Inverse Text Normalization of Air Traffic Control System Command Center Planning Telecon Transcriptions

Kevin Guo*, Stephen S. B. Clarke† and Krishna M. Kalyanam‡
NASA Ames Research Center, Moffett Field, California, 94035

We present a hybrid *neural network and rule-based* Inverse Text Normalization (ITN) method for domains containing unique technical phraseology, specifically ATCSCC planning telecon audio transcriptions. The Air Traffic Control System Command Center (ATCSCC) hosts bihourly planning telephone conferences (or planning telecons) to ensure smooth operations within the National Airspace (NAS). Access to both live and post meeting transcripts of this speech audio would enable quick review of meetings. ITN is the process of converting un-formatted "raw" speech-to-text transcripts into a human (expert) readable written form. Our hybrid ITN framework utilizes a neural network to format conversational English, and rule-based methods to format domain-specific aviation text. With a preliminary overall Word Error Rate with Punctuation and Capitalization (WER PC) of 8.26, we show that this method has vast potential in being applied to ATCSCC planning telecon audio and other audio/text based data available in ATM.

Nomenclature

ASR	=	automatic speech recognition
ATC	=	air traffic control
$ATCSCC$	=	air traffic control system command center
$ATCT$	=	air traffic control tower
ATM	=	air traffic management
CER	=	character error rate
ITN	=	inverse text normalization
$LSTM$	=	long short-term memory
NAS	=	national airspace system
NER	=	named entity recognition
$NFDC$	=	national flight data center
NLP	=	natural language processing
PER	=	punctuation error rate
SME	=	subject matter expert
TMI	=	traffic management initiative
TN	=	text normalization
$TRACON$	=	terminal radar approach control
WER	=	word error rate with no punctuation and no capitalization
$WER\,C$	=	word error rate with capitalization and no punctuation
$WER\,PC$	=	word error rate with punctuation and capitalization

Extended Abstract

This extended abstract does the following: (1) describes the background and motivation for conducting Inverse Text Normalization (ITN) on ATCSCC planning telecon audio transcripts, (2) provides an overview of the data collected, processed and used to train our models, (3) introduces a hybrid neural network and rule-based ITN method and (4)

*Undergraduate Student, University of Southern California, Viterbi School of Engineering
†Senior Aerospace Research Engineer, NASA Ames Research Center, Flight Research Aerospace
‡Senior Aerospace Research Engineer, NASA Ames Research Center, AIAA Associate Fellow

presents preliminary results from the model. In the final paper, we will provide further improvement to our result metrics and include discussion of the significance of these results within the context of Air Traffic Management (ATM).

I. Introduction

Every flight in the United States is managed by the Air Traffic Control System Command Center (ATCSCC). The Command Center maintains constant communication with stakeholders in the National Airspace System (NAS) to address current and future air traffic constraints, events, and delays, as well as how to mitigate the adverse effects of these events to ensure smooth traffic flow*. Traffic Management Initiatives (TMIs) are one of many techniques Air Traffic Control (ATC) managers utilize to prevent backlog in the NAS. For example, TMIs including airborne holding and ground delay programs are used to balance capacity with demand and ensure safe flow of traffic (e.g., under inclement weather). The key to the ATCSCC's ability in maintaining safe flight operations lies in its efficient communication pipeline. The Command Center maintains effective communication with multiple NAS users including Air Route Traffic Control Centers (ARTCCs), Terminal Radar Approach Control (TRACON) facilities, Air Traffic Control Towers (ATCTs), and the aviation industry's many partners and stakeholders. Every day, the Command Center organizes planning telecons, PERTI (Plan, Execute, Review, Train, and Improve) meetings, and pop-up side-bar meetings between different parties. Planning telecons are held every two hour with users to identify upcoming terminal and airspace constraints in the NAS and develop control measures (e.g., TMIs) on how to mitigate them. These planning telecons will be the focus of this paper.

An automated speech recognition (ASR) and natural language processing (NLP) workflow to transcribe, process, and analyze planning telecon audio has the ability to enhance the efficiency of the information sharing/dissemination process. It is clear that air users can manage their flights more efficiently if constraint information is made available to them quickly and without error (e.g., in a digital form). Additionally, air traffic management specialists currently manually review planning telecons (after the fact) for quality assurance. Having access to text transcriptions can enhance this traditionally time consuming process by providing searchable text data and allowing users to quickly identify areas of interest without having to listen to 10-30 minute long audio recordings. Moreover, building a historic data record of text transcriptions enables post-processing data analytics, trend identification, and development of AI/ML based modeling and prediction tools.

