March 2022

Decoders that Model Unknown Entities

Leonid Velikovich

Petar Aleksic

Decoders that Model Unknown Entities

ABSTRACT

A persistent challenge in noisy-channel decoding is the recognition of unusual or context-specific inputs such as named entities, e.g., names of personal contacts, songs, apps; specialized phrases, e.g., voice commands, technical/medical/legal jargon, etc. Noisy-channel decoding is a problem that occurs in automatic speech recognizers (ASR). This disclosure describes techniques to improve lattice diversity in noisy-channel decoding, resulting in richer decoding lattices. Decoding errors are reduced by using the decoder to model uncertain/unknown entities, e.g., by embedding uncertainty into the training data. Primary errors such as misrecognition of named entities become easier to fix because secondary errors are reduced or avoided. The resulting decoder is robust to words and phrases that are missing or uncommon in training data.

KEYWORDS

- Automatic speech recognition (ASR)
- Speech biasing
- Speech rewrite
- Virtual assistant
- Natural language processing (NLP)
- Noisy-channel decoding
- Virtual assistant correction
- Natural language understanding (NLU)
- Speech recognition lattice
- Speech recognition hypothesis
- Speech transcription
- Lattice decoding
- Context-specific speech recognition
- Sequence modeling
- Beam search
- Voice search

BACKGROUND

A persistent challenge in noisy-channel decoding is the recognition of unusual or context-specific inputs such as named entities, e.g., names of personal contacts, songs, apps; specialized phrases, e.g., voice commands, technical/medical/legal jargon, etc. Noisy-channel decoding is a problem that occurs in automatic speech recognizers (ASR), e.g., as used in smartphones, tablets, smart speakers, cars, or other devices with a voice interface; touchscreen keyboard inputs; natural language processing applications; etc.

Noisy-channel decoding often fails because unusual words are poorly modeled by default due to sparsity of training data; due to cross-lingual or rare words; due to the emergence of new entities (including creation by users); etc. As a result, the correct transcript in the ASR decoding lattice or hypothesis-set (also known as n-best list) either has a low rank or is missing.

Existing techniques to decode noisy channels typically use speech biasing, insertion of missing entities into recognition results (word lattice), etc. However, due to token-sequence modeling in an ASR decoder, an early misrecognition can cascade into additional downstream errors in recognition.

Example

User said: “call Beth on cell”

Hypothesis: “cold bath on sale”

In this example, an early misrecognition (‘bath’ for ‘Beth’) led to a subsequent misrecognition of ‘cell’ as ‘sale.’ The subsequent misrecognition is traced to the overfitting by the decoder to the history of ‘bath on,’ which is evidently followed frequently by the word ‘sale,’ that is phonetically close to ‘cell.’

For clarity, ASR errors are distinguished into primary and secondary types. Primary errors (misrecognition of a phrase) are hard cases that occur even when the surrounding words

are correctly recognized. For example, this may happen for an entity unknown to the decoder, such as the name of a contact or an obscure musical piece. Secondary errors are errors caused by a misleading word history. Secondary errors typically involve words/phrases that are familiar to the decoder (e.g., ‘cell’) but are deprioritized as a result of another error elsewhere (e.g., ‘cell’ misrecognized as ‘sale’). Secondary errors are potentially avoidable.

Even when a phrase is misrecognized (primary error), there is often a sufficient surrounding signal to recover. For example, ‘call lee oh need mobile’ can be fixed by replacing the middle part (‘lee oh need’) with a phonetically close entity (personal contact ‘Leonid’). However, with secondary errors added to the mix, recovery is harder, as more words are wrong; this also means the phonetics of the transcript are even further removed from the phonetic ground truth.

A related problem is lattice diversity, which refers to a decoding lattice featuring a rich set of paths, such that errors can be fixed by looking deeper into the lattice for word or phrase level alternatives. In practice, lattice diversity is often poor due to the ASR decoder’s use of beam search, where at every time step, poor-scoring partial hypotheses are removed to mitigate computational load. The removal of partial hypotheses early in the decoding procedure can remove a hypothesis that is temporarily poor-scoring but eventually close to the ground truth. The final lattice is sparse and includes few hypotheses, all largely similar to each other, and all removed from the ground truth. Indeed, it is possible for the correct hypothesis, identical to the groundtruth, to start off with a poor score and get pruned from the search beam for various reasons such as, for example: a noisy recording; poor starting diction; truncated recording due to speaking too soon; a slow start in the turning on of the microphone; etc.

Example

Transcript truth: “play Reforget by Lauv”

Hypothesis #1: “play we forget my love”

Hypothesis #2: “play we regret my love”

Hypothesis #3: “play we forget by love”

In this example, none of these hypotheses give us added value, as all are missing ‘Reforget’ and ‘Lauv.’ Indeed, partial hypotheses that included ‘Reforget’ and/or ‘Lauv’ did emerge early on during decoding, but were rapidly pruned due to the rarity of the words ‘Reforget’ and ‘Lauv.’

