

Rodolfo Corona, Jesse Thomason, and Raymond Mooney. 2017. Improving black-box speech recognition using semantic parsing. In *Proceedings of the 8th IJCNLP*, volume 2, pages 122–127.

Walter M. P. Daelemans and Antal P. J. van den Bosch. 1997. Language-independent data-oriented grapheme-to-phoneme conversion. In Jan P. H. van Santen, Joseph P. Olive, Richard W. Sproat, and Julia Hirschberg, editors, *Progress in Speech Synthesis*, pages 77–89. Springer New York, New York.

Yuqing Gao, Bhuvana Ramabhadran, Julian Chen, Hakan Erdogan, and Michael Picheny. 2001. Innovative approaches for large vocabulary name recognition. In *2001 IEEE ICASSP*, volume 1, pages 53–56.

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.

Timothy J Hazen, Theresa Burianek, Joseph Polifroni, and Stephanie Seneff. 2002. Recognition confidence scoring for use in speech understanding systems. *Computer Speech & Language*, 16:49–67.

Antoine Laurent, Sylvain Meignier, and Paul Deléglise. 2014. Improving recognition of proper nouns in ASR through generating and filtering phonetic transcriptions. *Computer Speech & Language*, 28(4):979–996.

Alex Marin, Mari Ostendorf, and Ji He. 2015. Learning phrase patterns for ASR name error detection using semantic similarity. In *INTERSPEECH-2015*, pages 1423–1427.

Fabrizio Morbini, Kartik Audhkhasi, Ron Artstein, Maarten Van Segbroeck, Kenji Sagae, Panayiotis Georgiou, David R Traum, and Shri Narayanan. 2012. A reranking approach for recognition and classification of speech input in conversational dialogue systems. In *2012 IEEE SLT*, pages 49–54. IEEE.

Lawrence Philips. 2000. The double metaphone search algorithm. *C/C++ Users J.*, 18(6):38–43.

Rohit Prasad, Rohit Kumar, Sankaranarayanan Ananthakrishnan, Wei Chen, Sanjika Hewavitharana, Matthew Roy, Frederick Choi, Aaron Challenner, Enoch Kan, Arvind Neelakantan, and Premkumar Natarajan. 2012. Active error detection and resolution for speech-to-speech translation. In *IWSLT 2012*.

Arushi Raghuvanshi, Lucien Carroll, and Karthik Raghunathan. 2018. Developing production-level conversational interfaces with shallow semantic parsing. In *Proceedings of the 2018 EMNLP: System Demonstrations*, pages 157–162.

Lufeng Zhai, Pascale Fung, Richard Schwartz, Marine Carpuat, and Dekai Wu. 2004. Using n-best lists for named entity recognition from Chinese speech. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 37–40.

Yong Zhao, Jinyu Li, Shixiong Zhang, Liping Chen, and Yifan Gong. 2018. Domain and speaker adaptation for Cortana speech recognition. In *2018 IEEE ICASSP*, pages 5984–5988.