Our primary focus in this work is Inverse Text Normalization (ITN), the process of converting unformatted text normalized (TN) speech text inferred by an ASR model into a more (human/expert) readable written form for end-users. By digitizing this process, we ensure both efficient and consistent planning telecon transcriptions. These transcripts can be used by the ATCSCC to view past trends as well as contribute to future modelling work. In order to create formatted transcripts, we require a pipeline that can process both the conversational and aviation-specific language used by air traffic managers in planning telecons. For this, we utilize a neural network and rule-based hybrid ITN framework [1]. In this hybrid framework, raw ASR output data is first processed by a BERT [2] classification model to predict capitalization and punctuation labels for each word in an input sequence. The results from the classification model are then passed into rule-based methods which utilize dictionaries and *regex* [3] (regular expression) search patterns to replace any remaining unformatted tokens into their ITN form. As we will continue to discuss, the neural-network-based punctuation/capitalization method benefits greatly from general-domain pre-trained transformer models like BERT [4], whereas the rule-based methods are easily tunable to cater to the specific formatting required in the ATM domain.

The rest of the paper is organized as follows. Section II provides an overview of the data collection and pre-processing methods. Section III dives deeper into the specific methodology of our hybrid neural network and rule-based ITN framework. Lastly, Section IV discusses the results of our approach and how well it works for domain-specific use-cases such as ours.

II. Data Collection and Pre-processing

The ATC speech dataset used to develop the hybrid ITN model includes 5 hours or $57,478$ words of speech transcripts from ATCSCC planning telecons. This dataset was developed from ATCSCC planning telecon audio collected in-house at NASA Ames Research Center in partnership with the FAA. The audio was first transcribed into TN transcriptions by an in-house ASR model, and then updated to ground truth by subject matter expert (SME) annotators (i.e., retired controllers and air traffic managers). This TN dataset consists only of audio-based text transcriptions. All text is lowercase, containing no punctuation, with numbers and numeric phrases in their expanded written form

*https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/systemops/nas_ops/atcscc

('thirteen fifteen zulu' instead of '1315Z') and acronyms/abbreviations in their expanded written form ('d f w' instead of 'DFW' or 'tracon' instead of 'TRACON'). A second pass was then done to convert the TN dataset into an ITN dataset by adding the above punctuation, capitalization, numeric formatting, acronyms and abbreviations to the text. See Table 1 for examples of the two different formats. After completing the required transcriptions, the TN and ITN text dataset was split 80/20 into training and testing datasets.

TN: seventeen hundred zulu **ITN**: 1700Z
TN: seattle g d ps **ITN**: Seattle GDPs
TN: toronto and minneapolis a f ps **ITN**: Toronto and Minneapolis AFP's
TN: v f r to i f r **ITN**: VFR to IFR
TN: o i s page philadelphia **ITN**: OIS page Philadelphia

Table 1 Example comparisons between TN and ITN

Additional data used for the rule based system includes lists of airport names from the National Flight Data Center (NFDC)[†], lists of cities and states from the airport dataset, and other miscellaneous lists such as TRACONs, ARTCCs and common FAA acronyms pertaining to the planning telecon content. These miscellaneous lists were developed by SMEs while reviewing the training dataset. See Table 2 for examples of each category.

Dictionary Name	**Examples**
Cities	Aberdeen, Abilene, Adak Island, Aguadilla, Akron, ...
Aerodromes	Chicago O'hare, Los Angeles, Dallas-Fort Worth, Denver, ...
ARTCCs	Albuquerque, Anchorage, Atlanta, ..., ZAB, ZAN, ...
TRACONs	A11, A80, A90, ..., Anchorage, Atlanta, ...
Common Carriers	American, Delta, JetBlue, FedEx, UPS, United, ...
Abbreviations	FAA, OIS, TCF, TAF, DSP, ...

Table 2 Dictionaries collected and used for rule-based ITN.

For training the neural-network-based capitalization and punctuation method, a script was used to compare our TN dataset (input) and our ITN dataset (output) to automatically generate labels for each token. These token labels and token counts are displayed in Table 3.

