Towards Relation Extraction from Speech

Tongtong Wu[*,◇,△] **Guitao Wang**[*,◇] **Jinming Zhao**[*,△] **Zhaoran Liu**[†]
Guilin Qi[◇] **Yuan-Fang Li**[△] **Gholamreza Haffari**[△]

[◇]Southeast University, China; [†]Zhejiang University, China; [△]Monash University, Australia

[◇]{wutong8023,220222117,gqi}@seu.edu.cn,[†]jiuyinlau@gmail.com

[△]{first_name.last_name}@monash.edu

Abstract

Relation extraction has focused on extracting semantic relationships between entities from the unstructured written textual data. However, with the vast and rapidly increasing amounts of spoken data, relation extraction from speech is an important but under-explored problem. In this paper, we propose a new information extraction task, speech relation extraction (SpeechRE). To facilitate further research, we construct the first synthetic training datasets, as well as the first human-spoken test set with native English speakers. We establish strong baseline performance for SpeechRE via two approaches. The pipeline approach connects a pretrained ASR module with a text-based relation extraction module. The end-to-end approach employs a cross-modal encoder-decoder architecture. Our comprehensive experiments reveal the relative strengths and weaknesses of these approaches, and shed light on important future directions in SpeechRE research. We share the source code and datasets on https://github.com/wutong8023/SpeechRE.

1 Introduction

Relation extraction (RE) (Han et al., 2020) is an important information extraction task, which aims at extracting structured semantic relations between entities from unstructured data, typically text. Besides text, there is also a plethora of speech data that is being continually produced. These include news reports, interviews, meetings and dialogues, to name a few. Extracting relations from speech is an important but under-explored problem.

In this work, we take the first step towards addressing relation extraction (RE) from speech, introducing a new information extraction task, Speech Relation Extraction (SpeechRE). The input for this task is raw audio and the output is one or more triplets, each of which representing a relation between a pair of two entities appearing in the speech, e.g., *[entity1, relation, entity2]*.

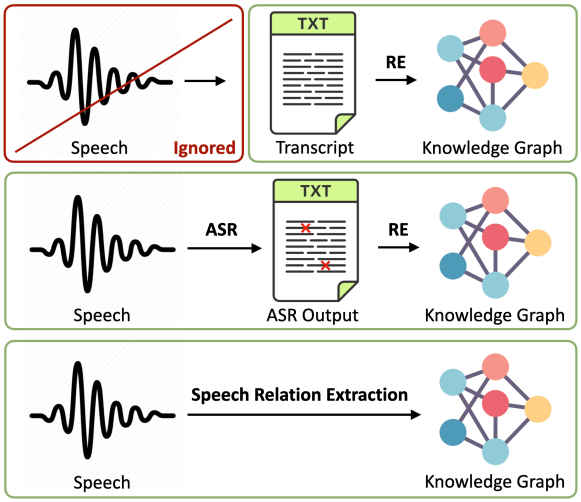

Figure 1: The comparison among conventional transcript-based relation extraction, ASR outputs-based relation extraction, and the end-to-end speech relation extraction.

SpeechRE and text-based RE (TextRE) both involve content understanding. The former is more challenging than the latter, mainly due to the characteristics of speech. (i) Speech carries much richer information beyond linguistic content (unlike text), for instance, emotion, speaker style and background noise; and it is non-trivial to disentangle the content element (Mohamed et al., 2022), which is needed for SpeechRE. (ii) Speech is continuous without sequence/word boundaries, implying the difficulty of determining the exact audio spans for target words (entity and relation). (iii) Audio signals are orders of magnitude longer than the corresponding transcripts, which makes speech encoding for long-span extraction more challenging due to more demanding hardware requirements, especially with Transformer (Vaswani et al., 2017).

In the absence of SpeechRE training data, we construct three benchmark datasets for this task by converting two commonly used TextRE

[*] denotes the equal contribution.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 10751–10762
December 7-11, 2022

datasets (i.e., CoNLL04 (Roth and Yih, 2004a) and ReTACRED (Stoica et al., 2021a) to speech with a SOTA text-to-speech (TTS) system. We then pair the synthetic speech with the corresponding target relation triplets as instances. To better evaluate model performance on real speeches, we also compile a human-read test set.

We approach SpeechRE with a pipeline method, SpeechRE$_{pipe}$, and an end-to-end (e2e) method, SpeechRE$_{e2e}$. In SpeechRE$_{pipe}$, we train our pipeline model with an automatic speech recognition (ASR) module that converts speech to text, followed by a RE module that extracts triplets. In SpeechRE$_{e2e}$, we build a single speech-to-text model that extracts triplets directly from speech. We use a SOTA pretrained speech encoder, WAV2VEC2 (W2V2) (Baevski et al., 2020) and BART (Lewis et al., 2020) decoder. Inspired by Li et al. (2021), we attach a length adaptor on top of the encoder to bridge the length mismatch between speech representation and text representation. The end-to-end approaches in speech processing tasks often face more severe data scarcity issues than the pipeline approaches (Sperber and Paulik, 2020), as the latter can essentially leverage massive ASR data and labelled data for the downstream text-based tasks. To tackle this challenge, we further propose two data augmentation techniques: upsampling via generating speech with different voices, and pseudo-labelling (He et al., 2021) by leveraging abundant ASR data and a SOTA TextRE system.

Our contributions can be summarized as follows.

- We present a new task, Speech Relation Extraction (SpeechRE). To support the development of this task, we create and release a synthetic SpeechRE dataset, including training/dev/test sets, as well as a human-read test set.
- We establish strong baseline performance via a pipeline approach and an e2e approach. Our extensive experiments identify a performance gap between TextRE and SpeechRE, and the gap between the pipeline approach and the e2e approach, motivating further research.
- Our analysis shows that the performance gap of the end-to-end approach mainly comes from the data scarcity problem and the difficulty of spoken name recognition. We propose two data augmentation methods to the problem.
- Based on our findings, we suggest three main directions for future exploration to advance speech relation extraction.

