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

Kevin H. 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 Air Traffic Control System Command Center (ATCSCC) planning telecon audio transcriptions. The ATCSCC hosts bi-hourly 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. Provided speech transcripts, ITN is the process of converting unformatted "raw" Automated Speaker Recognition (ASR) model transcripts into a human (expert) readable written form. Our hybrid ITN framework utilizes a fine-tuned *Bidirectional Encoder Representations from Transformers* neural network to format conversational English, and rule-based methods to format domain-specific aviation text. With an overall Punctuation Error Rate (PER) of 25.56 and Word Error Rate with Punctuation and Capitalization (WER PC) of 5.47, 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

I. Introduction

Every flight in the United States is affected by operations at the Air Traffic Control System Command Center (ATCSCC). The Command Center maintains constant communication with stakeholders in the National Airspace

*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

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 communications pipeline. The Command Center maintains effective communication with multiple NAS users: 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, Plan, Execute, Review, Train, and Improve (PERTI) meetings, and pop-up side-bar meetings between different parties. These planning telecons are hosted bi-hourly to identify and discuss upcoming terminal and airspace constraints in the NAS and develop control measures (e.g., TMIs) on how to mitigate them. This paper will focus on these planning telecons.

An automated speech recognition (ASR) and natural language processing (NLP) workflow to transcribe, process, and analyze planning telecon audio can enhance the efficiency of the information sharing/dissemination process. In the current quality control process, air traffic management specialists manually listen through and review planning telecons. Access to text transcriptions has the potential to optimize this traditionally time consuming process by providing text-searchable data, bypassing the time-consuming step of listening to 10-30 minute long audio recordings. Searching a text document for the mention of a specific airport, say 'DFW,' could take seconds using standard *find* tools in modern text processors. 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 then 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. To address this task, we apply a hybrid neural network and rule-based ITN framework [1]. In this hybrid framework, raw ASR output data is first processed by a 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* [2] (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 pretrained transformer models like Bidirectional Encoder Representations from Transformers (BERT) [3] and DistilBERT [4], whereas the rule-based methods can easily be fine-tuned 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 preprocessing 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 our domain-specific use-case.

II. Data Collection and Preprocessing

The ATC speech dataset used to develop the hybrid ITN model includes 5 hours, $1,249$ utterances, 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 ('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'). We then created the ITN dataset by adding the above punctuation, capitalization, numeric formatting, acronyms and abbreviations to the TN data. 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.

Additional data used for the rule based system includes lists of airport names from the National Flight Data Center

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

(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.

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

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.

When 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. We have labels for both lowercase and uppercase (capitalization of the first letter in a word), and three punctuation marks: periods, commas, and question marks. Note that there is no class for fully uppercase words since acronyms are either separated by letter (e.g., 'f a a' as seen in Table 4) or later handled by the rule-based method (e.g., 'tracon').

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

Label Name	Label	Count
Comma Lowercase	lower,	1,300
Comma Uppercase	Upper,	1,414
Period Lowercase	lower.	3,477
Period Uppercase	Upper.	825
Question Lowercase	lower?	524
Question Uppercase	Upper?	178
Lowercase	lower_	42,980
Uppercase	Upper_	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,	Upper,
any	any	lower_
other	other	lower_
f	F	Upper_
a	A	Upper_
a	A	Upper_
en	en	lower_
route	route	lower_
facilities	facilities	lower_
with	with	lower_
items	items	lower_
to	to	lower_
bring	bring	lower_
forward	forward?	lower?

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] [3] [5]. In the case of domain-specific aviation data however, we must address the specific formatting of zulu times, runways, runway configurations, altitudes, speeds, and the many acronyms and abbreviations used throughout the planning telecon.