[†]https://www.faa.gov/air_traffic/flight_info/aeronav/aero_data/

Label Name	Label	Count
Comma Lowercase	,O	1,300
Comma Uppercase	,U	1,414
Period Lowercase	.O	3,477
Period Uppercase	.U	825
Question Lowercase	?O	524
Question Uppercase	?U	178
Lowercase	OO	42,980
Uppercase	OU	6,780
Total		57,478

Table 3 Data label definitions, labels, and counts for neural-network-based capitalization and punctuation.

To illustrate the tokenization scheme, we use the example sentence, *Alright, any other FAA en route facilities with items to bring forward?*. The corresponding TN text, ITN text, and token for every word in the sentence are provided in table 4.

TN text	ITN text	token
alright	Alright,	,U
any	any	OO
other	other	OO
f a a	FAA	UU
en	en	OO
route	route	OO
facilities	facilities	OO
with	with	OO
items	items	OO
to	to	OO
bring	bring	OO
forward	forward?	O?

Table 4 Tokenization and labels for example sentence

III. ITN Methodology

Recall that ITN is the process of converting text from its spoken form to its natural written form. In general ITN is used to convert numbers and time (of the day) from alphabetic form to numeric form and it is commonly used as a post-processing step to ASR systems [1] [4] [5]. However, in our aviation domain data, we must include specific formatting such as zulu times, runways, runway configurations, altitudes, speeds, and the many acronyms and abbreviations used throughout the planning telecon.

There are many approaches to the ITN task including rule-based, probabilistic methods, and neural networks [6]. Existing ITN models that make use of neural networks are typically trained on large datasets of conversational English such as the Long Short-Term Memeory (LSTM) model, trained on a 1.1B word Wikipedia dataset [7], or the pretrained BART [8] model, fine-tuned on the 313K words Must-C dataset [1] [9]. However, training these large neural networks requires extensive amounts of training data, a difficult task to achieve when using small domain-specific datasets. Our work utilizes a hybrid approach [1] that combines neural-network-based and rule-based methods applied to our aviation-specific dataset. We find that this approach bridges the gap between the conversational English and technical aviation-specific phraseology found within the planning telecons.

A. Neural Network Based ITN

Punctuation and Capitalization, a subset of ITN, is a common task within NLP that is often applied to the output of ASR systems to improve readability [6]. This task has been previously addressed with many methods such as rule-based, n-gram-based, probabilistic models, and neural networks [6]. With the recent rise of transformer-based neural networks, one approach is to train a token-classifier on top of a pretrained transformer model like BERT [4]. Other neural network approaches include using an Evolved Transformer with Chunk Merging [10], Character-Level Recurrent Neural Networks [11] and End-To-End Conformer Language Models [12]. Additionally, both capitalization and punctuation can be combined into a single step for a neural network [10].

For our neural network model, we combine both punctuation and capitalization to restore periods (.), commas (,), question marks (?), and (upper or lower) casing as is appropriate. A more detailed view of our classes was seen in Table 3. A training pipeline was developed using the Huggingface library[‡]. For these initial results, we started with the *bert-base-uncased* [§] transformer model. Without any modification of the base model we added and trained a linear layer on top of the BERT output layer for classification, also known as a classification head[¶], to predict each token as described above. The process for converting TN data to ITN data is as follows:

1) Split the TN sentence into a list of tokens.
2) Classify each token with a label found in Table 3 using our bert-base-uncased model.
3) Format each token with its respective designation and recreate the sentence with ITN formatting.

Although there are large benefits to be had with neural-network-based punctuation and capitalization, it still lacks the tools needed to address non-conversational English. Next, we will discuss how we complement the neural network process by utilizing rule-based algorithms to format domain-specific acronyms and numeric phrases.

B. Rule-Based ITN

In prior ITN research, rule-based methods have been used Finite State Transducers [1] or rewrite tables [13]. Rule based systems are also used for other NLP tasks such as Named Entity Recognition (NER) [14], and NER in the aviation domain [15].