2 Related Work

2.1 Relation Extraction

As an essential component of information extraction, named entity recognition (NER) and relation extraction (RE) have attracted much attention in the research community. Relation extraction is usually studied as a natural language processing task of textual data (Nasar et al., 2021; Wu et al., 2021; Zheng et al., 2021c; Chen et al., 2022b). With the widespread of multimedia data on social media, some researchers have begun to explore relation extraction from data in other modalities such as images (Zheng et al., 2021a; Chen et al., 2022a; Zheng et al., 2021b). Although some work has focused on spoken language, such as dialogue relation extraction (Yu et al., 2020; Zhou et al., 2021), these studies are all based on transcripts, i.e. high-quality transcribed text from speech, which is still within the confines of text-based relation extraction. Moreover, given the transcribed text, the side information of voice, e.g., emotion, speaker identity is ignored from the spoken language.

2.2 Spoken Language Understanding

Spoken Language Understanding (SLU) aims to extract the meaning from speech utterances. It has wide applications from voice search to meeting summarization and has received great attention from industry and academia (Tur and De Mori, 2011). A typical SLU system involves mainly two tasks, i.e., intent detection and slot filling (Tur and De Mori, 2011). Traditionally, SLU systems have a pipeline structure, in which an ASR module is first used to convert speech to text and then a NLU system is deployed to determine semantics from text.

A major drawback of this approach is that each module is trained and optimized independently (Serdyuk et al., 2018). (i) The ASR model is optimized to minimized Word Error Rate (WER)[1], often equally weighting every word, whereas not every word has the same impact on SLU. (ii) The NLU model is trained on clean text without ASR errors, i.e. transcripts. During evaluation, however, it receives erroneous ASR outputs and these errors are propagated to NLU, impairing its performance. End-to-end (e2e) learning has

[1]WER is a commonly used metric in measuring the performance of ASR systems.

thus attracted interests from the community, for its potential to addressing SLU in a more principled way (Serdyuk et al., 2018). Since the first e2e approach proposed by Serdyuk et al. (2018), the field has made significant advances (Qin et al.) and many techniques, such as pretraining (Castellucci et al., 2019), have been proposed.

Similar to other speech processing tasks (e.g. speech translation (Sperber and Paulik, 2020)), SLU also faces the data scarcity issue, as it can be very expensive to annotate such a dataset, whereas the pipeline method can benefit from existing and emerging massive ASR data and NLU datasets.

Speech relation extraction (SpeechRE) is a new SLU task and thus inherits the merits and demerits of the pipeline and e2e approaches. We leverage advances that have been developed in related disciplines in this work and evaluate their relative strengths and weaknesses in §4.

3 Speech Relation Extraction

We define SpeechRE as a joint entity and relation extraction task that takes a speech utterance as the input and generates a set of relational triplets in the form of *[entity1, relation, entity2]* as the output.

In this section, we first describe the data construction method (§3.1). Next, we present our two approaches to the task (§3.2). Last, we describe our two data augmentation techniques (§3.3) to improve end-to-end SpeechRE performance.

3.1 Dataset Construction

Synthetic Data. As there is no readily available SpeechRE data, we generate SpeechRE data from existing TextRE corpora. Given a TextRE dataset consisting of pairs of <source (i.e. transcript), triplet> we convert the transcript to human-like speech with a TTS model. A typical TTS system comprises a Text-to-Spectrogram module, which takes discrete text as input and produces mel-spectrograms, and a vocoder, which converts the mel-spectrograms into waveforms. We choose Tacotron2-DCA as the TTS system and Mulitband-Melgan as the vocoder.[2] Once the synthesis process is complete, our data would contain triples of <synthetic speech, transcripts, triplets>. Training/dev/test sets are compiled following this process, while obeying the original TextRE split.

Real Data. To evaluate the performance of our models on realistic speech, we randomly choose 200 instances from the ReTACRED10 test set and engage a native English speaker to read the corresponding transcripts. This real SpeechRE test set can be used as a benchmark for future research. Please refer to §4.8 for demonstration of synthetic and real data.

[2] https://github.com/mozilla/TTS

3.2 SpeechRE Approaches

We describe our pipeline (SpeechRE$_{pipe}$) and end-to-end (SpeechRE$_{e2e}$) approaches in this section. As depicted in Figure 2, SpeechRE$_{pipe}$ consists of an ASR module for turning speech into text and a TextRE module for extracting triplets from the text, whereas SpeechRE$_{e2e}$ has a simple architecture with a speech encoder, a length adaptor and a text decoder, which outputs triplets directly.

The Pipeline Approach. We use W2V2-large as our ASR module. It is a speech encoder, pretrained in a self-supervised manner. Its architecture starts with a feature encoder composed of several 1D convolutional neural networks that process raw waveforms and emits latent speech representations. Following that, a quantization module is attached to extract discrete latent vectors. Next, a context encoder made of 24 Transformer (Vaswani et al., 2017) layers is used to learn contextualized representations from masked outputs of the feature encoder. The whole model is optimized to discriminate a true masked vector from the ones produced by the model. After pre-training, only the feature and context encoders are retained and used for downstream tasks. Compared with other ASR models, W2V2 obtains superior performance by fine-tuning it with a small amount of labelled speech. Additionally, it works on raw audio signals directly, avoiding the risk of information loss using hand-crafted features (Latif et al., 2020). The W2V2 model we use is already fine-tuned on ASR data and we do not further fine-tune it.

We utilize REBEL (Cabot and Navigli, 2021) as our TextRE module. It uses a pretrained language model BART (Lewis et al., 2020) as the backbone and treats TextRE as a text generation task. Concretely, the input is text (the output from ASR in our case), and the output is linearized triplets, in the form of *"<triplet> entity1 <subj> entity2 <obj> relation"*. The generative model may not restrict the output entities exactly the same with a mention in input text, which is an advantage for SpeechRE with ASR outputs, such a task extracting from the text containing noisy entity mentions.

