International Journal of Computer Engineering in Research Trends
ISSN: 2349-7084/ https://doi.org/10.22362/ijcert.v10i12.914
Volume 10, Issue 12, December 2023

Research Paper

Minimizing Errors in Air Traffic Speech Using Rule Based Algorithms

Narayanan Srinivasan[1*], Balasundaram S R[2]

[1*]*Computer Applications, National Institute of Technology, Tiruchirappalli, India*
[2] *Computer Applications, National Institute of Technology, Tiruchirappalli, India*

e-mail: nbasandroid@gmail.com, srb.nitt@gmail.com

**Corresponding Author: nbasandroid@gmail.com*

Received: 27/10/2023, Revised: 17/11/2023, Accepted: 18/12/2023, Published: 30/12/2023

Abstract:- *The application of automatic speech recognition for air traffic speech has attracted a lot of attention in recent times. The application of machine learning and deep learning models can address the challenges of speech recognition to some extent. This work focuses on understanding the intricacies of domain characteristics and how to leverage them to improve speech recognition. This work focuses on applying rule-based techniques in the post processing phase of speech recognition. The post-processing stage is defined as a stage where a set of algorithms are applied to the decoded output of speech recognition. Multiple techniques, like syntax separation techniques, co-occurrence-based clustering techniques, and string distance algorithms, are discussed in this work. The choice of these algorithms is based on an understanding of the significance of air traffic speech domain characteristics. Machine learning and deep learning models were applied in feature selection, language model generation, or acoustic model generation. Approaches based on rules could be selectively applied as an incremental update to language models or acoustic model generation. This work was able to improve the accuracy by 9% by applying selective algorithms for error detection and correction. Comparisons of different techniques were discussed when multiple techniques were used for clustering and string distance calculations. One of the good observations of this work is that leveraging the characteristics of speech in this domain helped improve accuracy. The improvement in accuracy was seen in two scenarios. One scenario has a complete utterance, and the other scenario has syntax-separated utterances. A rule-based algorithm was proposed for syntax separation.*

Keywords: *Air Traffic Speech, Rule based approaches, Automatic Speech Recognition, String Clusters, Syntax Algorithms*

---

1. Introduction

Speech is the easiest and natural means of communication. Even though it is natural and easy, there are lot of local factors that influence how the speech is rendered. Second, the person listening to the speech interprets it in his own way due to factors influencing his or her interpretation. Nevertheless, speech is the easiest, spontaneous way to express oneself. In Air traffic, speech communication plays a vital role in the safe operations of aircraft. The personas who use speech communication in air traffic are the controllers in airport and the crew in cockpit. Till date, despite text-based data link systems, speech is the fastest way to direct the flights to land or takeoff and move around in the airport.Figure 1 describes the building blocks of air traffic speech recognition. Earlier works are categorized into following,

1. Rule-based techniques that are applied at the speech engine level by suggesting improvements to language model, acoustic model, or lexical model.
2. Rule-based techniques that are applied to external systems to narrow down the search space to find the closest acoustic or n-gram or lexical match.
3. Rule-based techniques that are applied to extract entities or parts of speech in the utterances.

The proposed work considers the following building blocks to differentiate from the earlier work,

1. A standard syntax segregation algorithm leveraging the syntax rules already in use in the domain.
2. A clustering algorithm that can detect errors without depending on any external systems for context.
3. Choosing an appropriate string distance algorithm to correct errors in the utterances.

Cordero J et al.[1] proposed a system that intentionally operated independently of the ATC system data and flight plan details. The primary goal of this work was to create a highly effective ATC speech recognition system without relying on contextual information. Hence the need for an independent system which does not integrate with other ATC systems is required. While Oualil Y et al.[2], Nguyen V et al.[3], and Srinivasamurthy A4 applied language model rules to the speech recognition pipeline. Cordero J et al. [1] applied it at post processing to determine the call sign and command part of the utterance. Grammar based rules were applied to determine the language spoken by Pardo J et al.[5]. In summary, rule-based algorithms are still valid in ATSR and cannot be completely discarded. From these works it is observed that machine learning algorithms when combined with rule-based algorithms can further reduce the errors in ATSR.