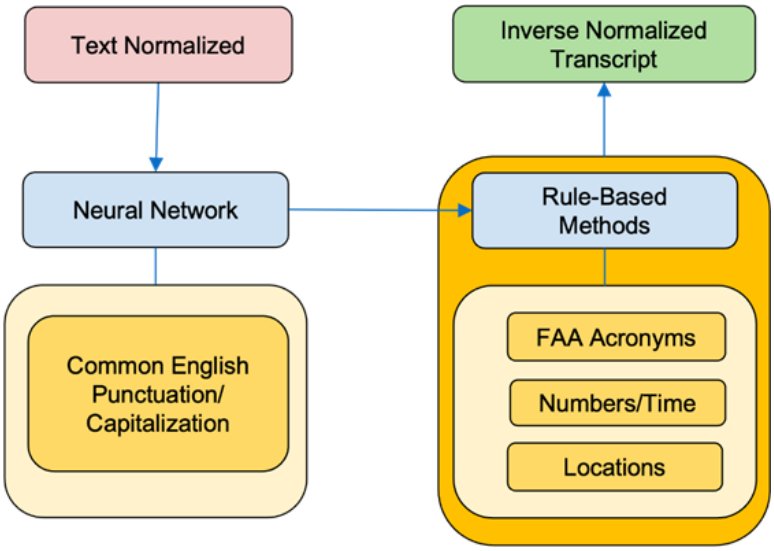

Fig. 1 Hybrid Neural and Rule-Based ITN Methodology

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 Memory (LSTM) model, trained on a 1.1 billion words Wikipedia dataset [7], or the pretrained BART [8] model, fine-tuned on the 313 thousand words Must-C dataset [1] [9]. Training these large neural networks however, requires extensive amounts of training data, a difficult task to address when using small domain-specific datasets. Our approach 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. Figure 1 shows how the input TN text is first piped through a neural network to inverse normalize the majority of common English punctuation and capitalization. That output is then fed into rule-based methods which are created to format most of the domain-specific acronyms and abbreviations that are used at the Command Center.

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 [3]. 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].

Our neural network model combines both punctuation and capitalization to restore periods (.), commas (,), question marks (?), and (upper or lower) casing as is appropriate. A detailed view of our classes is shown in Table 3. A training pipeline was developed using the Hugging Face library[‡]. Our first configuration utilizes BERT. 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. In our second configuration, we use a fine-tuned punctuation and capitalization model *DistilBERT-base re-punctuate* [¶] transformer model which has already learned some punctuation and capitalization labels. For the third configuration, we further fine-tune the *DistilBERT-base re-punctuate* with our data to provide for domain-specific aviation context. The process for applying the neural network to TN data is as follows:

1) Split the TN sentence into a list of tokens.

[‡]https://huggingface.co/

[§]https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertForSequenceClassification

[¶]https://huggingface.co/unikei/distilbert-base-re-punctuate

2) Classify each token with a label found in Table 3 using our punctuation and capitalization 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 and aviation-specific acronyms and abbreviations. Next, we will discuss how we complement the neural network process by utilizing rule-based algorithms to address this.

B. Rule-Based ITN

In prior ITN research, rule-based methods have been implemented by 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 input TN data is processed by the neural network, rule-based methods 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 capitalized only as a proper noun. To combat this, our rule-based system uses regex[‖] search strings. Regex is a widely used format to parse text for specific strings [2]. We created regex search strings for runways, routes, numbers, and times. These regex search strings are composed of capture groups condensing general searches for: single DIGIT (one, two, etc.), TEENS (eleven, twelve, etc.), and TIES (twenty, thirty, etc.). When a target search string is identified 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 capture group recipe used in this process. The input string "twenty two z" is first recognized by the regex search string as it starts with a TIES word (twenty), and follows with a DIGIT word (two) before ending 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. Some examples of these are airport designators, TRACON designators, ARTCC designators, common airspace fix names, airline names, and other FAA acronyms like TMIs.