An alternative to REBEL is to employ a classification-based model as our TextRE module. We experiment with Spert (Eberts and Ulges, 2020)

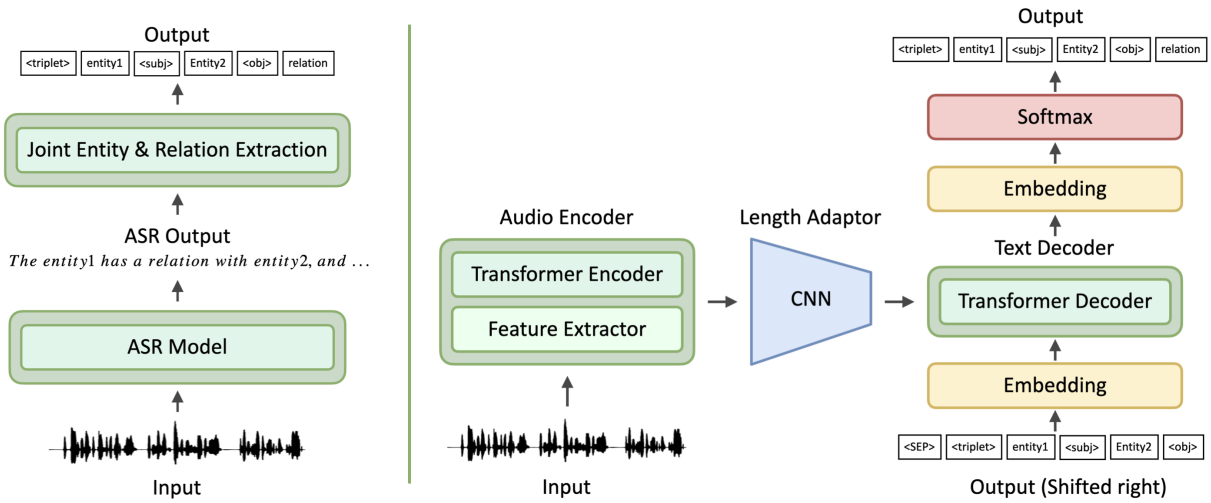

Figure 2: Overview of SpeechRE$_{pipe}$ (left) and SpeechRE$_{e2e}$ (right).

and TPlinker (Wang et al., 2020) in this work. Given that the ground-truth entity may not appear in the transcribed text computed by ASR, we modify the ground-truth entity in the training set by referring to the fuzzy-matched longest substring mentioned in transcribed text, such a fuzzy-matched longest substring is measured by Levinstein distance.[3]

The end-to-end Approach We formulate SpeechRE$_{e2e}$ as a speech-to-text task that requires a speech encoder and a text decoder. We employ the aforementioned W2V2 as our encoder, for its capabilities of encoding general-purpose knowledge. We take the decoder component of BART-large (Lewis et al., 2020) as the text decoder. Naïvely jointing them may lead to optimization issues, as they are pretrained on different modalities which differ significantly in length. To address this issue, inspired by Li et al. (2021), we introduce a length adaptor made of n number of 1-d convolutional layers, each of which is parametrized with kernel p, stride s and padding p. This adaptor has an sequence reduction effect of $\sim s^n$.

We follow the partial training strategy used by Gállego et al. (2021). We train the length adaptor together with part of the encoder and decoder (including encoder self-attention, encoder-decoder cross-attention and layer normalization), while freezing the rest of the parameters. The trained parameters account for $\sim 20\%$ of the entire model. This training strategy has shown to be efficient, while retaining performance in speech translation tasks (Zhao et al., 2022).

3.3 Data Augmentation for Speech Relation Extraction

To address the data scarcity issue facing SpeechRE$_{e2e}$, we propose two data augmentation techniques: upsampling and pseudo-labelling. For *upsampling*, given a SpeechRE corpus, we use a multi-speaker TTS system (Kim et al., 2021) and generate synthetic speeches with 4 different voices. This yields 4 more synthetic SpeechRE datasets. *Pseudo-labelling* has been widely used in NLP (He et al., 2021) due to its effectiveness in improving task performance. Specifically, given a SpeechRE dataset $\boldsymbol{D}$, we fine-tune the pretrained REBEL model on <transcript, triplet> training instances to adapt it to the current domain. Next, we run the fine-tuned REBEL over the large-scale English dataset of CommonVoice (V9)[4] where audios are recorded by volunteers of different demographic characteristics. Together with the original speech, this gives a total of 922k instances containing <real_speech, transcript, pseudo_triplet>. Then, we filter out noisy data if a pseudo_triplet meets any of the following criteria: i) relation is "no_relation"; 2) no subject/object entity is generated; and 3) subject and object entities are both pronouns. We thus obtain 380k clean instances. Depending on the type of relations available in $\boldsymbol{D}$, further filtering may be applied to remove spurious relation triplets.

[3] https://pypi.org/project/fuzzywuzzy/

[4] https://commonvoice.mozilla.org/en/datasets

4 Experiment

4.1 Dataset

We conduct experiments on our proposed SpeechRE datasets, i.e., Speech-CoNLL04 and Speech-ReTACRED, aligning with their original dataset CoNLL04 (Roth and Yih, 2004b) and ReTACRED (Stoica et al., 2021b). Furthermore, we pick 10 relations with the largest number of instances from the ReTACRED dataset, and remove the instances with none of relation or containing the other 30 relations. We named the sub-dataset of ReTACRED as ReTACRED10, and utilize it as the test-bed for sufficient supervised learning. We detail data statistics in Table 1.

Moreover, we use ReTACRED10 to fine-tune REBEL for pseudo-labelling. The fine-tuned model generates pseudo labels of 137 relations. We remove pseudo instances whose labels fall out of ReTACRED10. We then have 363k instances remained and each instance has one triple. Furthermore, we sample from pseudo set to 1.8× for each relation in ReTACRED10. At this point, the total number of data sampled is 2.5× to ReTACRED10, as a transcript in ReTACRED10 often has multiple relations.

4.2 Baselines

We select three joint entity and relation extraction methods as baselines: **TP-Linker** (Wang et al., 2020) formulates joint extraction as a token pair linking problem and introduces a handshaking tagging scheme that aligns the boundary tokens of entity pairs under each relation type. **Spert** (Eberts and Ulges, 2020) formulates the task as a two-stage classification task, with classifying each candidate continuous span for entity detection and then classifying the inter-context for relation classification. **REBEL** (Cabot and Navigli, 2021) treats joint entity and relation extraction as a text generation task. We also attempt these three methods as the pluggable TextRE modules in SpeechRE$_{pipe}$.