In this work a novel rule-based extraction algorithm is proposed leveraging the syntactical strength of the utterances. Badrinath S et al.[6] highlighted less or no focus on applying rule- based approaches in Call Sign extraction. Call Sign can be spoken in many forms and hence it can appear in separate ways in the decoded text from speech engine. The work overcame this with the error correction technique which requires the minimally extracted call sign from the utterances. This work strengthens Cordero J et al.[7] work by a clustering algorithm which can generate a local context and followed by a string distance algorithm to find the closest match for the whole sign or the rest of call sign part.

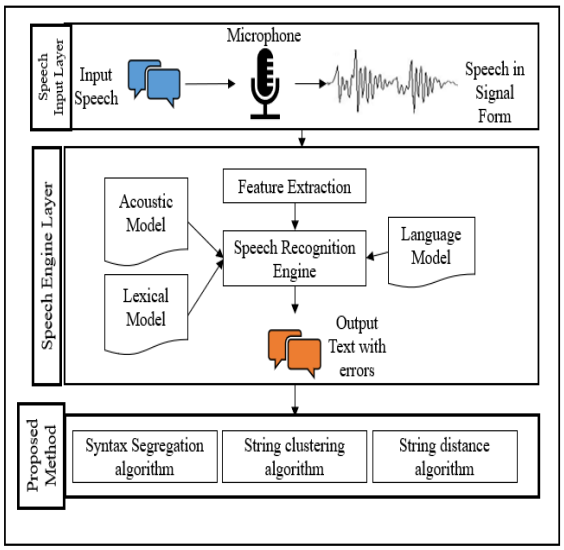


Figure 1– Architecture for Rule-based techniques

1.1. Utterance segregation

The International Civil Aviation Organization (ICAO) has defined a phraseology that includes command, phrases, dictionary that should be used for communication. This corresponds to the set of commands that are used for taxiway, runway, approach area, takeoff, and landing. Consider,

"LUFTHANSA SEVEN SEVEN CONTACT RHEIN ONE THREE SIX." …… (U1)

"AEROFLOT ONE TWO TWO HELLO RADAR CONTACT PROCEED TO PEMUR CLIMB TO FLIGHT LEVEL THREE ONE ZERO" ……. (U2)

Due to the nature of air traffic controller workload, it is natural for the controllers to issue multiple commands in one utterance as seen in (U2). Also, there is a non-standard word "HELLO" used in between. In another situation, the command would be,

"AIR BERLIN THREE SIX ZERO FIVE DESCEND FLIGHT LEVEL TWO NINER ZERO RATE ONE THOUSAND FIVE HUNDRED FEET PER MINUTE OR GREATER ……..(U3)

The examples are for multiple purposes, but all instructions must fit into the general syntax of CALLSIGN + COMMAND + COMMAND INFO.

Syntax Separation Algorithm

The syntax separation algorithm divides the problem into two parts. This allows us to build two algorithms or models which can independently predict the callsign and the rest of the call sign or any part of the rest of the call sign. It is easier to train a language model only for predicting call sign and rest of call sign. It is still challenging to build a generic language model to cover all scenarios. This separation cannot be applied to the rest of the sentence efficiently. Because the third part, which is the information about the command, can be of varying length based on the command itself. To start with, a rule-based algorithm is suggested to separate the call sign from the instruction as described by Figure 2.

The algorithm uses one predefined list derived from standard phraseology,

```
List1 =
('CONTACT','RHEIN','DESCEND','CLIMB','TURN',
'SET','PROCEED',
        DIRECT','GOOD', 'MILAN',
'CONFIRM','RADAR','IS',
'CONTINUE','BONJOUR',

'REPORT','REQUEST','FOR','MAINTAIN','AND',

'ZURICH','RIGHT','CORRECTION','CONFIRM','PR
OCEED', 'ON','NON','GUTEN',
```

'CALLING','CAN', 'AH','LEFT','RIGHT')