IV. Results

Our validation consisted of running our TN test data through our hybrid model utilizing three neural network progressions. We measure results of the end-to-end hybrid model with the following three neural networks: (1) BERT-base uncased model with fine-tuning *(BERT-base-uncased + FT)*, (2) DistilBERT-base re-punctuate model without fine-tuning *(DistilBERT-base re-punctuate)*, and (3) DistilBERT-base re-punctuate model with fine-tuning *(DistilBERT-base re-punctuate + FT)*. We measure results at the intermediate (neural-network-based) step and calculate end-to-end metrics of the entire framework after applying the rule-based methods. To measure neural network performance, we report the accuracy, precision, recall, and F1-scores of 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 end-to-end ITN for speech data [12].

[‖] `https://docs.python.org/3/library/re.html`

A. Neural Network Capitalization and Punctuation Results

Figure 2 shows the classification accuracy of normalized text in predicted and truth data as a confusion matrix for each model configuration. We see that our *BERT-base-uncased + FT* 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 *lower,* (lowercase with a comma), *lower.* (lowercase with a period) and *lower_* (lowercase without punctuation). 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 *Upper,*, *Upper.*, and *Upper_*. Another point of confusion is between *lower_* and *Upper_*. Similar inaccuracies have been reported in prior related work [10], and is to be expected as proper nouns often overlap with non-capitalized nouns. *The DistilBERT-base re-punctuate + FT* model shows similar performance across all labels, but the *The DistilBERT-base re-punctuate* model appears to over-predict punctuation and has confusion between labels.

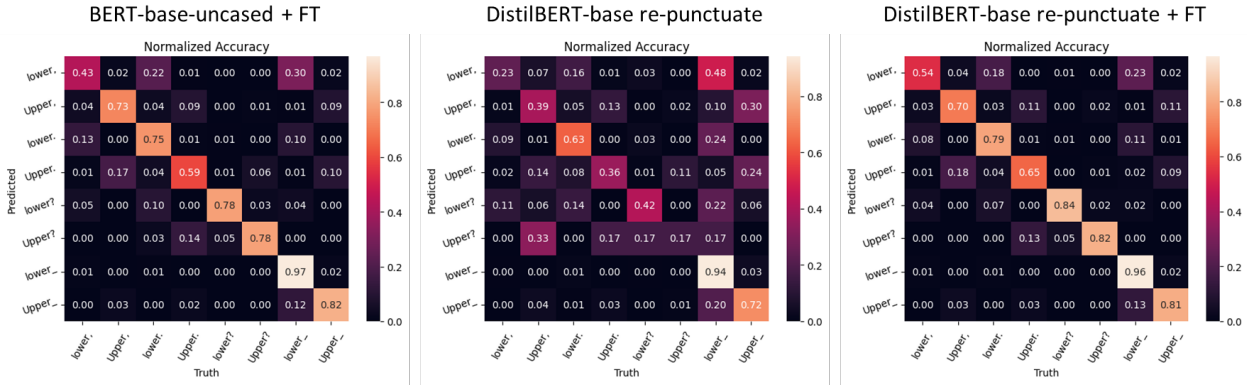

Fig. 2 Neural Network Capitalization and Punctuation Confusion Matrix

Overall, the *BERT-base-uncased + FT* and *DistilBERT-base re-punctuate + FT* models perform well by achieving 0.91 weighted average F1-Score respectively as seen in Table 6. We report the weighted average F1-Score because it is more representative of the imbalanced label dataset, with *lower_* being the majority class. This result is comparable to the results produced in existing general-domain punctuation methods [6, 16]. We believe that these results shows good performance of our model with the caveat that there is significant imbalance in our data with the *lower_* label representing nearly 75% of the dataset. While this imbalance exists because of the nature of capitalization and punctuation in the English language, future works can still look to improve model performance by adding additional data and ensuring that the annotation of specific punctuation like *commas* are consistent across the entire dataset.