4.3 Evaluation metrics

Evaluating SpeechRE is difficulty because of the strict matching of entities. The error of a letter or the difference in case lead to failure in entity matching, which lowers the results of triplets. For this reason, we evaluate the baseline models and SpeechRE models based on the metrics (i.e., Recall, Precision and micro-F1) commonly used in TextRE, with modifications. Specific to entities, we ignore the span of entities due to the lack of span information in audio, and TextRE applies the same method. Additionally, we do not consider entity when evaluating relations, which is equivalent to the task of relation classification of sentences. The reason is that spoken entity recognition is a very difficult tasks as most entities have low frequency in a dataset; when predicted entities were taken into account, the results would cover the true performance of relation generation. When evaluating triplets, we make sure that the head entity and tail entity and the relation between them are all correct.

4.4 Implementation Details

We use a pretrained W2V2 model[5] to convert speech to text, without fine-tuning it. Since the ASR outputs the model produces are all lower-cased without punctuation, we perform post-processing on the outputs for punctuation restoring and casing with another pretrained model.[6] For TextRE model, we mostly follow the instructions in Cabot and Navigli (2021) and start from the REBEL[7] that using Bart[8] as the pretrained model. The original REBEL labels the entities in the input text using punctuation marks to indicate entities' position in the input. Since SpeechRE$_{pipe}$ does not use labels to bias the model with entity information from plain audio or text, we remove the entity labels when preprocessing ReTACRED (source sentences) and ReTACRED10.

To train our SpeechRE$_{e2e}$ model, we use the pretrained W2V2 large[9] and the pretrained Bart[10]. We keep the W2V2 feature extractor frozen. We set kernel size, stride and padding to 3, 2, 1 for all 3 CNN layers for the length adaptor. We apply data augmentation (Potapczyk et al., 2019) on the audio data on the fly by applying the effects of "tempo" and "pitch" to change the speech speed, and "echo" to simulate echoing in large rooms. We train our SpeechRE models for $23k$ updates and set early stopping of 20 updates. We use Adam (Kingma and Ba, 2015) optimizer with parameters (0.99, 0.98), while setting clip norm to 20. We use the learning rate to 1e-4, monitored by a tri-stage scheduler.

[5] https://huggingface.co/facebook/wav2vec2-large-960h-lv60-self

[6] https://huggingface.co/flexudy/t5-small-wav2vec2-grammar-fixer

[7] https://github.com/Babelscape/rebel

[8] https://huggingface.co/facebook/bart-large

[9] https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_vox_960h_pl.pt

[10] https://dl.fbaipublicfiles.com/fairseq/models/bart.large.tar.gz

Datasets	# Relations	# Instances (train ‖ dev ‖ test)	# Triplets (train ‖ dev ‖ test)	# Avg. tokens (in transcripts)	# Avg. audio length (in seconds)
CoNLL04	5	922 ‖ 231 ‖ 288	1,283 ‖ 343 ‖ 422	29.1	11.3
ReTACRED	40	33,477 ‖ 9,350 ‖ 5,805	58,465 ‖ 19,584 ‖ 13,418	36.3	12.9
ReTACRED10	10	11,116 ‖ 3,892 ‖ 2,513	15,665 ‖ 5,970 ‖ 4,204	34.7	12.6

Table 1: Dataset statistics.

Method		CONLL04			ReTACRED			ReTACRED10		
		Entity	Relation	Triplet	Entity	Relation	Triplet	Entity	Relation	Triplet
TextRE	TP-Linker	78.63	83.49	58.56	50.46	51.83	20.39	65.51	65.17	37.01
	Spert	76.38	81.83	63.45	60.26	63.48	21.46	64.88	64.72	34.61
	REBEL	85.36	89.86	71.46	60.09	65.15	25.15	64.91	69.80	39.68
SpeechRE$_{pipe}$	TP-Linker$_{pipe}$	32.41	77.54	8.70	28.60	51.43	6.77	38.19	61.85	13.79
	Spert$_{pipe}$	28.95	75.44	10.47	33.20	58.36	7.10	55.23	57.42	27.43
	REBEL$_{pipe}$	35.78	82.86	12.53	30.21	53.20	6.93	51.08	67.46	28.06
SpeechRE$_{e2e}$		24.89	59.57	12.50	27.70	52.10	6.59	29.87	51.32	14.79

Table 2: Main results. **Upper rows**: TextRE models for which inputs are transcripts. **Middle rows**: SpeechRE$_{pipe}$ where inputs are ASR outputs. **Bottom row**: SpeechRE$_{e2e}$ where inputs are speech.

All experiments are conducted with fairseq.[11]. All our models are evaluated on the best performing checkpoint on the validation set. All experiments are conducted in a V100 GPU. Full training details can be found in *Appendix* A.1.

4.5 Results of TextRE and SpeechRE

We first compare and contrast among the text relation extraction method and two speech extraction methods, to understand the performance gap. We train various models, including three TextRE models, the pipeline version of them[12] and the e2e model, over CoNLL04, ReTACRED and ReTACRED10. Results are summarized in Table 2.

Despite of the good performance of REBEL with transcripts being inputs (up to 71.46 on CoNLL04), its performance drops hugely when the input becomes ASR outputs, which are erroneous. Notably, the performance gap on entity prediction is huge. These highlight the challenge with the pipeline approach. SpeechRE$_{e2e}$ does not outperform SpeechRE$_{pipe}$ across the three datasets.

All SpeechRE methods have achieved low accuracy of entity recognition. Particularly, the gap between TextRE and SpeechRE on entity detection is far larger than the gap on relation classification. It suggests that speech entity recognition may be the core bottleneck of the performance degradation of triplet extraction.

[11] https://github.com/facebookresearch/fairseq

[12] Where the input is changed to ASR outputs, instead of transcripts in TextRE.