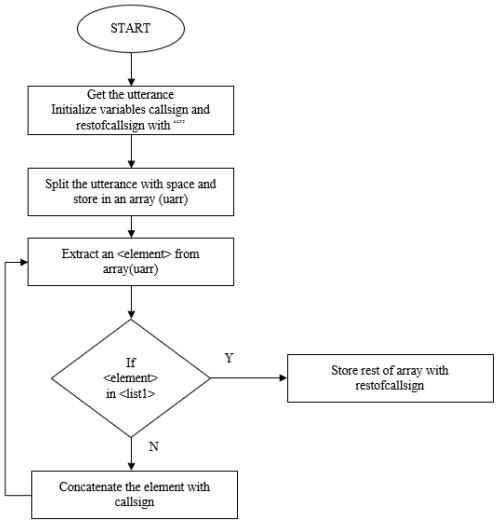


Figure 2 – Flowchart of Syntax Separation

WER values for the split utterances after syntax separation gave a positive result for Call Sign and Rest of call sign as listed in Figure 3 and **Error! Reference source not found.** respectively. This sets a positive outlook for the accuracy of the proposed method.

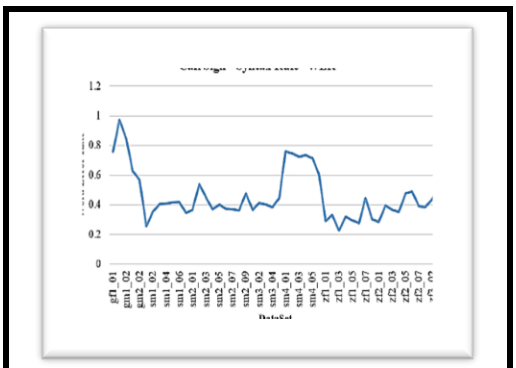


Figure 3 – Call Sign WER

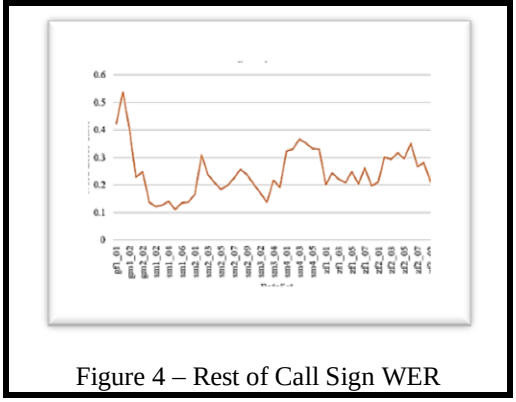


Figure 4 – Rest of Call Sign WER

The above results clearly convey that there is a good improvement of WER across the data sets when compared with the adapted baseline discussed earlier. Once again, gf1 dataset did not show good improvement whereas there is a significant improvement with gm1 dataset. The results are convincing that performing a syntax-based separation has helped in reducing the errors. Moreover, the syntax separation allows to have different algorithms to reduce the errors further which are discussed later. Almost all the prior work has considered the entire utterance for error detection and correction, and this is the first time the separation algorithm has been tried before performing error detection and correction.

1.2. Utterance Clustering

In string clustering, the interplay between string distance scores and context scores is harnessed through clustering analysis. Conventionally, "bag of words" or co-occurrence scores are computed within a single sentence, encompassing multiple words or characters. However, in the context of ATSR, co-occurrence cluster analysis is extended to capture the conversations centered around a specific aircraft. During a conversation that spans around 20 minutes (example), a controller interacts with multiple pilots, switching between different aircraft. Thus, a dedicated component is devised to retrieve the context for a particular aircraft, subsequently calculating the context score for a given utterance. Co-occurrence analysis centers on repeatability – the tendency for specific word combinations to appear recurrently in conversations. In the context of ATC utterances, the robust syntactic structure, often comprising various named entities, tends to deviate from typical word combination patterns that excel in co-occurrence analysis. Nevertheless, an exception to this trend is Call Signs. Call Signs, due to their static nature throughout a conversation, exhibit a more distinct co-occurrence pattern. In contrast, the rest of the Call Sign is characterized by greater dynamism, leading to a reduced co-occurrence pattern. This discrepancy highlights the nuanced nature of co-occurrence analysis in the ATC domain, further reinforcing the significance of context and semantics in deciding error correction strategies. This is depicted in Table 1.