After our input TN data is processed by the neural network, we utilize rule-based methods to locate and normalize any remaining domain-specific text that requires formatting. Using only the neural network to normalize character patterns of domain-specific locations and acronyms revealed inconsistencies where acronyms were either not recognized, or only capitalized as a proper noun. To combat this, our rule-based system uses regex[‖] search strings. Regex is a widely used format to search for specific strings within bodies of text [3]. For our work, regex search strings have been created for runways, numbers, and time (of the day). These regex search strings are also composed of capture groups which condense general searches such as single DIGIT (one, two, etc.), TEENS (eleven, twelve, etc.,) and TIES (twenty, thirty, etc.,). When the regex search string is recognized within the TN data, the selected character pattern is passed into its associated modification recipe to be modified and replaced. Table 5 shows an example of how this process is used. The input string "twenty two z" is first recognized by the regex search string since it starts with a TIES word (twenty), then followed by a DIGIT word (two) and ends with *z* or *zulu* (in this case, z). The modification recipe then removes the spaces between the words and capitalizes the *Z*.

Name: Ties Digit Zulu
Regex Search String: `(?P<TIES_00>)\s+(?P<DIGIT_00>)\s+(z|zulu)`
Modification Recipe: `(?P<TIES_00>)(?P<DIGIT_00>)Z`
Captured String: twenty two z
Replacement: 22Z

Table 5 Capture Group Recipe Replacement

After regex replacement, we iterate through each dictionary dataset in Table 2 replacing the remaining unformatted tokens such as locations and domain-specific acronyms with its corresponding formatted token.

[‡] https://huggingface.co/
[§] https://huggingface.co/bert-base-uncased
[¶] https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertForSequenceClassification
[‖] https://docs.python.org/3/library/re.html

IV. Preliminary Results

Our validation consisted of running our TN test data through our hybrid model, measuring results at the intermediate (neural-network-based) step, and calculating end-to-end metrics of the entire framework. For the neural network, we report accuracy, precision, recall, and F1-scores for each label. After running the neural network output through the rule-based system, we calculate the punctuation error rate (PER), word error rate with no punctuation and no capitalization (WER), word error rate with capitalization and no punctuation (WER C), word error rate with punctuation and capitalization (WER PC), and character error rate (CER). These metrics have been reported to provide a standardized and accurate way of evaluating ITN for end-to-end ASR models [12].

A. Neural Network Capitalization and Punctuation Results

Figure 1 shows the classification accuracy of normalized text in predicted and truth data as a confusion matrix. We can see that our BERT-based model's classification per-label accuracy is consistent with existing transformer-based methods [10]. Even so, there appears to be noticeable confusion between some classes. The first notable example is confusion between ,O, .O and OO. This makes sense since different annotators may have different writing styles, leading to variable use of commas. There is also confusion with the uppercase variants ,U, .U, and OU. Another point of confusion is between OO and OU. Similar inaccuracies have been reported in prior related work [10], and is to be expected since proper nouns often overlap with nouns that don't require capitalization.