4.6 SpeechRE in Low-resource Scenarios

To evaluate and compare the performance of our SpeechRE models in resource-constrained conditions, we simulate different training conditions by sampling 20%, 40%, 60%, 80% (and 100%) from the ReTACRED10 training set. We use REBEL as the TextRE model, since both REBEL and our SpeechRE models are generative models. For evaluation, we randomly sample 20 instances for each relation (10 relations) from the ReTACRED10 test set, totally 200 instances (the same subset corresponding to our human-read test set, described in §3.1). We present F1 scores on entity prediction in the left plot of Figure 3, and F1 scores on relation prediction in the right plot, in both of which the lines left to the red dashed vertical line refer to the setting discussed in this subsection. To measure the extent of error propagation in SpeechRE$_{pipe}$, we also evaluate its performance when the TextRE modules are trained on noisy ASR output as their input (instead of ground-truth transcripts). We summarize our observations from different perspectives below.

Training with transcripts v.s. ASR outputs. To investigate the impact of the quality of the text input to REBEL, we compare the performance of REBEL and SpeechRE$_{pipe}$ models, referring to TextRE (Transcript, TTS) and SpeechRE$_{e2e}$ (Single speaker, TTS) in Figure 3, whose inputs are transcripts and ASR outputs, respectively. Overall, the SpeechRE$_{pipe}$ model, compared to REBEL, produces comparable, yet slightly lower results on relation prediction, whereas performing significantly worse on entity prediction.

On the one hand, this indicates the reliability of transcribed texts on relation words. On the other hand, the significant ASR errors on entity words are propagated to the downstream extraction module, greatly degrading its performance.

Pipeline v.s. end-to-end SpeechRE. Comparing the two approaches, SpeechRE$_{e2e}$ performs worse than SpeechRE$_{pipe}$, referring to SpeechRE$_{pipe}$ (ASR, TTS) and SpeechRE$_{e2e}$ (Single speaker,TTS) in Figure 3. However, with the increase in training data, its performance starts to catch up with SpeechRE$_{pipe}$. This is expected[13], because not only does SpeechRE$_{pipe}$ have a bigger model size, its two components also excel in their own tasks by leveraging abundant ASR and TextRE data. In comparison, SpeechRE$_{e2e}$ has a smaller model size and is trained with a much smaller training set. Despite the large gap, the rising trend is promising, indicating the potential of SpeechRE$_{e2e}$ reaching parity with, and even surpassing SpeechRE$_{pipe}$.

Training SpeechRE$_{e2e}$ with multi-speaker v.s. single-speaker. We examine the impacts of single-speaker and multiple-speaker data. In most cases, when a model is trained with multi-speaker data, it has better performance on relational classification than the one trained on single-speaker data. Their performance on entity recognition is roughly the same.

Evaluation on synthetic and human-read data. When comparing the performance of our models on the synthetic test set and the human-read test set, it is surprising to obverse that most of the time, both SpeechRE$_{pipe}$ and SpeechRE$_{e2e}$ models have higher accuracy on relation prediction on the human-read data than on the synthetic one. This demonstrates the effectiveness of the use of synthetic speech. Please see *Appendix* A.2 for full results.

4.7 SpeechRE with Data Augmentation

Based on the trend observed previously, we expect the SpeechRE$_{e2e}$ model to improve with more training data. We leverage the two data augmentation methods introduced in §3.3, namely, *up-sampling* and *pseudo-labelling*. For each method, we build larger training corpora by adding augmented SpeechRE data to ReTACRED10, with sizes 100%, 200%, 250%, 300% and 350% that of ReTACRED10. This gives us 10 new training sets. The evaluation protocol is identical to the one in §3.3.

[13]The trend has long been observed in other speech processing tasks (Sperber and Paulik, 2020).

Results of these data augmentation can be found in Figure 3, to the right of the vertical dashed line in each subfigure. We outline our findings below.

Training with transcripts v.s. ASR Outputs. With more training data, REBEL trained on ground-truth transcripts plus augmented pairs, <transcript, pseudo_triplet>, has roughly the same accuracy on relation prediction in all conditions. We can observe a slightly decreasing trend on entity prediction. The performance of SpeechRE$_{pipe}$ has a moderately rising trend before leveling out.

Pipeline v.s. end-to-end SpeechRE. Both data augmentation techniques bring significant improvements to SpeechRE$_{e2e}$ with *pseudo-labelling* being superior. *Pseudo-labelling* reaches the same performance both on entity and relation predictions as TextRE on synthetic speech at 350%. The results are surprising, especially with entity generation, considering the difficulty of the task in the speech domain in general. In contrast, augmented data do not help much with SpeechRE$_{pipe}$ due to the error prorogation issue discussed above. Please see *Appendix* A.3 and A.4 for full results.

4.8 Case Study

We perform a qualitative error analysis of SpeechRE through a case study. Table 3 shows typical errors in this task.

Error accumulation in the pipeline method. The two rows "TTS ASR" and "Human ASR" illustrate that it is challenging for the SOTA ASR model to spell entity names correctly, especially the names of people and institutions. Being a deep learning model, it may tend to generate high-frequency words (Razeghi et al., 2022). This presents both a great challenge and opportunity for entity-sensitive tasks such as relation extraction, since low-frequency entities often contain more information and are more likely to be useful knowledge.

Hallucination in the end-to-end method. As shown in the two rows "SpeechRE$_{e2e}$", the e2e model may generate entities and relation types that are not present in speech, creating hallucinations. TextRE, in contrast, can restrict generated words via controllable text generation techniques. This is less surprising: being a cross-modal task, it is difficult for a SpeechRE model to effectively restrict the generated content, especially when the size of training data is limited. We also detail the impact of data augmentation on the accuracy of entity prediction by entity type. *Appendix* A.4 contains further results.