Table 1 - Example of Co-occurrences

Co-occurrence in Call Sign	Co-occurrence in Rest of Call Sign
1. aero lloyd five nine zero 2. aero lloyd five nine zero 3. aero lloyd five nine zero 4. aero lloyd five nine zero	1. proceed direct to Frankfurt 2. proceed direct to gotil 3. proceed direct to Frankfurt 4. proceed direct to frankfurt

Indeed, within the rest of the Call Sign, certain segments, particularly the <command> portion of the ATC utterance syntax, do co-occur. However, the <command information> segment boasts a higher degree of dynamism, resulting in a diminished co-occurrence score for the rest of the Call Sign. It is worth noting that while word combinations hold significance in ATC utterances, their contextual and semantic implications differ substantially from those in

conventional English conversations. To address this context-specific challenge, a context-aware co-occurrence algorithm is introduced. This algorithm is designed to generate the context surrounding a callsign or rest of Call Sign utterance. By capturing the surrounding dialogue, the algorithm can then compute a co-occurrence score those accounts for repeatability patterns within the specific context. This approach leverages the unique nature of ATC communication to yield more relevant co-occurrence scores, tailored to the distinct linguistic and operational nuances of this domain. These scores are emphasizing the algorithm's potential in enhancing error correction measures and optimizing the accuracy of utterance interpretation.680 utterances were analyzed, and 12 utterances were repeated more than 10 times in the set of 680 utterances. The distribution of occurrence count for the different utterances is provided in Figure 4.

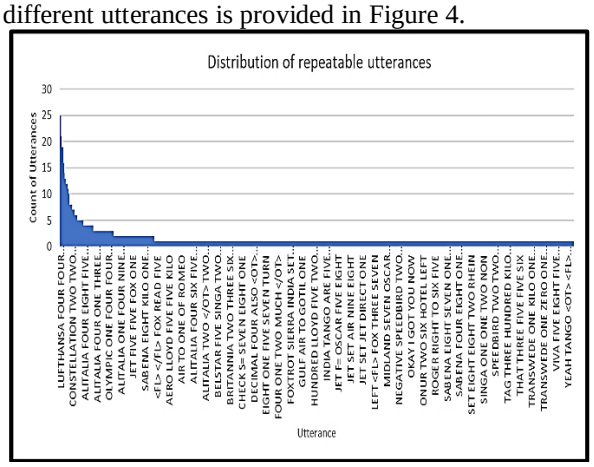


Figure 4 - Repeatability in Air Traffic Utterances

Out of 680 utterances only 124 (18%) of the utterances appeared only once and the remaining 82% of the utterances were repeated more than once. Few limitations are observed to group the occurrence based on co-occurrence score and hence better unsupervised grouping algorithms i.e., clustering algorithms are required. The proposed method uses different clustering algorithms to perform error detection and then uses string distance algorithms for error correction. The purpose of considering different algorithms is to understand the characteristic of various parts of utterance in ATSR. As observed, co-occurrence pattern is a strong indication of repeatability and leading us to apply unsupervised clustering algorithms.

1.3. Co-occurrences Using String Clustering Algorithms

Maximum occurrences piggybacking on the repeatability pattern was discussed in the previous section. Quite a few earlier works discussed how to use context to improve the error correction Oualil Y et al. [7], Nguyen V et al.[8], Srinivasamurthy A [9]. Context ranges from an airport database to an external system such as radar which has the ground truth of callsigns, frequencies, waypoints, and other named entities in ATC. These methods look good but there are numerous systems which maintain different data in the air traffic systems and cockpit systems. So, choosing a standard system is still a challenge and the earlier work did not address this. The proposal here is to use the local context, that is, group the utterances, named entities for a specific duration. Here the duration becomes the context. This work compares standard string clustering algorithms such as k-means, Agglomerative Clustering and Density based scan (DB Scan) for the purpose of co-occurrence generation. Sum of squared distances determines the cluster size and the cluster entries.

K-Means stands out as a widely employed "clustering" algorithm renowned for its efficacy. This algorithm operates by maintaining a set of 'k' centroids, which serve as reference points for delineating clusters. The determination of a data point's cluster affiliation is contingent upon its proximity to the centroid of that cluster in comparison to all other centroids. K-Means achieves optimal centroids through an iterative process involving two main steps. The algorithm initializes the centroids to be distant from each other leading to more stable results than random initialization.