Lattice diversity is typically poor because, as the speech decoder progresses from the beginning to the end of an utterance (as in a unidirectional sequence model), its token-sequence modeling results in overfitting to prior word histories. Once the decoder progresses past the first one or two words of an utterance, its earlier chosen word paths become increasingly influential on later decoded words. The decoder is typically conditioned on earlier words, e.g., there is a conditional assumption that they were correctly recognized when predicting later words. When a phrase (such as a named entity) is not recalled by the decoder, there is no surviving hypothesis with correct prior-word history. That makes it more likely that other parts of the lattice/n-best-list include secondary errors.

Lattice diversity can also be poor in bidirectional sequence models, which use the words to the left *and* to the right of a given segment of utterance to transcribe the segment. Wrong token(s) in one place in the sequence affect the accuracy of surrounding tokens on *both* sides of the misrecognition. Furthermore, due to the total path score being computed over the entire sequence, bidirectional-style inaccuracies affect unidirectional models as well: even in

unidirectional models, secondary errors on *either* side of the primary error can yield an incorrect but high-ranked hypothesis.

DESCRIPTION

This disclosure describes techniques to improve lattice diversity in noisy-channel decoding, resulting in richer decoding lattices. Primary errors such as misrecognition of named entities become easier to fix because secondary errors are reduced or avoided. The resulting decoder is more robust to words and phrases that are missing or uncommon in training data.

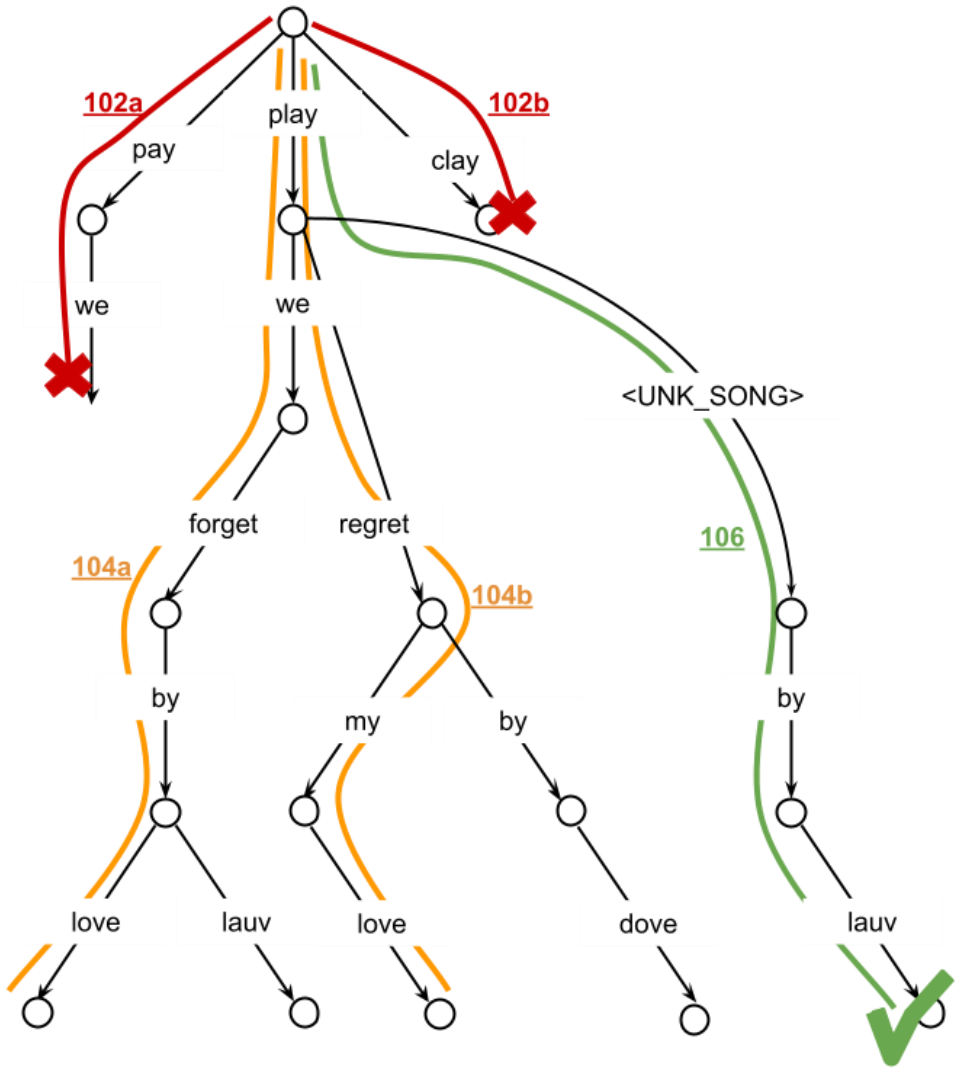

Fig. 1: A decoding lattice that incorporates uncertainty

In the previous example, ‘play Reforget by Lauv,’ even if ‘Reforget’ and ‘Lauv’ were unknown during training, per techniques of this disclosure, at inference, their likeness is embedded into the decoding lattice. This is illustrated in Fig. 1, which depicts a decoding lattice that incorporates uncertainty. During inference (decoding), paths such as 102a-b (red) die out naturally due to their low metrics. Incorrect paths such as 104a-b (orange), which might have survived over the correct hypothesis, now compete with a newly introduced path 106 (green), which incorporates uncertainty by replacing unrecognized words with a token that indicates unknown phrases - <UNK_SONG>.