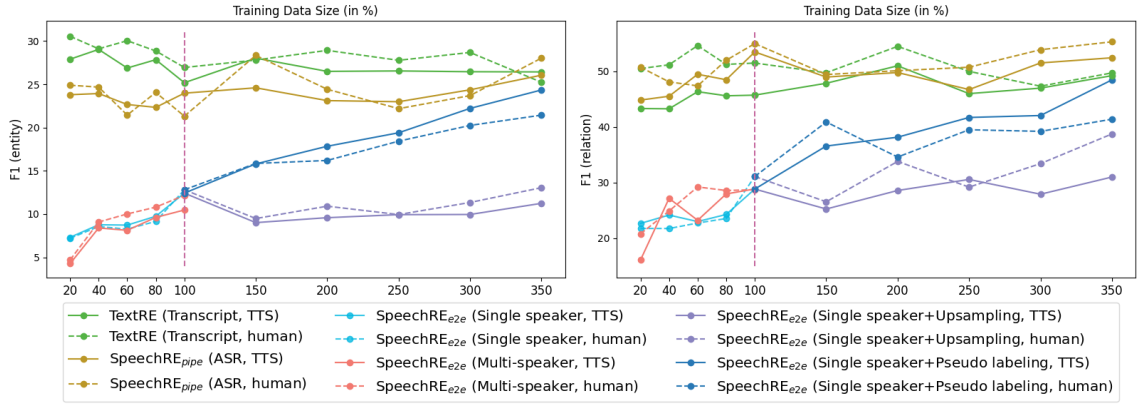

Figure 3: F1 scores of entity (left) and relation (right) predictions on 200 synthetic and human-read instances in various training resource conditions. **Left to vertical dashed lines**: low-resource scenarios. **Right to vertical dashed lines**: data augmentation. **MODEL** (*Train*, *Test*): MODEL (i.e. SpeechRE and TextRE) is trained on *Train* and tested on *Test*.

Text	When bin Laden fled the US invasion in 2001 , he took refuge with Haqqani in a safe house between the Afghan city of Khost and Miran Shah, according to Pakistani author Ahmed Rashid.	Golden	<triplet> Ahmed Rashid <subj> Pakistani <obj> person origin
TTS ASR	When bin-laden fled the U-S invasion in 2001, he took refuge with Hakone in a safe house between the Afghan City of Coast and Muran Shaw, according to Pakistani author Akmed Rashid.	SpeechRE$_{pipe}$	<triplet> Akmed Rashid <subj> Pakistani <obj> person origin
Human ASR	When bin Laden fled the U-S invasion in 2001, he took refuge with Hakwani in a safe house between the Afghan City of Cost and Mirishah, according to Pakistani author Ahmed Rashid.	SpeechRE$_{pipe}$	<triplet> Ahmed Rashid <subj> author <obj> person title
TTS Audio	Synthetic Audio	SpeechRE$_{e2e}$	<triplet>Bernama<subj>U.S.<obj>organization country of branch
Human Audio	Human Audio	SpeechRE$_{e2e}$	<triplet>Mohamed ElBaradei<subj>Sultan<obj>person title

Table 3: A qualitative error analysis for both the pipeline and end-to-end approaches. Models are trained with 100% ReTACRED10 data.

5 Discussion

Based on our analysis, we discuss the following question: **For a new SpeechRE task, should we choose a pipeline or an end-to-end approach?** While raw performance is largely attributed to data resources, to answer this question, other factors need to be taken into account in addition to the availability of data resources. These include compute power and latency.

Pipeline method is suitable in the low-resource scenarios. As shown in Figure 3, prior to 100%, SpeechRE$_{pipe}$ requires less training data to train than SpeechRE$_{e2e}$, while exhibiting reasonably good performance. Therefore, the general ASR method based on pre-training provides a reasonable performance lower bound for low-resource speech extraction. The major concern is errors contained in entities, as shown in §4.8. As a future direction, we conjecture that this issue could be potentially alleviated by the mixed extraction method from both transcript and speech. Yet, the pipeline approach may be limited by fundamental issues (e.g. error propagation and high latency) that cannot be solved easily.

The end-to-end method is preferred when labelled training data size is sufficient or external data is accessible. According to Figure 3, with the increasing volumes in training set, the performance of SpeechRE$_{e2e}$ on extracting correct entities and relations steadily improves. Extracting meaning from speech directly avoids the risk of information loss and error propagation, unlike in the pipeline setting. Because of this, the e2e approach can potentially solve the extraction task in a principled manner. The data scarcity issue that it faces can be eased through data augmentation, for its effectiveness on both machine-generated speech and realistic speech. Exploring more sophisticated augmentation and filtering techniques is thus a fruitful future direction. Further, it is of importance to improve data efficiency and enhance entity prediction performance. Particularly, enforcing constraints on entities in the decoding process and the inclusion of memory banks are promising directions.

6 Conclusion

We propose a new spoken language understanding task, Speech Relation Extraction (SpeechRE), and present two synthetic datasets and a human-read test set. We approach SpeechRE with two methods, the pipeline and e2e approaches. Through extensive experiments, quantitative and qualitative analyses,

we identify data scarcity and spoken entity recognition as two main challenges for this task. We then present two augmentation techniques that are effective in addressing these challenges. Lastly, being the first working on the task, we outline key directions for future research.

7 Limitations

This paper discusses the utterance-level speech relation extraction task where the average length of audio inputs is less than 15s. Constrained by computing resources, processing long audio signals is challenging, a known issue in the speech domain. For this reason, while speech relations can be useful in other scenarios such as summarization of dialogues, news and meetings, we were not in the position to carry out our study in these scenes. Another limitation is that we did not fully utilize the information contained in speech signals (e.g. speaker style and emotion), which could be beneficial to the task. Addressing these two limitations is part of our plan for future research.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. Rebel: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381.

Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. Multi-lingual intent detection and slot filling in a joint bert-based model. *arXiv preprint arXiv:1907.02884*.

Xiang Chen, Ningyu Zhang, Lei Li, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022a. Good visual guidance makes a better extractor: Hierarchical visual prefix for multimodal entity and relation extraction. *CoRR*, abs/2205.03521.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022b. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of WWW*, pages 2778–2788.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *Proceedings of ECAI*, pages 2006–2013.

Gerard I Gállego, Ioannis Tsiamas, Carlos Escolano, José AR Fonollosa, and Marta R Costa-jussà. 2021. End-to-end speech translation with pre-trained models and adapters: Upc at iwslt 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 110–119.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction. In *Proceedings of AACL*, pages 745–758.