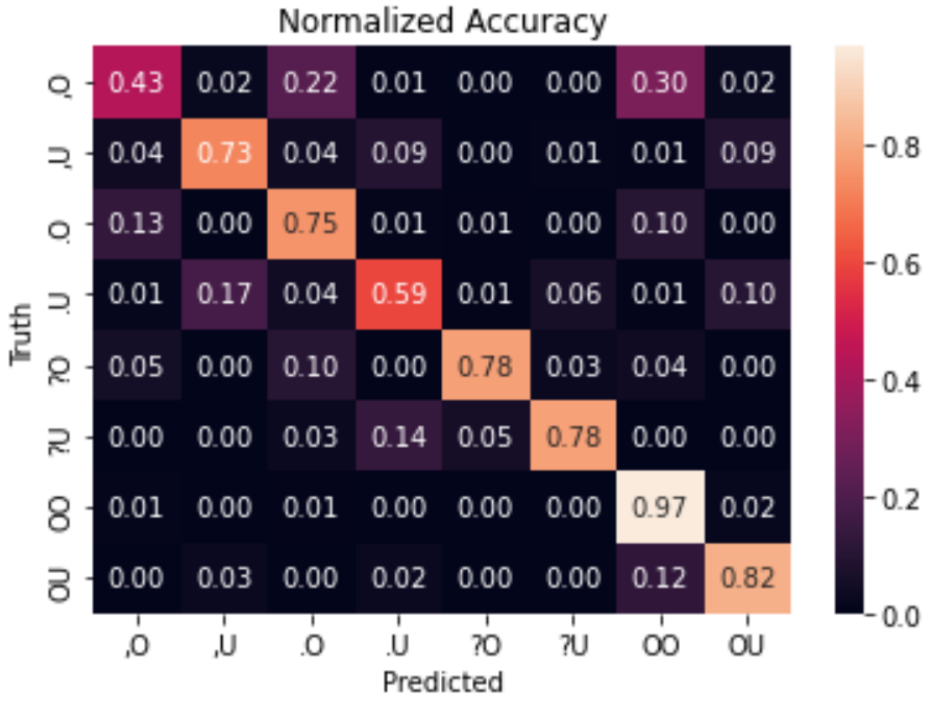

Fig. 1 Neural Network Capitalization and Punctuation Confusion Matrix

Overall, the trained model achieves 0.72 micro average F1-Score as seen in Table 6. This result is comparable to the results produced in existing general-domain punctuation methods [6, 16]. We also achieve an overall accuracy of 0.91, which is significant. We believe that this number shows good performance of our model with the caveat that there is significant imbalance in our data with the OO label representing nearly 75% of the dataset. In future work the model performance can be improved by both adding data and ensuring that the annotation of specific punctuations like *commas* is consistent across the dataset. We will also experiment with splitting punctuation and capitalization as separate tasks to see if a unified model may be too complex for the problem.

Label	Precision	Recall	F1-Score
,O	0.25	0.43	0.32
,U	0.76	0.73	0.72
.O	0.83	0.75	0.79
.U	0.53	0.59	0.56
?O	0.88	0.78	0.83
?U	0.67	0.78	0.72
OO	0.96	0.97	0.97
OU	0.86	0.82	0.84
Accuracy			0.91
Micro Avg	0.72	0.73	0.72
Macro Avg	0.92	0.91	0.92

Table 6 Classification report of neural network capitalization and punctuation model testing

B. End-to-end Results

When evaluated on the test datset, our end-to-end hybrid model achieves a PER of **42.98%**, WER of **1.15%**, and WER PC of **8.26%**, as seen in Table 7. Since there is no comparison or baseline to aviation domain sources, we compare these results to general-domain sources. The results achieved on the general-domain LibriSpeech dataset are 29.27% PER, 2.22% WER, and 7.66% WER PC using an End-To-End Conformer Language Model [12]. From these results, we can see that our WER and WER PC are at similar levels to the current state-of-the-art in general domain, but our PER is higher. In our final paper, we hope to improve this PER by training and evaluating additional neural network models such as the End-To-End Conformer Language Model [12] and an Evolved Transformer with Chunk Merging [10].

Method	PER (%)	WER (%)	WER C (%)	WER PC (%)	CER (%)
bert-base-uncased + rule-based	42.98	1.15	6.71	8.26	3.13

Table 7 End-to-end evaluation metrics

Table 8 shows a selection of comparisons between our hybrid model's predictions and the ground truth. We see that numbers and time (of the day), locations and aviation acronyms are consistently predicted correctly while punctuation is sometimes missed or is incorrect. We also see room for improvement for the rule-based algorithm such as in the second example where '*3 north*' was incorrectly predicted as '*3N.*' This is as simple as adding an additional regex search string and modification recipe for digits followed by a cardinal direction. An interesting example of punctuation error appears in the third example with the truth phrase " *southeast corner also. But like I said, I think*" versus the ITN prediction "*southeast corner also, but like I said, I think*". One could reasonably argue that both are correct and replacing the period followed by a "B" with a comma followed by "b" does not change the meaning or intent of the speaker. For this reason, we can accept this output even though it is recorded as an error in punctuation.

TN: no not much more to add weather still clear the bridge a few at the airport expecting a fifty four rate at fourteen z and we'll just see how the weather goes from there **ITN Truth**: No, not much more to add. Weather still clear the bridge. A few at the airport. Expecting a 54 rate at 14Z and we'll just see how the weather goes from there. **ITN Pred**: No not much more to add. Weather. Still clear the bridge a few at the airport, expecting a 54 rate at 14Z and we'll just see how the weather goes from there.
TN: good evening from the tracon we're planning on three north operation with a ninety two arrival rate all the gates will be open we're looking for a smooth end of the week **ITN Truth**: Good evening from the TRACON. We're planning on 3N operation with a 92 arrival rate. All the gates will be open. We're looking for a smooth end of the week. **ITN Pred**: Good evening from the TRACON we're planning on 3 north operation with a 92 arrival rate. All the gates will be open. We're looking for a smooth end of the week.
TN: and umm just looking at our solid bunch in our southeast corner also but like i said i think we're just gonna remain tactical through this first bank **ITN Truth**: And umm, just looking at our solid bunch in our southeast corner also. But like I said, I think we're just gonna remain tactical through this first bank **ITN Pred**: And umm, just looking at our solid bunch in our southeast corner also, but like I said, I think we're just gonna remain tactical through this first bank.