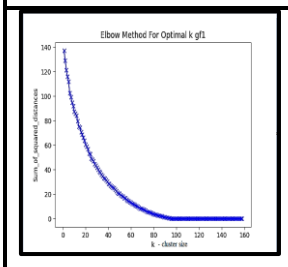
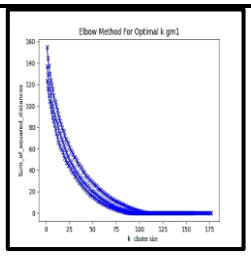
Agglomerative clustering works in a way that build nested clusters by merging successive data points.

Density-based spatial clustering of applications with noise (DBSCAN) groups together points that are closely packed together.

1.4. Determination of Cluster Size

Determination of cluster size is another key challenge and elbow principle was used to define the cluster size. In an ideal situation, the number of unique utterances should be the cluster size. This parameter was evaluated for all the datasets. Optimal cluster size was determined for each of the data sets using the elbow curves as given in Table 2.

Table 2 – Cluster Size Determination

Dataset gf1	**Dataset gm1**
	

A comparison chart of cluster size for the different algorithms based on elbow curve is given in Figure 5. A closer look at the chart reveals that the cluster size is in line with the unique count which is the expectation. Call Sign counts in few points are not in line with the utterance count. This is observed due to the incorrectly decoded utterances.

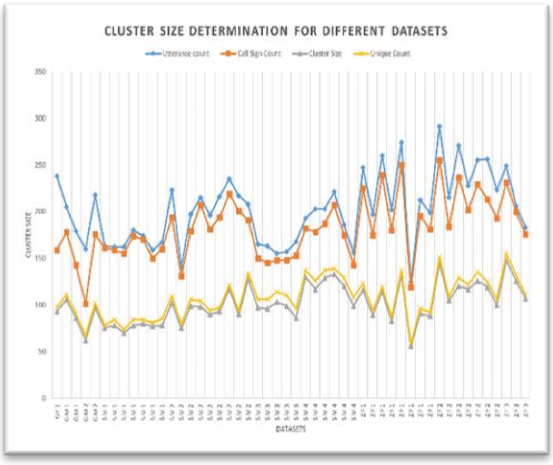


Figure 5 – Cluster size Determination

Four metrics are presented in Figure 5.

1. Utterance Count – Indicates the number of utterances in the dataset referred in x-axis.
2. Call Sign Count – Indicates the extracted call sign count using syntax segregation for the dataset mapped in x-axis.
3. Cluster Size – Dynamically determined cluster size for each of the data set in x-axis.
4. Unique Count – Unique number of call signs in the dataset which is the ideal or expected cluster size in x-axis.
5. Cluster size on the y-axis refers to the range of cluster size for all datasets.

2. Methodology

Figure 6 lists the steps in this methodology on how syntax segregation, co-occurrences, clustering algorithms and string distance algorithms are combined to reduce errors in ATSR.

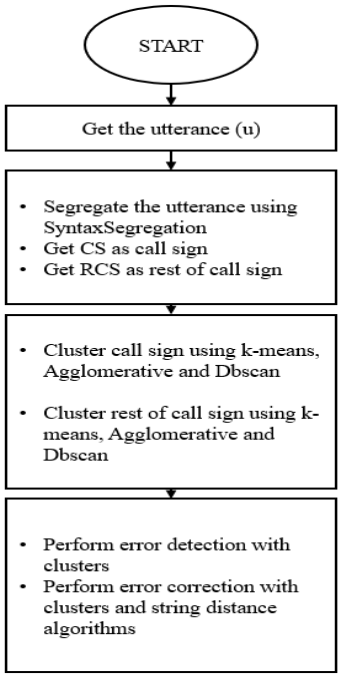


Figure 6 – Steps in Proposed Rule Based Method

2.1. Error Detection Using Clusters

After the syntax segregation, the results are sent to clustering step where the data is clustered based on k-means, agglomerative and dbscan algorithms. The output of the clustering algorithms is given in Table 3. The sample output reveals how the similar utterances of callsigns are clustered by the algorithm. All the entries in each cluster can point to the same call sign. Clustering helps to group similar callsigns in a particular group.