Xuanli He, Islam Nassar, Jamie Ryan Kiros, Gholamreza Haffari, and Mohammad Norouzi. 2021. Generate, annotate, and learn: Generative models advance self-training and knowledge distillation.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, Junaid Qadir, and Björn W Schuller. 2020. Deep representation learning in speech processing: Challenges, recent advances, and future trends. *arXiv preprint arXiv:2001.00378*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Multilingual speech translation from efficient finetuning of pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 827–838.

Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, et al. 2022. Self-supervised speech representation learning: A review. *arXiv preprint arXiv:2205.10643*.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2021. Named entity recognition and relation extraction: State-of-the-art. *ACM Computing Surveys (CSUR)*, 54(1):1–39.

Tomasz Potapczyk, Paweł Przybysz, Marcin Chochowski, and Artur Szumaczuk. 2019. Samsung's system for the iwslt 2019 end-to-end speech translation task. In *Proceedings of the 16th International Conference on Spoken Language Translation*.

Libo Qin, Tianbao Xie, Wanxiang Che, and Ting Liu. A survey on spoken language understanding: Recent advances and new frontiers.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*.

Dan Roth and Wen-tau Yih. 2004a. A linear programming formulation for global inference in natural language tasks. Technical report, Illinois Univ at Urbana-Champaign Dept of Computer Science.

Dan Roth and Wen-tau Yih. 2004b. A linear programming formulation for global inference in natural language tasks. In *Proceedings of NAACL*, pages 1–8.

Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio. 2018. Towards end-to-end spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5754–5758. IEEE.

Matthias Sperber and Matthias Paulik. 2020. Speech translation and the end-to-end promise: Taking stock of where we are. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7409–7421.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021a. Re-tacred: Addressing shortcomings of the tacred dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13843–13850.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021b. Re-tacred: Addressing shortcomings of the TACRED dataset. In *Proceedings of AAAI*, pages 13843–13850.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. Tplinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1572–1582.

Tongtong Wu, Xuekai Li, Yuan-Fang Li, Gholamreza Haffari, Guilin Qi, Yujin Zhu, and Guoqiang Xu. 2021. Curriculum-meta learning for order-robust continual relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10363–10369.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4927–4940.

Jinming Zhao, Hao Yang, Ehsan Shareghi, and Gholamreza Haffari. 2022. M-adapter: Modality adaptation for end-to-end speech-to-text translation. *arXiv preprint arXiv:2207.00952*.

Changmeng Zheng, Junhao Feng, Ze Fu, Yi Cai, Qing Li, and Tao Wang. 2021a. Multimodal relation extraction with efficient graph alignment. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5298–5306.

Changmeng Zheng, Junhao Feng, Ze Fu, Yi Cai, Qing Li, and Tao Wang. 2021b. Multimodal relation extraction with efficient graph alignment. In *Proceedings of MM*, pages 5298–5306.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. 2021c. PRGC: Potential relation and global correspondence based joint relational triple extraction. In *Proceedings of ACL*, pages 6225–6235.

Mengjia Zhou, Donghong Ji, and Fei Li. 2021. Relation extraction in dialogues: A deep learning model based on the generality and specialty of dialogue text. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:2015–2026.

A Appendix

A.1 Training Details

A.1.1 Implementation Details

For all the experiments, we train our REBEL for 30 epochs with Adam optimizer (0.9, 0.999) of a linear scheduler with a warmup rate of 0.1, a learning rate of 5e-5, a weight decay of 0.01, and a gradient clip value of 10. For other settings, our REBEL is consistent with the original ones.

A.1.2 More details about pseudo-labelling

The Common Voice Corpus 9.0 dataset consists of 2,224 validated hours in English and 81,085 voices. Each entry in the dataset consists of a unique MP3 and corresponding text file. Many of the recorded hours in the dataset also include demographic metadata like age, sex, and accent that can help train speech recognition engines. Here we use the script from speechbrain[14] to help us process text files in the dataset, but we made two changes to the processing script. Firstly, this processing script will make all text uppercase, which we do not do, but retain the original case of the text. Secondly, we add full stops to all sentences, whereas the original text has no full stops. We have made these two changes to make the processed text more realistic and to harmonise it with other datasets (e.g. CoNLL04, ReTACRED, etc.). We fine-tune REBEL (using rebel-large[15] as the pretrained model) on the ReTACRED10 dataset and conduct pseudo-labelling on the processed text to extract sentences and corresponding audio that contains target relations. A total of 922k instances were extracted from the Common Voice Corpus 9.0 dataset, of which 380k clean instances were retained after filtering.

A.2 Low-Resource Analysis

We report the exact values of low resource analysis in Table 4, which corresponds to the left half of each sub-figure of Figure 3.

A.3 Data Augmentation Analysis

We report the exact values of low resource analysis in Table 4, which corresponds to the right half of each sub-figure of Figure 3.

A.4 Entity Analysis

To further understand why the pseudo labelling can perform better than the multi-speaker up-sampling, we conduct the following analysis experiments. Firstly, we selected high-frequency entities with frequency greater than three from the test set of ReTACRED10. Moreover, we count the frequency of these entities in the training set constructed by two augmentation manners, i.e., pseudo labelling and multi-speaker up-sampling (as shown in Table 6). Then, we counted the classification accuracy of these entities in the test set (in Table 7). Comparing the two tables by location, we can observe that the method of pseudo labelling can effectively improve the recognition accuracy of the model for unseen entities.