	BERT-base-uncased + FT			**DistilBERT-base re-punctuate**			**DistilBERT-base re-punctuate + FT**		
Label	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
lower,	0.43	0.25	0.32	0.23	0.55	0.33	0.54	0.44	0.48
Upper,	0.73	0.76	0.74	0.39	0.62	0.48	0.70	0.74	0.72
lower.	0.75	0.83	0.79	0.63	0.64	0.63	0.79	0.81	0.80
Upper.	0.59	0.53	0.56	0.36	0.29	0.32	0.65	0.47	0.55
lower?	0.78	0.88	0.83	0.42	0.17	0.24	0.84	0.84	0.84
Upper?	0.78	0.67	0.72	0.17	0.02	0.04	0.82	0.72	0.77
lower_	0.97	0.96	0.97	0.94	0.91	0.92	0.96	0.96	0.96
Upper_	0.82	0.86	0.84	0.72	0.65	0.68	0.81	0.84	0.82
Accuracy			0.91			0.83			0.91
Macro	0.73	0.72	0.72	0.48	0.48	0.46	0.76	0.73	0.74
Weighted	0.91	0.91	0.91	0.85	0.83	0.83	0.91	0.91	0.91

Table 6 Classification report of BERT-base-uncased + FT capitalization and punctuation model testing

B. End-to-end Results

When evaluated on the test dataset, the best performing end-to-end hybrid model with both fine-tuning (FT) and rule-based (RB) is *BERT-base-uncased + FT + RB*. This model achieves a PER of **25.56%**, WER of **0.68%**, and WER PC of **5.47%** 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]. We observe that our results measure similarly when compared to current state-of-the-art models in the general domain.

Method	PER (%)	WER (%)	WER C (%)	WER PC (%)	CER (%)
BERT-base-uncased + FT + RB	25.56	0.68	4.53	5.47	2.00
DistilBERT-base re-punctuate + RB	43.24	1.58	8.05	12.53	4.04
DistilBERT-base re-punctuate + FT + RB	26.05	0.95	4.70	5.61	2.03

Table 7 End-to-end evaluation metrics

As for the *DistilBERT-base re-punctuate* models, we can see that without fine-tuning the model achieves a 43.24% PER. While this number is almost double the best PER, this model still performs well when viewing specific examples. We believe this drop in performance is a result of being trained on out-of-domain data. By further fine-tuning the *DistilBERT-base re-punctuate* model, we achieve comparable results to the *BERT-base-uncased + FT* model. Although the addition of fine-tuning provides results comparable to our the *BERT-base-uncased + FT* model, we expected to see slight a slight improvement given overall more data was included in the *DistilBERT-base re-punctuate* model training. Future works should investigate training and testing additional off-the-shelf models with the additional fine-tuning task for comparison.

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 incorrectly predicted or missed entirely. The rule-based algorithm could be improved in examples such as where '*3 north*' is 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 "*...evening from the TRACON. We're planning on...*" versus the ITN prediction "*...evening from the TRACON, We're planning on...*". Both of these sentences have the same meaning and speaker intent, but a period was replaced with a comma. Although many end users may term both correct, our evaluation still considers this an error. Given that these misalignments in punctuation occur often in English, we deduce that some minimum level of PER will always exist.

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 Prediction**: 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 Prediction**: 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 Prediction**: 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

These results show promise for enhancing the ATC quality assurance workflow. Future works should examine additional models and data cleaning techniques to further lower the overall PER while maintaining the current low WER and WER PC numbers. We aim to extend the current training and testing datasets to larger planning telecon datasets since this work only encompasses data from 2018. Work should also extend to other subsets of aviation speech in addition to ATC management. While this model performs well on planning telecons, it must be tested on other sub-domains (i.e., air traffic control and other air traffic management speech data) before it can be used. Additionally, we will continue to work with ATM SMEs to ensure the usage of this work is accurate and industry-relevant.

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

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

[4] Sanh, V., Debut, L., Chaumond, J., and Wolf, T., "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *ArXiv*, Vol. abs/1910.01108, 2019.

[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.