Table 8 End-to-end Examples

Overall, the preliminary results are encouraging. For our final paper, we plan to experiment with additional models and data cleaning techniques to further lower the overall PER while maintaining the current low WER and WER PC numbers. Additionally, we will continue to work with ATM SMEs to ensure the usage of this work is accurate and helpful.

V. Acknowledgements

We are grateful for the support and guidance provided by subject matter experts and other stakeholders at the FAA Office of NextGen.

References

[1] Sunkara, M., Shivade, C., Bodapati, S., and Kirchhoff, K., "Neural inverse text normalization," *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 7573–7577.

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

[3] Goyvaerts, J., "Regular Expressions," *Regular Expression*, 2006.

[4] Nagy, A., Bial, B., and Ács, J., "Automatic punctuation restoration with bert models," *arXiv preprint arXiv:2101.07343*, 2021.

[5] Gaur, Y., Kibre, N., Xue, J., Shu, K., Wang, Y., Alphanso, I., Li, J., and Gong, Y., "Streaming, Fast and Accurate on-Device Inverse Text Normalization for Automatic Speech Recognition," *2022 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2023, pp. 237–244.

[6] Păiş, V., and Tufiş, D., "Capitalization and punctuation restoration: a survey," *Artificial Intelligence Review*, 2021, pp. 1–42.

[7] Sproat, R., and Jaitly, N., "RNN approaches to text normalization: A challenge," *arXiv preprint arXiv:1611.00068*, 2016.

[8] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L., "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[9] Di Gangi, M. A., Cattoni, R., Bentivogli, L., Negri, M., and Turchi, M., "MuST-C: a Multilingual Speech Translation Corpus," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, edited by J. Burstein, C. Doran, and T. Solorio, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 2012–2017. https://doi.org/10.18653/v1/N19-1202, URL https://aclanthology.org/N19-1202.

[10] Nguyen, B., Nguyen, V. B. H., Nguyen, H., Phuong, P. N., Nguyen, T.-L., Do, Q. T., and Mai, L. C., "Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging," *2019 22nd conference of the oriental COCOSDA international committee for the co-ordination and standardisation of speech databases and assessment techniques (O-COCOSDA)*, IEEE, 2019, pp. 1–5.

[11] Susanto, R. H., Chieu, H. L., and Lu, W., "Learning to Capitalize with Character-Level Recurrent Neural Networks: An Empirical Study," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, edited by J. Su, K. Duh, and X. Carreras, Association for Computational Linguistics, Austin, Texas, 2016, pp. 2090–2095. https://doi.org/10.18653/v1/D16-1225, URL https://aclanthology.org/D16-1225.

[12] Meister, A., Novikov, M., Karpov, N., Bakhturina, E., Lavrukhin, V., and Ginsburg, B., "LibriSpeech-PC: Benchmark for Evaluation of Punctuation and Capitalization Capabilities of end-to-end ASR Models," *arXiv preprint arXiv:2310.02943*, 2023.

[13] Pusateri, E., Ambati, B. R., Brooks, E., Platek, O., McAllaster, D., and Nagesha, V., "A Mostly Data-Driven Approach to Inverse Text Normalization." *INTERSPEECH*, Stockholm, 2017, pp. 2784–2788.

[14] Sekine, S., and Nobata, C., "Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy." *LREC*, Lisbon, Portugal, 2004, pp. 1977–1980.

[15] Clarke, S. S., Zhu, Z., He, O., Almeida, J. A. A., Kalyanam, K., and Pai, R., "Natural Language Understanding and Extraction of Flight Constraints Recorded in Letters of Agreement," *AIAA Aviation Forum*, 2022.

[16] Yi, J., and Tao, J., "Self-attention Based Model for Punctuation Prediction Using Word and Speech Embeddings," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7270–7274. https://doi.org/10.1109/ICASSP.2019.8682260.