[14] https://github.com/speechbrain/speechbrain/blob/develop/recipes/CommonVoice/common_voice_prepare.py

[15] https://huggingface.co/Babelscape/rebel-large

Method	Input	Metrics	20%		40%		60%		80%		100%	
			w/TTS	w/Human	w/TTS	w/Human	w/TTS	w/Human	w/TTS	w/Human	w/TTS	w/Human
TextRE	Transcript	Entity	27.91	30.56	29.04	29.11	26.89	30.05	27.85	28.86	25.19	26.95
		Relation	43.32	50.50	43.27	51.19	46.35	54.64	45.61	51.26	45.73	51.50
		Triplet	6.55	9.50	6.86	8.45	5.54	8.52	8.02	8.54	4.02	7.50
SpeechRE-pipe	ASR	Entity	23.80	24.90	23.94	24.68	22.67	21.45	22.34	24.06	23.98	21.32
		Relation	44.84	50.75	45.53	48.11	49.48	47.40	48.50	52.00	53.50	55.00
		Triplet	4.53	4.02	7.05	5.41	4.64	3.65	4.50	4.50	6.00	4.50
SpeechRE-e2e	Single Speaker	Entity	7.31	7.2	8.76	8.55	8.72	8.26	9.74	9.15	12.42	12.81
		Relation	22.65	21.74	24.15	21.73	22.98	22.72	24.26	23.53	28.83	31.09
		Triplet	0.45	0.45	1.87	0.96	1.9	1.44	2.78	2.34	2.95	2.16
SpeechRE-e2e	Multi Speaker	Entity	4.26	4.69	8.39	9.08	8.12	10.01	9.61	10.8	10.47	12.16
		Relation	16.06	20.77	27.16	24.88	23.21	29.21	27.95	28.57	28.85	28.71
		Triplet	0.95	0.95	2.35	1.42	1.45	1.91	2.27	2.67	2.27	1.83

Table 4: The low resource analysis.

Method	Input	Metrics	100%		100% + 50%		100% + 100%		100% + 150%		100% + 200%		100% + 250%	
			w/TTS	w/Human	w/TTS	w/Human	w/TTS	w/Human	w/TTS	w/Human	w/TTS	w/Human	w/TTS	w/Human
TextRE	Text + Pseudo Labeling	Entity	25.19	26.95	28.03	27.81	26.50	28.93	26.55	27.78	26.46	28.69	26.44	25.24
		Relation	45.73	51.50	47.86	49.75	51.00	54.50	46.00	50.00	46.99	47.34	49.23	49.73
		Triplet	4.02	7.50	1.40	8.04	6.50	8.50	5.00	7.50	7.18	8.51	7.42	7.41
SpeechRE-pipe	Transcript + Pseudo Labeling	Entity	23.98	21.32	24.60	28.35	23.12	24.43	22.99	22.19	24.37	23.70	26.06	28.04
		Relation	53.50	55.00	48.98	49.44	49.79	50.12	46.73	50.75	51.52	53.90	52.45	55.32
		Triplet	6.00	4.50	4.67	4.88	6.56	7.01	3.52	3.52	6.06	7.56	6.18	5.25
SpeechRE-e2e	One Speaker + Upsampling	Entity	12.42	12.81	9.01	9.48	9.58	10.91	9.93	9.94	9.95	11.34	11.24	13.05
		Relation	28.83	31.09	25.24	26.54	28.57	33.8	30.56	29.18	27.9	33.41	30.99	38.74
		Triplet	2.95	2.16	2.22	2.65	2.19	2.17	2.14	2.59	1.37	0.91	1.87	1.87
SpeechRE-e2e	One Speaker + Pseudo Labeling	Entity	12.42	12.81	15.8	15.85	17.84	16.2	19.4	18.43	22.22	20.25	24.35	21.44
		Relation	28.83	31.09	36.57	40.89	38.16	34.59	41.71	39.5	42.06	39.21	48.47	41.4
		Triplet	2.95	2.16	2.56	1.95	3.67	1.99	5.01	2.49	4.9	4.46	7.22	4.99

Table 5: The data augmentation analysis.

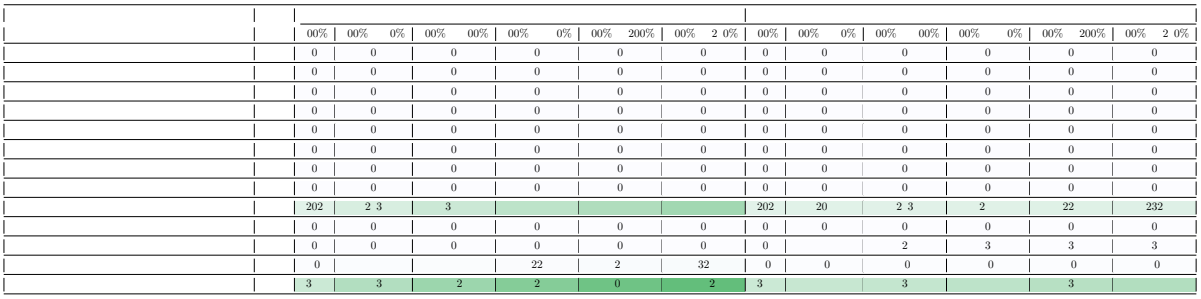

Table 6: The frequency in the training set of some entities which is demonstrated because their frequency in the test set is greater than 3.

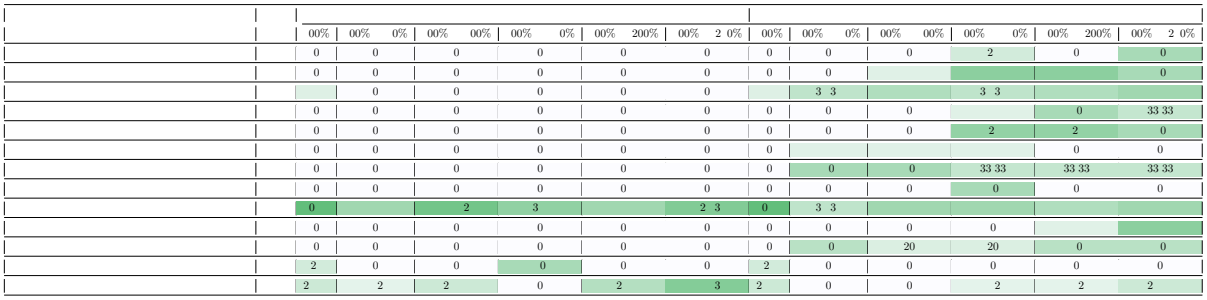

Table 7: The prediction accuracy of some entities which is demonstrated because their frequency in the test set is greater than 3.