In this manner, partial hypotheses including unusual words (‘Reforget,’ ‘Lauv’) are not prematurely rejected. Rather, they are re-introduced via unknown (<UNK>) tokens to maintain lattice diversity and are recovered using lattice-repair techniques. The correct hypothesis survives the end of decoding to compete viably with high-scoring but incorrect hypotheses. Errors are reduced by modeling within the decoder uncertain/unknown entities by embedding uncertainty into the training data, e.g., by replacing words with an ‘unknown’ (<UNK>) token.

Example

Training instance: “call Petar mobile”

Added instance: “<UNK> Petar mobile”

Added instance: “call <UNK> mobile”

Added instance: “call Petar <UNK>”

In this example, the decoder is trained to output <UNK> as an alternative to actual literal words. In cases when the literal words are incorrect, <UNK> produces a word history state unattached to a misrecognized word.

Example

User says: “call Petar mobile”

Decoder (without the described techniques): “call pete arm mobil”

Decoder (with the described techniques): One hypothesis is “call <UNK> mobile”

In this example, the secondary error of misrecognizing ‘mobile’ is averted, and the part corresponding to <UNK> later recovered and matched to ‘Petar’ using lattice repair techniques.

In practice, it is unnecessary to insert <UNK> as an alternative for each and every word or phrase. It can be sufficient to do so specifically for phrases that are expected to vary over time, geography, individual users, or in other ways that are not reflected during model training. For example, named entities can have alternative decoder paths with <UNK> tokens, as one may have a supply of use-case-specific named entities that can be inserted at inference time (to correct unavoidable primary decoder errors). Named entities can be annotated in the training data and replacement can be performed on just the spans of those named entities.

Example

Training instance: “call Petar mobile”

After annotation: “call <contacts> Petar </contacts> mobile”

Added instance: “call <UNK> mobile”

Added instance: “call <UNK_CONTACT> mobile”

In this example, a named entity in the training instance is identified and annotated, such that additional training instances including <UNK> tokens can be generated. As illustrated, <UNK> tokens can be further narrowed in scope, for example, an <UNK_CONTACT> token indicates an unknown word that is part of the user’s contact list. Similarly, there can be unknown tokens of type <UNK_SONG>, <UNK_ARTIST>, etc.

Alternative to following named entities, spans of utterances that include dynamic content can be identified using information-theoretic techniques, e.g., by comparing time slices of traffic

and computing the Kullback-Leibler divergence between the content of spans that otherwise have similar left and right context. The following of named entities can be somewhat easier, and a match for <UNK> data may be readily available when making corrections during inference.

Alternatively, phonemes can be introduced into named entities, which forces the speech decoder to output the phonetic representation of the unknown entity, simplifying its later correction, e.g., by matching to a relevant named entity known at inference time. The introduction of phonemes uses up modeling history with phoneme tokens, which may sacrifice the attention being given to the earlier word history.

Example

Training (original) instance: “Call Mary on cell”

Added (generated) instance: “Call <UNK_CONTACT> m E r \i </UNK_CONTACT> on cell”

In this example, a named entity ‘Mary’ is split into phonemes ‘m E r \i’ forcing the speech decoder to output the phonetic representation of the named entity.

In this manner, the techniques of this disclosure enable a noisy-channel decoder to represent unknown entities instead of being forced to emit an a priori wrong phrase (primary error) that locks the decoder into an incorrect word history, potentially leading to more (secondary) errors. The emitted unknown entity may further contain information about its type, e.g., the name of a personal contact, a musical piece, etc., and the phonetics, e.g., a sequence of phonemes, to further enable error recovery/correction at inference time. The techniques apply to unidirectional and bidirectional sequence models.

CONCLUSION

This disclosure describes techniques to improve lattice diversity in noisy-channel decoding, resulting in richer decoding lattices. Decoding errors are reduced by using the decoder to model uncertain/unknown entities, e.g., by embedding uncertainty into the training data. Primary errors such as misrecognition of named entities become easier to fix because secondary errors are reduced or avoided. The resulting decoder is robust to words and phrases that are missing or uncommon in training data.

REFERENCES

[1] Ramaswamy, Swaroop; Breiner, Theresa; Pisarev, Igor; Zivkovic, Dan; Chen, Mingqing; Mathews, Rajiv; and McConnaughey, Lara, “Personalizing Speech Recognition Based on User-entered Text”, Technical Disclosure Commons, (February 08, 2022)
https://www.tdcommons.org/dpubs_series/4887