Table 3 - Cluster Output

Dataset	Cluster data
gf1	k-means output Cluster 1 CROSS AIR TWO SIX NINE ZERO, CROSS AIR TWO SIX NINE ZERO, CROSS AIR TWO SIX NINE ZERO, CROSS AIR TWO SIX NINE ZERO Agglomerative output Cluster 1 CROSS AIR FIVE ONE TWO, CROSS AIR FIVE ONE TWO, CROSS AIR FIVE ONE TWO FOUR SIX FIVE TWO DBScan output Cluster 1 CROSS AIR FIVE ONE TWO, CROSS AIR FIVE ONE TWO, CROSS AIR FIVE ONE TWO

The rule to detect errors using clusters is implemented in such a way that, for each callsign, look for the maximum number of cluster items. This maximum number per cluster is compared among the three algorithms to find out which has given the better results. Figure 7 plots the cluster items count for all call signs in gf1 dataset across different clustering algorithms.

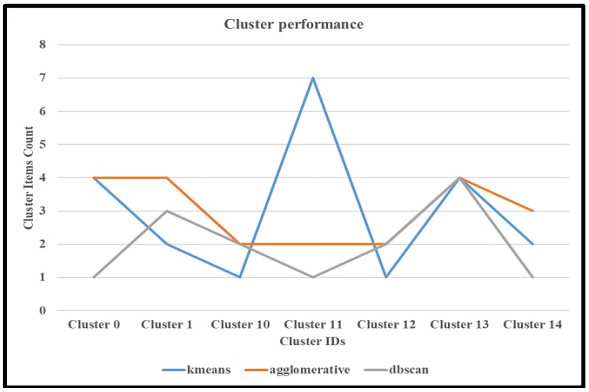


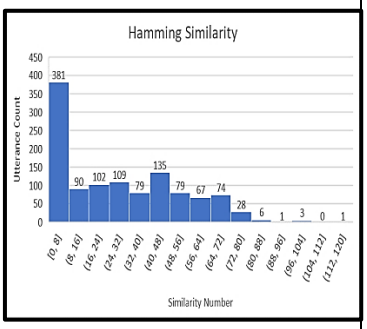
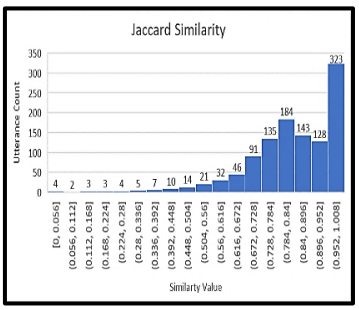
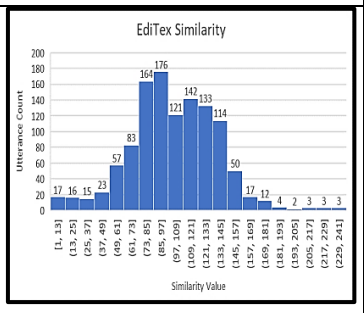
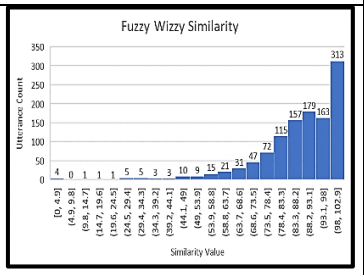
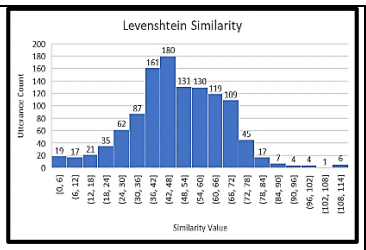
Figure 7 – Cluster Performance

Based on the above chart, k-means had better clustering than the rest of the algorithms. For subsequent steps, the output from k-means clusters is used.

2.2. Error Correction Using String Distance Based Algorithms

Similarity score provided by string-matching algorithms has better metrics to determine errors and correct the utterances. Table 4 shows the histogram of similarity score for different similarity algorithms.

Table 4 – Error Correction Performance of Similarity Algorithms

Similarity Algorithm	Histogram	WER
Hamming		**16%**
Jaccard		**37%**
EdiTex		**33%**
Fuzzy Wizzy		**37%**
Levenshtein		**33%**

WER was reduced by 9% with Hamming similarity algorithm when compared with the baseline and it is closer to Chen S et al. [10] adapted WER of 16%. The top value from the histogram indicates that it would go down further. Concept Recognition rate (ConR) was higher than observed by Chen S et al. [10].

The proposed method's metrics comparison (best observed with our experiments) with the baseline is given in Table 5.

Table 5 – Rule-based WER with Baseline

Approaches	Full Utterance WER	Call Sign only WER	Rest of Call Sign only WER
Baseline (Adapted LM)	27%	29%	17%
Proposed method	16%	17%	22%

A comparison of WER for rule-based with the earlier approaches is given in Figure 8. As seen, WER for rule based has improved for most datasets when compared with the WER with adaptation technique.

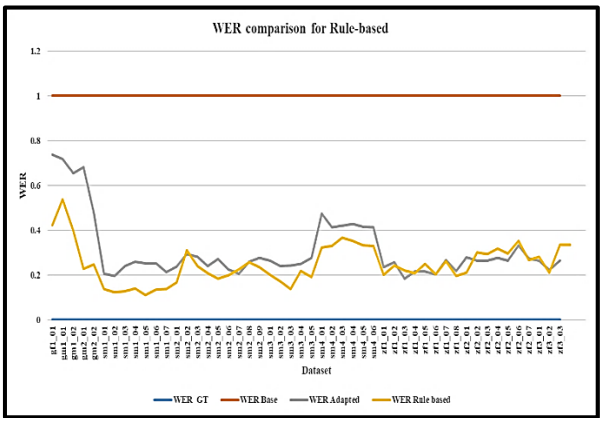


Figure 8 – WER for Rule-based Method

3. Conclusion

This work starts with a need and significance of rule-based algorithms in air traffic speech recognition and then proposes Syntax Segregation algorithm, clustering algorithm for performing error detection using the local context. This was followed by string distance-based error correction algorithm. Overall, the feasibility of using different algorithms for various parts of the utterance has been proven. The rest of call sign part is not explored much due to the less co-

occurrence score in a local context. Call signs can appear at the end of the utterance or sometimes in between. This work does not handle these exceptional cases well and requires further tuning. This work can be extended by considering n-gram models if Call Signs appear at distinct positions. Second level of segregation of call sign or rest of call sign is not considered in this work. The algorithm needs to take care of cases where words are omitted in the decoded output.

References

[1] Cordero, J.M., Rodríguez, N.D., Miguel, J., Pablo, D., & Crida (2013). Automated Speech Recognition in Controller Communications applied to Workload Measurement.

[2] Oualil, Y., M. Schulder, H. Helmke, A. Schmidt, & D. Klakow. (2015). Real-Time Integration of Dynamic Context Information for Improving Automatic Speech Recognition. 2107–2111.

[3] Nguyen V., & H. Holone. (2016). N-best list re-ranking using syntactic score: A solution for improving speech recognition accuracy in air traffic control. 1309–1314.

[4] Srinivasamurthy, A., P. Motlicek, I. Himawan, G. Szaszák, Y. Oualil, & H. Helmke. (2017). Semi-supervised Learning with Semantic Knowledge Extraction for Improved Speech Recognition in Air Traffic Control. 2017, 2406–2410

[5] Pardo, J. M., J. Ferreiros, V. Sama, R. de Cordoba, J. Macias-Guarasa, J. M. Montero, R. San-Segundo, L. F. D'Haro, G. Gonzalez. (2011). Automatic Understanding of ATC Speech: Study of Prospectives and Field Experiments for Several Controller Positions. IEEE Transactions on Aerospace and Electronic Systems, 47(4), 2709–2730.

[6] Badrinath, S., & B, Hasini. (2021). Automatic Speech Recognition for Air Traffic Control Communications. Transportation Research Record, 036119812110363.

[7] Oualil, Y., M. Schulder, H. Helmke, A. Schmidt, & D. Klakow. (2015). Real-Time Integration of Dynamic Context Information for Improving Automatic Speech Recognition. 2107–2111.

[8] Nguyen V., & H. Holone. (2016). N-best list re-ranking using semantic relatedness and syntactic score: An approach for improving speech recognition accuracy in air traffic control. International Conference on Control, Automation and Systems, 1315–1319.

[9] Srinivasamurthy, A., Motlicek, P., Singh, M., Oualil, Y., Kleinert, M., Ehr, H., & Helmke, H. (2018). Iterative Learning of Speech Recognition Models for Air Traffic Control. Interspeech, 3519–3523.

[10] Chen S, H. D. Kopald, R. Chong, D. Wei, & Z. Levonian. (2017). Read Back Error Detection using Automatic Speech Recognition.